Cisco IOS Security Configuration Guide

Release 12.4

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Customer Order Number: DOC-7817484=
Text Part Number: 78-17484-01

Cisco IOS Security Configuration Guide

About Cisco IOS Software Documentation for Release 12.4 xcvii

Documentation Objectives xcvii

Audience xcvii

Documentation Organization for Cisco IOS Release 12.4 xcviii

Document Conventions civ

Obtaining Documentation cv

Cisco.com cv

Product Documentation DVD cvi

Ordering Documentation cvi

Documentation Feedback cvi

Cisco Product Security Overview cvii

Reporting Security Problems in Cisco Products cvii

Obtaining Technical Assistance cviii

Cisco Technical Support & Documentation Website cviii

Submitting a Service Request cviii

Definitions of Service Request Severity cix

Obtaining Additional Publications and Information cix

Using Cisco IOS Software for Release 12.4 cxi

Understanding Command Modes cxi

Getting Help cxii

Example: How to Find Command Options cxiii

Using the no and default Forms of Commands cxv

Saving Configuration Changes cxvi

Filtering Output from the show and more Commands cxvi

Finding Additional Feature Support Information cxvii

Security Overview 1

About This Guide 1

Authentication, Authorization, and Accounting (AAA) 2

Security Server Protocols 2

Traffic Filtering, Firewalls, and Virus Detection 3

IP Security (IPSec) and Internet Key Exchange (IKE) 4

Public Key Infrastructure (PKI) 5

Other Security Features 5
Cisco IOS Secure Infrastructure 6
Appendixes 7
Creating Effective Security Policies 7
The Nature of Security Policies 7
Two Levels of Security Policies 8
Tips for Developing an Effective Security Policy 8
Identifying Your Network Assets to Protect 8
Determining Points of Risk 9
Limiting the Scope of Access 9
Identifying Assumptions 9
Determining the Cost of Security Measures 9
Considering Human Factors 9
Keeping a Limited Number of Secrets 10
Implementing Pervasive and Scalable Security 10
Understanding Typical Network Functions 10
Remembering Physical Security 10
Identifying Security Risks and Cisco IOS Solutions 11
Preventing Unauthorized Access into Networking Devices 11
Preventing Unauthorized Access into Networks 12
Preventing Network Data Interception 13
Preventing Fraudulent Route Updates 14

PART 1: AUTHENTICATION, AUTHORIZATION, AND ACCOUNTING (AAA)

AAA Overview 17
In This Chapter 17
About AAA Security Services 17
Benefits of Using AAA 18
AAA Philosophy 19
Method Lists 19
Where to Begin 20
Overview of the AAA Configuration Process 21
Enabling AAA 21
Disabling AAA 22
What to Do Next 22

Authentication

Configuring Authentication 25

In This Chapter 25

Named Method Lists for Authentication 25

Method Lists and Server Groups 26

Method List Examples 27

AAA Authentication General Configuration Procedure 28

AAA Authentication Methods Configuration Task List 28

Configuring Login Authentication Using AAA 29

Login Authentication Using Enable Password 31

Login Authentication Using Kerberos 31

Login Authentication Using Line Password 31

Login Authentication Using Local Password 31

Login Authentication Using Group RADIUS 32

Login Authentication Using Group TACACS+ 32

Login Authentication Using group group-name 32

Configuring PPP Authentication Using AAA 33

PPP Authentication Using Kerberos 34

PPP Authentication Using Local Password 34

PPP Authentication Using Group RADIUS 35

PPP Authentication Using Group TACACS+ 35

PPP Authentication Using group group-name 35

Configuring AAA Scalability for PPP Requests 36

Configuring ARAP Authentication Using AAA 36

ARAP Authentication Allowing Authorized Guest Logins 38

ARAP Authentication Allowing Guest Logins 38

ARAP Authentication Using Line Password 38

ARAP Authentication Using Local Password 38

ARAP Authentication Using Group RADIUS 39

ARAP Authentication Using Group TACACS+ 39

ARAP Authentication Using Group group-name 39

Configuring NASI Authentication Using AAA 40

NASI Authentication Using Enable Password 41

NASI Authentication Using Line Password 41

NASI Authentication Using Local Password 41

NASI Authentication Using Group RADIUS 42

NASI Authentication Using Group TACACS+ 42

NASI Authentication Using group group-name 42

Specifying the Amount of Time for Login Input 43

Enabling Password Protection at the Privileged Level 43
Changing the Text Displayed at the Password Prompt 44
Configuring Message Banners for AAA Authentication 44
Configuring a Login Banner 45
Configuring a Failed-Login Banner 45
Configuring AAA Packet of Disconnect 46
Enabling Double Authentication 46
How Double Authentication Works 46
Configuring Double Authentication 47
Accessing the User Profile After Double Authentication 48
Enabling Automated Double Authentication 49
Non-AAA Authentication Methods 51
Configuring Line Password Protection 51
Establishing Username Authentication 52
Enabling CHAP or PAP Authentication 53
Enabling PPP Encapsulation 54
Enabling PAP or CHAP 54
Inbound and Outbound Authentication 55
Enabling Outbound PAP Authentication 55
Refusing PAP Authentication Requests 56
Creating a Common CHAP Password 56
Refusing CHAP Authentication Requests 56
Delaying CHAP Authentication Until Peer Authenticates 57
Using MS-CHAP 57
Authentication Examples 58
RADIUS Authentication Examples 59
TACACS+ Authentication Examples 60
Kerberos Authentication Examples 61
AAA Scalability Example 61
Login and Failed Banner Examples 62
AAA Packet of Disconnect Server Key Example 63
Double Authentication Examples 63
Configuration of the Local Host for AAA with Double Authentication Examples 64
Configuration of the AAA Server for First-Stage (PPP) Authentication and Authorization Example 64
Configuration of the AAA Server for Second-Stage (Per-User) Authentication and Authorization Examples 65
Complete Configuration with TACACS+ Example 66
Automated Double Authentication Example 69
MS-CHAP Example 71

AAA Double Authentication Secured by Absolute Timeout 73

Contents 73

Prerequisites for AAA Double Authentication Secured by Absolute Timeout 74

Restrictions for AAA Double Authentication Secured by Absolute Timeout 74

Information About AAA Double Authentication Secured by Absolute Timeout 74

Securing a AAA Double Authentication 74

How to Apply AAA Double Authentication Secured by Absolute Timeout 75

Verifying AAA Double Authentication Secured by Absolute Timeout 75

Examples 77

AAA Double Authentication Secured by Absolute Timeout: Examples 77

RADIUS User Profile: Example 77

TACACS+ User Profile: Example 78

Additional References 81

Related Documents 81

Standards 81

MIBs 81

RFCs 81

Technical Assistance 82

Command Reference 82

Login Password Retry Lockout 83

Contents 83

Prerequisites for Login Password Retry Lockout 83

Restrictions for Login Password Retry Lockout 84

Information About Login Password Retry Lockout 84

Locking Out a Local AAA User Account 84

How to Configure Login Password Retry Lockout 84

Configuring Login Password Retry Lockout 85

Unlocking a Locked-Out User 86

Clearing the Unsuccessful Attempts of a User 86

Monitoring and Maintaining Login Password Retry Lockout 87

Configuration Examples for Login Password Retry Lockout 87

Login Password Retry Lockout: Example 87

show aaa local user lockout Command: Example 88

Additional References 88
Related Documents 88
Standards 89
MIBs 89
RFCs 89
Technical Assistance 89
Command Reference 89
Glossary 90

MSCHAP Version 2 91
Feature Overview 91
Benefits 91
Restrictions 92
Related Documents 92
Supported Platforms 92
Supported Standards, MIBs, and RFCs 93
Prerequisites 93
Configuration Tasks 94
Configuring MSCHAP V2 Authentication 94
Verifying MSCHAP V2 Configuration 94
Configuration Examples 95
Local Authentication Example 96
RADIUS Authentication Example 96
Command Reference 96

RADIUS EAP Support 97
Feature Overview 97
How EAP Works 98
Newly Supported Attributes 98
Benefits 98
Restrictions 98
Related Documents 99
Supported Platforms 99
Supported Standards, MIBs, and RFCs 100
Prerequisites 100
Configuration Tasks 100
Configuring EAP 101
Verifying EAP 101
Configuration Examples 101

EAP Local Configuration on Client Example 102
EAP Proxy Configuration for NAS Example 102
Command Reference 103
Glossary 104

RADIUS Packet of Disconnect 105
Feature Overview 105
Benefits 106
Restrictions 106
Related Features and Technologies 106
Related Documents 106
Supported Platforms 107
Supported Standards, MIBs, and RFCs 108
Prerequisites 108
Configuration Tasks 108
Configuring AAA POD Server 109
Verifying AAA POD Server 110
Troubleshooting Tips 110
Configuration Examples 111
AAA POD Server Example 111
Command Reference 111
Glossary 112

Configuring Authorization 113
In This Chapter 113
Named Method Lists for Authorization 113
AAA Authorization Methods 114
Method Lists and Server Groups 115
AAA Authorization Types 116
AAA Authorization Prerequisites 116
AAA Authorization Configuration Task List 116
Configuring AAA Authorization Using Named Method Lists 117
Authorization Types 117
Authorization Methods 118
Disabling Authorization for Global Configuration Commands 118
Configuring Authorization for Reverse Telnet 119
Authorization Attribute-Value Pairs 119
Authorization Configuration Examples 120

Named Method List Configuration Example 120
TACACS+ Authorization Examples 121
RADIUS Authorization Example 122
Reverse Telnet Authorization Examples 122

Configuring Accounting 125
In This Chapter 125
Named Method Lists for Accounting 125
Method Lists and Server Groups 127
AAA Accounting Methods 128
AAA Accounting Types 128
Network Accounting 128
Connection Accounting 131
EXEC Accounting 133
System Accounting 134
Command Accounting 135
Resource Accounting 135
AAA Resource Failure Stop Accounting 135
AAA Resource Accounting for Start-Stop Records 137
AAA Accounting Enhancements 137
AAA Broadcast Accounting 138
AAA Session MIB 138
AAA Accounting Prerequisites 139
AAA Accounting Configuration Task List 139
Configuring AAA Accounting Using Named Method Lists 140
Accounting Types 140
Accounting Record Types 141
Accounting Methods 141
Suppressing Generation of Accounting Records for Null Username Sessions 142
Generating Interim Accounting Records 143
Generating Accounting Records for Failed Login or Session 143
Specifying Accounting NETWORK-Stop Records Before EXEC-Stop Records 143
Configuring AAA Resource Failure Stop Accounting 144
Configuring AAA Resource Accounting for Start-Stop Records 144
Configuring AAA Broadcast Accounting 145
Configuring Per-DNIS AAA Broadcast Accounting 145
Configuring AAA Session MIB 145
Monitoring Accounting 146
Troubleshooting Accounting 146

Accounting Attribute-Value Pairs 146
Accounting Configuration Examples 146
Configuring Named Method List Example 147
Configuring AAA Resource Accounting 149
Configuring AAA Broadcast Accounting Example 149
Configuring Per-DNIS AAA Broadcast Accounting Example 150
AAA Session MIB Example 150

PART 2: SECURITY SERVER PROTOCOLS

RADIUS

Configuring RADIUS 155
In This Chapter 155
About RADIUS 155
RADIUS Operation 156
RADIUS Configuration Task List 157
Configuring Router to RADIUS Server Communication 158
Configuring Router to Use Vendor-Specific RADIUS Attributes 160
Configuring Router for Vendor-Proprietary RADIUS Server Communication 161
Configuring Router to Query RADIUS Server for Static Routes and IP Addresses 162
Configuring Router to Expand Network Access Server Port Information 162
Configuring AAA Server Groups 163
Configuring AAA Server Groups with Deadtime 164
Configuring AAA DNIS Authentication 165
Configuring AAA Server Group Selection Based on DNIS 165
Configuring AAA Preauthentication 167
Setting Up the RADIUS Profile for DNIS or CLID Preauthentication 168
Setting Up the RADIUS Profile for Call Type Preauthentication 169
Setting Up the RADIUS Profile for Preauthentication Enhancements for Callback 169
Setting Up the RADIUS Profile for a Remote Host Name Used for Large-Scale Dial-Out 170
Setting Up the RADIUS Profile for Modem Management 170
Setting Up the RADIUS Profile for Subsequent Authentication 170
Setting Up the RADIUS Profile for Subsequent Authentication Type 171
Setting Up the RADIUS Profile to Include the Username 171
Setting Up the RADIUS Profile for Two-Way Authentication 172
Setting Up the RADIUS Profile to Support Authorization 172
Configuring a Guard Timer 173
Specifying RADIUS Authentication 173

Specifying RADIUS Authorization 173
Specifying RADIUS Accounting 173
Configuring RADIUS Login-IP-Host 174
Configuring RADIUS Prompt 174
Configuring Suffix and Password in RADIUS Access Requests 175
Monitoring and Maintaining RADIUS 175
RADIUS Attributes 175
Vendor-Proprietary RADIUS Attributes 176
RADIUS Tunnel Attributes 176
RADIUS Configuration Examples 176
RADIUS Authentication and Authorization Example 177
RADIUS Authentication, Authorization, and Accounting Example 177
Vendor-Proprietary RADIUS Configuration Example 178
RADIUS Server with Server-Specific Values Example 179
Multiple RADIUS Servers with Global and Server-Specific Values Example 179
Multiple RADIUS Server Entries for the Same Server IP Address Example 180
RADIUS Server Group Examples 180
Multiple RADIUS Server Entries Using AAA Server Groups Example 180
AAA Server Group Selection Based on DNIS Example 181
AAA Preauthentication Examples 182
RADIUS User Profile with RADIUS Tunneling Attributes Example 183
Guard Timer Examples 183
L2TP Access Concentrator Examples 184
L2TP Network Server Examples 185

AAA Dead-Server Detection 187

Contents 187
Prerequisites for AAA Dead-Server Detection 187
Restrictions for AAA Dead-Server Detection 188
Information About AAA Dead-Server Detection 188
Criteria for Marking a RADIUS Server As Dead 188
How to Configure AAA Dead-Server Detection 188
Configuring AAA Dead-Server Detection 189
Troubleshooting Tips 189
Verifying AAA Dead-Server Detection 190
Configuration Examples for AAA Dead-Server Detection 190
Configuring AAA Dead-Server Detection: Example 191
debug aaa dead-criteria transactions Command: Example 191
show aaa dead-criteria Command: Example 191

Additional References 192
Related Documents 192
Standards 192
MIBs 192
RFCs 192
Technical Assistance 193
Command Reference 193

ACL Default Direction 195
Feature Overview 195
Benefits 195
Related Documents 196
Supported Platforms 196
Supported Standards, MIBs, and RFCs 196
Prerequisites 197
Configuration Tasks 197
Configuring RADIUS Attribute 11 (Filter-Id) 197
Verifying RADIUS Attribute 11 (Filter-Id) 197
Configuration Examples 198
Default Direction of Filters via RADIUS Attribute 11 (Filter-Id) Configuration Example 198
RADIUS User Profile with Filter-Id Example 198
Command Reference 198

Attribute Screening for Access Requests 199
Contents 199
Prerequisites for Attribute Screening for Access Requests 200
Restrictions for Attribute Screening for Access Requests 200
Information About Attribute Screening for Access Requests 200
Configuring an NAS to Filter Attributes in Outbound Access Requests 200
How to Configure Attribute Screening for Access Requests 200
Configuring Attribute Screening for Access Requests 201
Configuring a Router to Support Downloadable Filters 202
Troubleshooting Tips 203
Monitoring and Maintaining Attribute Filtering for Access Requests 203
Configuration Examples for Attribute Filtering for Access Requests 204
Attribute Filtering for Access Requests: Example 204
Attribute Filtering User Profile: Example 204
debug radius Command: Example 205

Additional References 205
Related Documents 205
Standards 205
MIBs 206
RFCs 206
Technical Assistance 206
Command Reference 206

Enable Multilink PPP via RADIUS for Preauthentication User 207

Feature Overview 207
How MLP via RADIUS Works 208
Roles of the L2TP Access Server and L2TP Network Server 208
New Vendor-Specific Attributes 208
Benefits 209
Related Documents 209
Supported Platforms 209
Supported Standards, MIBs, and RFCs 210
Prerequisites 210
Configuration Tasks 210
Verifying MLP Negotiation via RADIUS in Preauthentication 210
Configuration Examples 211
LAC for MLP Configuration Example 211
LAC RADIUS Profile for Preauthentication Example 212
LNS for MLP Configuration Example 212
LNS RADIUS Profile Example 212
Command Reference 212
Glossary 213

Enhanced Test Command 215

Feature Overview 215
Benefits 215
Restrictions 216
Related Documents 216
Supported Platforms 216
Supported Standards, MIBs, and RFCs 217
Configuration Tasks 217
Configuring a User Profile 217
Associating a User Profile with a **test aaa group** Command 218
Verifying Enhanced Test Command 218

Configuration Examples 218
User Profile Associated With a test aaa group command Example 218
Command Reference 219
Glossary 220

Framed-Route in RADIUS Accounting 221

Contents 221
Prerequisites for Framed-Route in RADIUS Accounting 221
Information About Framed-Route in RADIUS Accounting 222
Framed-Route, Attribute 22 222
Framed-Route in RADIUS Accounting Packets 222
How to Monitor Framed-Route in RADIUS Accounting 222
Examples 222
debug radius Command Output: Example 223
Additional References 224
Related Documents 224
Standards 224
MIBs 224
RFCs 224
Technical Assistance 225
Command Reference 225

Offload Server Accounting Enhancement 227

Feature Overview 227
Benefits 228
Related Documents 228
Supported Platforms 228
Supported Standards, MIBs, and RFCs 229
Prerequisites 229
Configuration Tasks 229
Configuring Unique Session IDs 230
Configuring Offload Server to Synchronize with NAS Clients 230
Verifying Offload Server Accounting 230
Configuration Examples 230
Unique Session ID Configuration Example 231
Offload Server Synchronization with NAS Clients Example 231
Command Reference 231
Glossary 232

Per VRF AAA 233

Contents 233

Restrictions for Per VRF AAA 234

Information About Per VRF AAA 234

Per VRF AAA Functionality Overview 234

Benefits of Per VRF AAA 235

New Vendor-Specific Attributes (VSAs) 235

How to Configure Per VRF AAA 238

Configuring Per VRF AAA 239

Configuring AAA 239

Configuring Server Groups 239

Configuring Authentication, Authorization, and Accounting for Per VRF AAA 240

Configuring RADIUS-Specific Commands for Per VRF AAA 242

Configuring Interface-Specific Commands for Per VRF AAA 242

Configuring Per VRF AAA Using Local Customer Templates 244

Prerequisites 244

Configuring Authorization for Per VRF AAA with Local Customer Templates 244

Configuring Local Customer Templates 245

Configuring Per VRF AAA Using Remote Customer Templates 247

Prerequisites 247

Configuring Authentication for Per VRF AAA with Remote Customer Profiles 247

Configuring Authorization for Per VRF AAA with Remote Customer Profiles 248

Configuring the RADIUS Profile on the SP RADIUS Server 249

Verifying VRF Routing Configurations 249

Troubleshooting Per VRF AAA Configurations 250

Configuration Examples for Per VRF AAA 250

Per VRF AAA Example 251

Per VRF AAA Using a Locally Defined Customer Template Example 251

Per VRF AAA Using a Remote RADIUS Customer Template Example 252

Locally Configured Customer Template with RADIUS Attribute Screening and Broadcast Accounting Example 252

Remotely Configured Customer Template with RADIUS Attribute Screening and Broadcast Accounting Example 253

Additional References 254

Related Documents 254

Standards 255

MIBs 255

RFCs 255

Technical Assistance 255

Command Reference 256
Glossary 257

RFC-2867 RADIUS Tunnel Accounting 259

Contents 259
Restrictions for RFC-2867 RADIUS Tunnel Accounting 259
Information About RFC-2867 RADIUS Tunnel Accounting 260
Benefits of RFC-2867 RADIUS Tunnel Accounting 260
RADIUS Attributes Support for RADIUS Tunnel Accounting 260
How to Configure RADIUS Tunnel Accounting 264
Enabling Tunnel Type Accounting Records 264
VPDN Tunnel Events 264
What To Do Next 265
Verifying RADIUS Tunnel Accounting 266
Configuration Examples for RADIUS Tunnel Accounting 266
Configuring RADIUS Tunnel Accounting on LAC: Example 266
Configuring RADIUS Tunnel Accounting on LNS: Example 268
Additional References 270
Related Documents 270
Standards 270
MIBs 270
RFCs 270
Technical Assistance 270
Command Reference 271

RADIUS Attribute Screening 273

Feature Overview 273
Benefits 274
Restrictions 274
Related Documents 275
Supported Platforms 275
Supported Standards, MIBs, and RFCs 276
Prerequisites 276
Configuration Tasks 276
Configuring RADIUS Attribute Screening 277
Verifying RADIUS Attribute Screening 278
Configuration Examples 278
Authorization Accept Example 278
Accounting Reject Example 279

Authorization Reject and Accounting Accept Example 279
Rejecting Required Attributes Example 279
Command Reference 280
Glossary 281

RADIUS Centralized Filter Management 283
Contents 283
Feature Overview 284
Cache Management 284
New Vendor-Specific Attribute Support 284
Benefits 285
Restrictions 285
Related Documents 285
Supported Standards, MIBs, and RFCs 285
Prerequisites 286
Configuration Tasks 286
Configuring the RADIUS ACL Filter Server 286
Configuring the Filter Cache 287
Verifying the Filter Cache 287
Troubleshooting Tips 288
Monitoring and Maintaining the Filter Cache 288
Configuration Examples 288
NAS Configuration Example 288
RADIUS Server Configuration Example 289
RADIUS Dictionary and Vendors File Example 289
Debug Output Example 289
Command Reference 290

RADIUS Debug Enhancements 291
Feature Overview 291
Benefits 292
Restrictions 292
Related Features and Technologies 292
Related Documents 292
Supported Platforms 292
Supported Standards, MIBs, and RFCs 293
Prerequisites 293
Configuration Tasks 294
Configuring Default Debug ASCII Display 294

Configuring Debug Display in Brief Format 294
Configuring Debug Display in Hex Format 295
Verifying the debug radius Command 295
Configuration Examples 297
Default debug radius Command Example 297
Compact Debugging Output Example 298
Command Reference 299
Glossary 300

RADIUS Logical Line ID 301
Feature Overview 301
Benefits 302
Restrictions 302
Related Documents 302
Supported Platforms 302
Supported Standards, MIBs, and RFCs 303
Configuration Tasks 304
Configuring Preauthorization 304
Configuring the LLID in a RADIUS User Profile 304
Verifying Logical Line ID 305
Configuration Examples 305
LAC for Preauthorization Configuration Example 305
RADIUS User Profile for LLID Example 306
Command Reference 306

RADIUS NAS-IP-Address Attribute Configurability 307
Contents 307
Prerequisites for RADIUS NAS-IP-Address Attribute Configurability 308
Restrictions for RADIUS NAS-IP-Address Attribute Configurability 308
Information About RADIUS NAS-IP-Address Attribute Configurability 308
Problem Definition and Solution Background Information 308
Using the RADIUS NAS-IP-Address Attribute Configurability Feature 309
How to Configure RADIUS NAS-IP-Address Attribute Configurability 309
Configuring a RADIUS NAS-IP-Address Attribute Configurability 310
Monitoring and Maintaining RADIUS NAS-IP-Address Attribute Configurability 310
Examples 311
Configuration Examples for RADIUS NAS-IP-Address Attribute Configurability 311
Configuring a RADIUS NAS-IP-Address Attribute Configurability: Example 311
Additional References 312

Related Documents 312
Standards 312
MIBs 312
RFCs 312
Technical Assistance 313
Command Reference 313

RADIUS Route Download 315

Contents 315
Feature Overview 315
Benefits 316
Related Documents 316
Supported Platforms 316
Supported Standards, MIBs, and RFCs 317
Prerequisites 317
Configuration Tasks 317
Configuring RADIUS Route Download 317
Verifying RADIUS Route Download 318
Configuration Examples 318
RADIUS Route Download Configuration Example 318
Command Reference 318

RADIUS Support of 56-Bit Acct Session-Id 319

Contents 319
Prerequisites for RADIUS Support of 56-Bit Acct Session-Id 320
Information About RADIUS Support of 56-Bit Acct Session-Id 320
Acct-Session-Id Attribute 320
Acct-Session-Id-Count Attribute 320
Benefits of RADIUS Support of 56-Bit Acct Session-Id 321
How to Configure RADIUS Support of 56-Bit Acct Session-Id 321
Configuring RADIUS Support of 56-Bit Acct Session-Id 321
Configuration Examples for RADIUS Support of 56-Bit Acct Session-Id 322
Configuring RADIUS Support of 56-Bit Acct Session-Id Example 322
Additional References 322
Related Documents 322
Standards 322
MIBs 323
RFCs 323
Technical Assistance 323

Command Reference 323

RADIUS Tunnel Preference for Load Balancing and Fail-Over 325

Feature Overview 325

Industry-Standard Rather Than Proprietary Attributes 325

Load Balancing and Fail-Over in a Multivendor Network 326

Benefits 327

Restrictions 327

Related Features and Technologies 328

Related Documents 328

Supported Platforms 328

Supported Standards, MIBs, and RFCs 329

Prerequisites 329

Configuration Tasks 329

Configuration Example 330

Command Reference 330

Glossary 331

RADIUS Server Reorder on Failure 333

Contents 333

Prerequisites for RADIUS Server Reorder on Failure 334

Restrictions for RADIUS Server Reorder on Failure 334

Information About RADIUS Server Reorder on Failure 334

RADIUS Server Failure 334

How the RADIUS Server Reorder on Failure Feature Works 335

When RADIUS Servers Are Dead 335

How to Configure RADIUS Server Reorder on Failure 335

Configuring a RADIUS Server to Reorder on Failure 335

Monitoring RADIUS Server Reorder on Failure 337

Examples 338

Configuration Examples for RADIUS Server Reorder on Failure 339

Configuring a RADIUS Server to Reorder on Failure Example 339

Determining Transmission Order When RADIUS Servers Are Dead 339

Additional References 341

Related Documents 341

Standards 341

MIBs 342

RFCs 342

Technical Assistance 342

Command Reference 342

Subscriber Service Switch 343

Contents 343

Restrictions for Subscriber Service Switch 344

Information About Subscriber Service Switch 344

Benefits of Subscriber Service Switch 344

Backward Compatibility 345

How to Use Subscriber Service Switch 347

Enabling Domain Preauthorization on a LAC 347

Creating a RADIUS User Profile for Domain Preauthorization 348

Enabling Subscriber Service Switch Preauthorization 348

Verifying Subscriber Service Switch Call Operation 349

Troubleshooting the Subscriber Service Switch 350

Debug Commands Available for Subscriber Service Switch 350

Troubleshoot the Subscriber Service Switch 351

Configuration Examples for Subscriber Service Switch 353

Enable LAC Domain Authorization Example 353

Domain Preauthorization RADIUS User Profile Example 353

Enable Subscriber Service Switch Preauthorization Example 354

Verify Subscriber Service Switch Call Operation Example 354

Troubleshoot Subscriber Service Switch Examples 356

Troubleshoot the Subscriber Service Switch Operation Example 357

Troubleshoot the Subscriber Service Switch on the LAC—Normal Operation Example 358

Troubleshoot the Subscriber Service Switch on the LAC—Authorization Failure Example 360

Troubleshoot the Subscriber Service Switch on the LAC—Authentication Failure Example 362

Troubleshoot the Subscriber Service Switch at the LNS—Normal Operation Example 365

Troubleshoot the Subscriber Service Switch at the LNS—Tunnel Failure Example 367

Additional References 368

Related Documents 369

Standards 369

MIBs 369

RFCs 370

Technical Assistance 370

Command Reference 370

Glossary 372

Tunnel Authentication via RADIUS on Tunnel Terminator 375

Feature Overview 375

New RADIUS Attributes 377

Benefits 377
Restrictions 377
Related Documents 377
Supported Platforms 377
Supported Standards, MIBs, and RFCs 378
Prerequisites 378
Configuration Tasks 378
Configuring the LNS or LAC for Remote RADIUS Tunnel Authentication and Authorization 379
Verifying Remote RADIUS Tunnel Authentication and Authorization Configurations 379
Configuration Examples 380
L2TP Network Server (LNS) Configuration Example 381
RADIUS User Profile for Remote RADIUS Tunnel Authentication Example 381
Command Reference 381
Glossary 382

TACACS+

Configuring TACACS+ 385
In This Chapter 385
About TACACS+ 385
TACACS+ Operation 386
TACACS+ Configuration Task List 387
Identifying the TACACS+ Server Host 388
Setting the TACACS+ Authentication Key 389
Configuring AAA Server Groups 389
Configuring AAA Server Group Selection Based on DNIS 390
Specifying TACACS+ Authentication 391
Specifying TACACS+ Authorization 391
Specifying TACACS+ Accounting 392
TACACS+ AV Pairs 392
TACACS+ Configuration Examples 392
TACACS+ Authentication Examples 392
TACACS+ Authorization Example 394
TACACS+ Accounting Example 395
TACACS+ Server Group Example 395
AAA Server Group Selection Based on DNIS Example 395
TACACS+ Daemon Configuration Example 396

Per VRF for TACACS+ Servers 397

Contents 397

Prerequisites for Per VRF for TACACS+ Servers 397

Restrictions for Per VRF for TACACS+ Servers 398

Information About Per VRF for TACACS+ Servers 398

Per VRF for TACACS+ Servers Overview 398

How to Configure Per VRF for TACACS+ Servers 398

Configuring Per VRF on a TACACS+ Server 398

Verifying Per VRF for TACACS+ Servers 400

Configuration Examples for Per VRF for TACACS+ Servers 401

Configuring Per VRF for TACACS+ Servers: Example 401

Additional References 402

Related Documents 402

Standards 402

MIBs 402

RFCs 402

Technical Assistance 403

Command Reference 403

Configuring Kerberos 405

In This Chapter 405

About Kerberos 405

Kerberos Client Support Operation 407

Authenticating to the Boundary Router 407

Obtaining a TGT from a KDC 408

Authenticating to Network Services 408

Kerberos Configuration Task List 409

Configuring the KDC Using Kerberos Commands 409

Adding Users to the KDC Database 410

Creating SRVTABs on the KDC 410

Extracting SRVTABs 411

Configuring the Router to Use the Kerberos Protocol 411

Defining a Kerberos Realm 412

Copying SRVTAB Files 412

Specifying Kerberos Authentication 413

Enabling Credentials Forwarding 413

Opening a Telnet Session to the Router 414

Establishing an Encrypted Kerberized Telnet Session 414

Enabling Mandatory Kerberos Authentication 415

Enabling Kerberos Instance Mapping 415
Monitoring and Maintaining Kerberos 416
Kerberos Configuration Examples 416
Kerberos Realm Definition Examples 416
SRVTAB File Copying Example 416
Kerberos Configuration Examples 416
Encrypted Telnet Session Example 426

PART 3: TRAFFIC FILTERING, FIREWALLS, AND VIRUS DETECTION

Access Control Lists: Overview and Guidelines 429
In This Chapter 429
About Access Control Lists 429
What Access Lists Do 429
Why You Should Configure Access Lists 430
When to Configure Access Lists 430
Basic Versus Advanced Access Lists 431
Overview of Access List Configuration 431
Creating Access Lists 431
Assigning a Unique Name or Number to Each Access List 432
Defining Criteria for Forwarding or Blocking Packets 433
Creating and Editing Access List Statements on a TFTP Server 434
Applying Access Lists to Interfaces 434
Finding Complete Configuration and Command Information for Access Lists 435

Cisco IOS Firewall Overview 437
About Firewalls 437
The Cisco IOS Firewall Solution 437
The Cisco IOS Firewall Feature Set 438
Creating a Customized Firewall 438
Other Guidelines for Configuring Your Firewall 442

Configuring Lock-and-Key Security (Dynamic Access Lists) 445
In This Chapter 445
About Lock-and-Key 446
Benefits of Lock-and-Key 446
When to Use Lock-and-Key 446
How Lock-and-Key Works 447
Compatibility with Releases Before Cisco IOS Release 11.1 447

Risk of Spoofing with Lock-and-Key 448
Router Performance Impacts with Lock-and-Key 448
Prerequisites to Configuring Lock-and-Key 448
Configuring Lock-and-Key 449
Lock-and-Key Configuration Guidelines 450
Dynamic Access Lists 450
Lock-and-Key Authentication 450
The autocommand Command 451
Verifying Lock-and-Key Configuration 452
Maintaining Lock-and-Key 452
Displaying Dynamic Access List Entries 452
Manually Deleting Dynamic Access List Entries 453
Lock-and-Key Configuration Examples 453
Lock-and-Key with Local Authentication Example 453
Lock-and-Key with TACACS+ Authentication Example 454

Configuring IP Session Filtering (Reflexive Access Lists) 455
In This Chapter 455
About Reflexive Access Lists 455
Benefits of Reflexive Access Lists 456
What Is a Reflexive Access List? 456
How Reflexive Access Lists Implement Session Filtering 456
With Basic Access Lists 456
With Reflexive Access Lists 457
Where to Configure Reflexive Access Lists 457
How Reflexive Access Lists Work 457
Temporary Access List Entry Characteristics 457
When the Session Ends 458
Restrictions on Using Reflexive Access Lists 458
Prework: Before You Configure Reflexive Access Lists 458
Choosing an Interface: Internal or External 459
Reflexive Access Lists Configuration Task List 460
External Interface Configuration Task List 460
Internal Interface Configuration Task List 460
Defining the Reflexive Access List(s) 460
Mixing Reflexive Access List Statements with Other Permit and Deny Entries 461
Nesting the Reflexive Access List(s) 462
Setting a Global Timeout Value 463
Reflexive Access List Configuration Examples 463

External Interface Configuration Example 463
Internal Interface Configuration Example 465

Configuring TCP Intercept (Preventing Denial-of-Service Attacks) 467

In This Chapter 467
About TCP Intercept 467
TCP Intercept Configuration Task List 468
Enabling TCP Intercept 468
Setting the TCP Intercept Mode 469
Setting the TCP Intercept Drop Mode 469
Changing the TCP Intercept Timers 469
Changing the TCP Intercept Aggressive Thresholds 470
Monitoring and Maintaining TCP Intercept 471
TCP Intercept Configuration Example 471

Context-Based Access Control

Configuring Context-Based Access Control 475

In This Chapter 475
About Context-Based Access Control 475
What CBAC Does 476
Traffic Filtering 476
Traffic Inspection 476
Alerts and Audit Trails 477
Intrusion Prevention 477
What CBAC Does Not Do 478
How CBAC Works 478
How CBAC Works—Overview 478
How CBAC Works—Details 479
When and Where to Configure CBAC 481
The CBAC Process 481
Supported Protocols 482
CBAC Supported Protocols 482
RTSP and H.323 Protocol Support for Multimedia Applications 483
Restrictions 485
FTP Traffic and CBAC 485
IPSec and CBAC Compatibility 485
Memory and Performance Impact 485
CBAC Configuration Task List 486
Picking an Interface: Internal or External 486

Configuring IP Access Lists at the Interface 488
Basic Configuration 488
External Interface 490
Internal Interface 490
Configuring Global Timeouts and Thresholds 490
Half-Open Sessions 492
Defining an Inspection Rule 492
Configuring Application-Layer Protocol Inspection 492
Configuring Generic TCP and UDP Inspection 495
Applying the Inspection Rule to an Interface 496
Configuring Logging and Audit Trail 496
Other Guidelines for Configuring a Firewall 497
Verifying CBAC 498
RTSP with RDT 499
RTSP with TCP Only (Interleaved Mode) 499
RTSP with SMIL 500
RTSP with RTP (IP/TV) 500
H.323 V2 501
Monitoring and Maintaining CBAC 501
Debugging Context-Based Access Control 502
Generic Debug Commands 502
Transport Level Debug Commands 503
Application Protocol Debug Commands 503
Interpreting Syslog and Console Messages Generated by CBAC 504
Denial-of-Service Attack Detection Error Messages 504
SMTP Attack Detection Error Messages 504
Java Blocking Error Messages 505
FTP Error Messages 505
Audit Trail Messages 505
Turning Off CBAC 506
CBAC Configuration Examples 506
Ethernet Interface Configuration Example 507
ATM Interface Configuration Example 507
Remote Office to ISP Configuration Example 509
Remote Office to Branch Office Configuration Example 511
Two-Interface Branch Office Configuration Example 514
Multiple-Interface Branch Office Configuration Example 517

Cisco IOS Firewall Performance Improvements 525

Feature Overview 525

Throughput Improvement 526
Connections Per Second Improvement 526
CPU Utilization Improvement 526
Benefits 527
Restrictions 527
Related Documents 527
Supported Platforms 527
Supported Standards, MIBs, and RFCs 528
Configuration Tasks 529
Changing the Size of the Hash Table 529
Verifying CBAC Configurations 529
Configuration Examples 529
Changing the Size of the Hash Table Example 529
Command Reference 530

E-mail Inspection Engine 531

Contents 531
Prerequisites for E-mail Inspection Engine 532
Restrictions for E-mail Inspection Engine 532
Information About E-mail Inspection Engine 532
E-mail Inspection Engine Operation 532
Inspection 533
POP3 533
IMAP Protocol 533
Client Command Validation 534
SMTP 534
SSL 534
How to Configure E-mail Inspection Engine 534
Configuring Firewall Inspection of POP3 or IMAP E-mail 535
Verifying the E-mail Inspection Engine Configuration 536
Configuration Examples for E-mail Inspection Engine 537
Configuring IMAP and POP3 Protocol E-mail: Example 537
Additional References 538
Related Documents 538
Standards 538
MIBs 538
RFCs 538
Technical Assistance 539
Command Reference 539

Glossary 540

ESMTP Support for Cisco IOS Firewall 541

Contents 541

Prerequisites for ESMTP Support for Cisco IOS Firewall 541

Information About ESMTP Support for Cisco IOS Firewall 542

SMTP Functionality Overview 542

ESMTP Overview 543

SMTP Firewall and ESMTP Firewall Comparison 543

How to Configure a Firewall to Support ESMTP 546

Configuring a Firewall for ESMTP Inspection 546

Restrictions 546

Troubleshooting Tips 547

What to Do Next 548

Configuration Examples for Firewall ESMTP Support 548

ESMTP Inspection Configuration: Example 548

Additional References 548

Related Documents 548

Standards 549

MIBs 549

RFCs 549

Technical Assistance 550

Command Reference 550

Firewall ACL Bypass 551

Contents 551

Information About Firewall ACL Bypass 551

Benefits of Firewall ACL Bypass 552

Firewall ACL Bypass Functionality Overview 552

How to Use Firewall ACL Bypass 552

Configuration Examples for Verifying Firewall Session Information 552

Old **show ip inspect** CLI Output: Example 553

New show ip inspect CLI Output: Example 553

Additional References 554

Related Documents 554

Standards 554

MIBs 554

RFCs 554

Technical Assistance 554

Command Reference 555
Glossary 556

Firewall N2H2 Support 557

Contents 557
Restrictions for Firewall N2H2 Support 558
Information About Cisco N2H2 Support 558
Benefits of Firewall N2H2 Support 558
Feature Design of Firewall N2H2 Support 560
Supported N2H2 Filtering Methods 561
How to Configure N2H2 URL Support 561
Configuring Cisco IOS Firewall N2H2 URL Filtering 561
Prerequisites 561
Restrictions 562
Troubleshooting Tips 565
Verifying Firewall and N2H2 URL Filtering 566
Maintaining the Cache Table 566
Monitoring the URL Filter Subsytems 567
Configuration Examples for Firewall and Webserver 567
URL Filter Client (Firewall) Configuration Example 568
Additional References 572
Related Documents 572
Standards 572
MIBs 572
RFCs 573
Technical Assistance 573
Command Reference 573
Glossary 575

Firewall Stateful Inspection of ICMP 577

Contents 577
Restrictions for Firewall Stateful Inspection of ICMP 578
Information About Firewall Stateful Inspection of ICMP 578
Feature Design of Firewall Stateful Inspection of ICMP 578
How to Use Firewall Stateful Inspection of ICMP 579
Configuring Firewall Stateful Inspection for ICMP 579
Verifying Firewall and ICMP Session Information 580
Monitoring Firewall and ICMP Session Information 581
Configuration Examples for Stateful Inspection of ICMP 581

Firewall Stateful Inspection for ICMP Configuration Example 582
ICMP Session Verification Example 582
Additional References 583
Related Documents 583
Standards 583
MIBs 584
RFCs 584
Technical Assistance 585
Command Reference 585
Glossary 586

Firewall Support for SIP 587

Contents 587
Restrictions for Firewall Support for SIP 588
Information About Firewall Support for SIP 588
Firewall and SIP Overviews 588
Cisco IOS Firewall 588
SIP (Session Initiation Protocol) 589
Firewall for SIP Functionality Description 591
SIP Message Treatment by the Firewall 592
Call Database 593
How to Configure Your Firewall for SIP 594
Configuring Firewall for SIP Support 594
Prerequisite 594
Verifying Firewall for SIP Support 595
Monitoring Firewall for SIP Support 596
Configuration Examples for Firewall SIP Support 597
Firewall and SIP Configuration Example 597
Additional References 597
Related Documents 597
Standards 598
MIBs 598
RFCs 598
Technical Assistance 599
Command Reference 599

Firewall Websense URL Filtering 601

Contents 601
Restrictions for Firewall Websense URL Filtering 602

Information About Firewall Websense URL Filtering 602
Benefits of Firewall Websense URL Filtering 602
Feature Design of Firewall WEBSENSE Url Filtering 604
Supported Websense Server Features on a Cisco IOS Firewall 605
How to Configure Websense URL Filtering 605
Configuring Firewall WEBSENSE URL Filtering 605
Prerequisites 606
Restrictions 606
Troubleshooting Tips 609
Verifying Cisco IOS Firewall and Websense URL Filtering 610
Maintaining the Cache Table 610
Monitoring the URL Filter Subsytems 611
Configuration Examples for the Firewall and Webserver 612
URL Filter Client (Firewall) Configuration Example 612
Additional References 614
Related Documents 614
Standards 614
MIBs 614
RFCs 615
Technical Assistance 615
Command Reference 615
Glossary 617

Firewall Support of Skinny Client Control Protocol (SCCP) 619
Contents 619
Prerequisites for Firewall Support of Skinny Client Control Protocol (SCCP) 620
Restrictions for Firewall Support of Skinny Client Control Protocol (SCCP) 620
Information About Firewall Support of Skinny Client Control Protocol (SCCP) 620
Context-Based Access Control Overview 620
Skinny Overview 621
CBAC and Skinny Functionality Overview 621
How to Configure Your Firewall for Skinny Support 622
Configuring Basic Skinny CBAC Inspection 622
Setting Skinny CBAC Session Timeouts 623
Configuring Port to Application Mapping 624
Verifying Cisco IOS Firewall for Skinny Support 625
Monitoring Cisco IOS Firewall for Skinny Support 626
Configuration Examples for Firewall Skinny Support 626
Firewall and Skinny Configuration Example 627

Additional References 628
Related Documents 628
Standards 628
MIBs 628
RFCs 629
Technical Assistance 629
Command Reference 629

Granular Protocol Inspection 631

Contents 631
Prerequisites for Granular Inspection Protocol 631
Restrictions for Granular Inspection Protocol 632
Information About Granular Protocol Inspection 632
Cisco IOS Firewall 632
Granular Protocol Inspection 632
Benefits 633
How to Configure Granular Protocol Inspection 633
Defining Applications 633
Setting Up Inspection Rules 634
Verifying the Configuration 635
Configuration Examples for Granular Protocol Inspection 636
Defining an Application for the PAM Table: Example 636
Setting Up an Inspection Rule: Example 636
Verifying the Configuration: Example 638
Additional References 638
Related Documents 638
Standards 638
MIBs 639
RFCs 639
Technical Assistance 639
Command Reference 639
Glossary 640

HTTP Inspection Engine 641

Contents 641
Restrictions for HTTP Inspection Engine 642
Information About HTTP Inspection Engine 642
What Is a Security Policy? 642
Cisco IOS HTTP Application Policy Overview 642

How to Define and Apply an HTTP Application Policy to a Firewall for Inspection 642
Defining an HTTP Application Policy 643
Restrictions 643
What to Do Next 646
Applying an HTTP Application Policy to a Firewall for Inspection 646
Prerequisites 646
Troubleshooting Tips 648
Configuration Examples for Setting Up an HTTP Inspection Engine 649
Setting Up and Verifying an HTTP Inspection Engine: Example 649
Additional References 650
Related Documents 650
Standards 650
MIBs 650
RFCs 650
Technical Assistance 651
Command Reference 651

Inspection of Router-Generated Traffic 653

Contents 653
Prerequisites for Inspection of Router-Generated Traffic 653
Restrictions for Inspection of Router-Generated Traffic 654
Information About Inspection of Router-Generated Traffic 654
CBAC 654
Inspection of Router-Generated Traffic Overview 655
How to Configure Inspection of Router-Generated Traffic 655
Configuring H.323 Inspection 655
Configuring CBAC 656
Verifying the CBAC Configuration 658
Configuration Examples for Inspection of Router-Generated Traffic 660
Configuring CBAC with Inspection of H.323 Traffic: Example 660
Additional References 660
Related Documents 660
Standards 661
MIBs 661
RFCs 661
Technical Assistance 661
Command Reference 661
Glossary 662

Transparent Cisco IOS Firewall 663

Contents 663

Restrictions for Transparent Cisco IOS Firewall 664

Information About Transparent Cisco IOS Firewall 664

Benefit of the Transparent Firewall 664

Transparent Firewall Overview 664

Transparent Bridging Overview 665

Layer 2 and Layer 3 Firewalls Configured on the Same Router 665

How to Configure a Transparent Cisco IOS Firewall 665

Configuring a Bridge Group 665

BVI Configuration Requirements 665

Restrictions 666

Examples 667

Troubleshooting Tips 668

What to Do Next 668

Configuring Inspection and ACLs 668

Examples 669

Forwarding DHCP Traffic 670

Monitoring Transparent Firewall Events 670

Examples 671

Configuration Examples for Transparent Cisco IOS Firewall 671

Comprehensive Transparent Firewall Configuration: Example 672

Monitoring Telnet Connections via debug and show Output: Examples 674

Telnet Connection from the Client (97.0.0.2) to the Server (97.0.0.23) 675

Telnet Connection from the Server (97.0.0.23) to the Client (97.0.0.2) 677

Configuring and Verifying DHCP Pass-Through Traffic: Examples 677

Allowing DHCP Pass-Through Traffic: Example 678

Denying DHCP Pass-Through Traffic: Example 678

Additional References 679

Related Documents 679

Standards 679

MIBs 679

RFCs 680

Technical Assistance 680

Command Reference 680

Virtual Fragmentation Reassembly 681

Contents 681

Restrictions for Virtual Fragmentation Reassembly 682

Information About Virtual Fragmentation Reassembly 682
Detected Fragment Attacks 682
Automatically Enabling or Disabling VFR 683
How to Use Virtual Fragmentation Reassembly 683
Configuring VFR 683
Troubleshooting Tips 684
Configuration Examples for Fragmentation Reassembly 684
Configuring VFR and a Cisco IOS Firewall: Example 685
Additional References 686
Related Documents 686
Standards 687
MIBs 687
RFCs 687
Technical Assistance 687
Command Reference 687
Glossary 688

VRF Aware Cisco IOS Firewall 689

Contents 689
Prerequisites for VRF Aware Cisco IOS Firewall 690
Restrictions for VRF Aware Cisco IOS Firewall 690
Information About VRF Aware Cisco IOS Firewall 690
Cisco IOS Firewall 690
VRF 691
VRF-lite 692
Per-VRF URL Filtering 693
Alerts and Audit Trails 693
MPLS VPN 693
VRF-aware NAT 693
VRF-aware IPSec 694
VRF Aware Cisco IOS Firewall Deployment 695
Distributed Network Inclusion of VRF Aware Cisco IOS Firewall 695
Hub-and-Spoke Network Inclusion of VRF Aware Cisco IOS Firewall 697
How to Configure VRF Aware Cisco IOS Firewall 699
Configuring and Checking ACLs to Ensure that Only Inspected Traffic Can Pass Through the Firewall and that Non-Firewall Traffic is Blocked 699
Creating and Naming Firewall Rules and Applying the Rules to the Interface 700
Identifying and Setting Firewall Attributes 701
Verifying the VRF Aware Cisco IOS Firewall Configuration and Functioning 702

Configuration Examples for VRF Aware Cisco IOS Firewall 703
Additional References 712
Related Documents 712
Standards 713
MIBs 713
RFCs 713
Technical Assistance 713
Command Reference 714
Glossary 715

Configuring Cisco IOS Intrusion Prevention System (IPS) 717
Contents 717
Prerequisites for Configuring Cisco IOS IPS 718
Restrictions for Configuring Cisco IOS IPS 718
Information About Cisco IOS IPS 719
Cisco IOS IPS Overview 719
Benefits of Cisco IOS IPS 719
The Signature Definition File 720
Signature Microengines: Overview and Lists of Supported Engines 720
Lists of Supported Signature Engines 720
Supported Cisco IOS IPS Signatures in the attack-drop.sdf File 722
How to Load IPS-Based Signatures onto a Router 730
Installing Cisco IOS IPS on a New Router 731
Upgrading to the Latest Cisco IOS IPS Signature Definition File (SDF) 733
Prerequisites 733
Merging Built-In Signatures with the attack-drop.sdf File 735
Prerequisites 735
Monitoring Cisco IOS IPS Signatures via Syslog Messages or SDEE 739
SDEE Overview 739
Prerequisites 739
Troubleshooting Tips 740
Troubleshooting Cisco IOS IPS 741
Interpreting Cisco IOS IPS System Messages 741
Conditions of an SME Build Failure 743
Configuration Examples 743
Loading the Default Signatures: Example 743
Loading the attack-drop.sdf: Example 744
Merging the attack-drop.sdf File with the Default, Built-in Signatures: Example 744
Additional References 745

Related Documents 745
Standards 745
MIBs 745
RFCs 745
Technical Assistance 746
Feature Information for Configuring Cisco IOS IPS 746

Network Admission Control 749

Contents 749
Prerequisites for Network Admission Control 750
Restrictions for Network Admission Control 750
Information About Network Admission Control 750
Virus Infections and Their Effect on Networks 750
How Network Admission Control Works 751
Network Access Device 751
Cisco Trust Agent 751
Cisco Secure ACS 752
Remediation 752
Network Admission Control and Authentication Proxy 753
How to Configure Network Admission Control 753
Configuring the ACL and Admission Control 753
Configuring Global EAPoUDP Values 756
Configuring an Interface-Specific EAPoUDP Association 757
Configuring AAA for EAPoUDP 758
Configuring the Identity Profile and Policy 759
Clearing EAPoUDP Sessions That Are Associated with an Interface 761
Verifying Network Admission Control 761
Troubleshooting Network Admission Control 762
Configuration Examples for Network Admission Control 763
Network Admission Control: Example 763
Additional References 764
Related Documents 764
Standards 765
MIBs 765
RFCs 765
Technical Assistance 765
Command Reference 765
Glossary 767

Authentication Proxy

Configuring Authentication Proxy 771

In This Chapter 771

About Authentication Proxy 771

How the Authentication Proxy Works 772

Secure Authentication 774

Operation with JavaScript 774

Operation Without JavaScript 774

Using the Authentication Proxy 775

When to Use the Authentication Proxy 776

Applying the Authentication Proxy 777

Operation with One-Time Passwords 778

Compatibility with Other Security Features 778

NAT Compatibility 778

CBAC Compatibility 779

VPN Client Compatibility 779

Compatibility with AAA Accounting 779

Protection Against Denial-of-Service Attacks 780

Risk of Spoofing with Authentication Proxy 780

Comparison with the Lock-and-Key Feature 780

Restrictions 781

Prerequisites to Configuring Authentication Proxy 781

Authentication Proxy Configuration Task List 781

Configuring AAA 782

Configuring the HTTP Server 783

Configuring the Authentication Proxy 783

Verifying the Authentication Proxy 784

Checking the Authentication Proxy Configuration 785

Establishing User Connections with JavaScript 785

Establishing User Connections Without JavaScript 786

Monitoring and Maintaining the Authentication Proxy 787

Displaying Dynamic ACL Entries 787

Deleting Authentication Proxy Cache Entries 788

Authentication Proxy Configuration Examples 788

Authentication Proxy Configuration Example 788

AAA Configuration Example 789

HTTP Server Configuration Example 789

Authentication Proxy Configuration Example 789

Interface Configuration Example 789
Authentication Proxy, IPSec, and CBAC Configuration Example 789
Router 1 Configuration Example 790
Router 2 Configuration Example 791
Authentication Proxy, IPSec, NAT, and CBAC Configuration Example 793
Router 1 Configuration Example 794
Router 2 Configuration Example 794
AAA Server User Profile Example 797
CiscoSecure ACS 2.3 for Windows NT 797
CiscoSecure ACS 2.3 for UNIX 798
TACACS+ Server 799
Livingston Radius Server 800
Ascend Radius Server 800

Firewall Support of HTTPS Authentication Proxy 801

Contents 801
Prerequisites for Firewall Support of HTTPS Authentication Proxy 802
Restrictions for Firewall Support of HTTPS Authentication Proxy 802
Information About Firewall Support of HTTPS Authentication Proxy 802
Authentication Proxy 802
Feature Design for HTTPS Authentication Proxy 803
How to Use HTTPS Authentication Proxy 804
Configuring the HTTPS Server 804
Prerequisites 804
What to Do Next 805
Verifying HTTPS Authentication Proxy 805
Monitoring Firewall Support of HTTPS Authentication Proxy 806
Configuration Examples for HTTPS Authentication Proxy 807
HTTPS Authentication Proxy Support Example 807
RADIUS User Profile Example 810
TACACS User Profile Example 810
HTTPS Authentication Proxy Debug Example 811
Additional References 812
Related Documents 813
Standards 813
MIBs 813
RFCs 814
Technical Assistance 814
Command Reference 814

Glossary 815

Firewall Authentication Proxy for FTP and Telnet Sessions 817

Contents 817

Restrictions for Firewall Authentication Proxy for FTP and Telnet Sessions 818

Information About Firewall Authentication Proxy for FTP and Telnet Sessions 818

Feature Design for FTP and Telnet Authentication Proxy 818

FTP and Telnet Login Methods 818

Absolute Timeout 823

How to Configure FTP or Telnet Authentication Proxy 823

Configuring AAA 823

What to Do Next 825

Configuring the Authentication Proxy 825

Verifying FTP or Telnet Authentication Proxy 827

Monitoring and Maintaining FTP or Telnet Authentication Proxy Sessions 828

Configuration Examples for FTP and Telnet Authentication Proxy 828

Authentication Proxy Configuration Example 828

AAA Server User Profile Examples 829

TACACS+ User Profiles Example 829

Livingston RADIUS User Profiles Example 830

Ascend RADIUS User Profiles Example 831

Additional References 831

Related Documents 832

Standards 832

MIBs 832

RFCs 832

Technical Assistance 833

Command Reference 833

Configuring Port to Application Mapping 835

In This Chapter 835

About Port to Application Mapping 835

How PAM Works 836

System-Defined Port Mapping 836

User-Defined Port Mapping 837

Host-Specific Port Mapping 838

PAM and CBAC 838

When to Use PAM 838

PAM Configuration Task List 838

Configuring Standard ACLs 839
Configuring PAM 839
Verifying PAM 839
Monitoring and Maintaining PAM 840
PAM Configuration Examples 840
Mapping an Application to a Non-Standard Port Example 840
Mapping an Application with a Port Range Example 840
Invalid Port Mapping Entry Example 840
Mapping an Application to a Port for a Specific Host Example 841
Mapping an Application to a Port for a Subnet Example 841
Overriding a System-Defined Port Mapping Example 841
Mapping Different Applications to the Same Port Example 841

PART 4: IPSEC AND IKE

Internet Key Exchange for IPSec VPNs

Call Admission Control for IKE 847
Contents 847
Prerequisites for Call Admission Control for IKE 848
Information About Call Admission Control for IKE 848
IKE Session 848
Security Association Limit 848
System Resource Usage 848
How to Configure Call Admission Control for IKE 849
Configure the IKE Security Association Limit 849
Configure the System Resource Limit 850
Verifying the Call Admission Control for IKE Configuration 850
Configuration Examples for Call Admission Control for IKE 851
Configuring the IKE Security Association Limit: Example 851
Configuring the System Resource Limit: Example 851
Additional References 852
Related Documents 852
Standards 852
MIBs 852
RFCs 852
Technical Assistance 853
Command Reference 853

Certificate to ISAKMP Profile Mapping 855

Contents 855

Prerequisites for Certificate to ISAKMP Profile Mapping 855

Restrictions for Certificate to ISAKMP Profile Mapping 856

Information About Certificate to ISAKMP Profile Mapping 856

Certificate to ISAKMP Profile Mapping Overview 856

How Certificate to ISAKMP Profile Mapping Works 856

Assigning an ISAKMP Profile and Group Name to a Peer 857

How to Configure Certificate to ISAKMP Profile Mapping 857

Mapping the Certificate to the ISAKMP Profile 858

Verifying That the Certificate Has Been Mapped 858

Assigning the Group Name to the Peer 859

Monitoring and Maintaining Your Certificate to ISAKMP Profile Mapping 860

Configuration Examples for Certificate to ISAKMP Profile Mapping 860

Certificates Mapped to the ISAKMP Profile on the Basis of Arbitary Fields: Example 860

Group Name Assigned to a Peer That Is Associated with an ISAKMP Profile: Example 861

Mapping a Certificate to an ISAKMP Profile Verification: Example 861

Group Name Assigned to a Peer Verfication: Example 862

Additional References 863

Related Documents 863

Standards 864

MIBs 864

RFCs 864

Technical Assistance 864

Command Reference 864

Encrypted Preshared Key 865

Contents 865

Restrictions for Encrypted Preshared Key 865

Information About Encrypted Preshared Key 866

Using the Encrypted Preshared Key Feature to Securely Store Passwords 866

Changing a Password 866

Deleting a Password 866

Unconfiguring Password Encryption 866

Storing Passwords 867

Configuring New or Unknown Passwords 867

Enabling the Encrypted Preshared Key 867

How to Configure an Encrypted Preshared Key 867

Configuring an Encrypted Preshared Key 867

Troubleshooting Tips 868
Monitoring Encrypted Preshared Keys 869
Examples 869
What To Do Next 869
Configuring an ISAKMP Preshared Key 870
Example 870
Configuring an ISAKMP Preshared Key in ISAKMP Keyrings 871
Example 871
Configuring ISAKMP Aggressive Mode 872
Example 872
Configuring a Unity Server Group Policy 873
Example 874
Configuring an Easy VPN Client 874
Example 875
Configuration Examples for Encrypted Preshared Key 875
Encrypted Preshared Key: Example 875
No Previous Key Present: Example 876
Key Already Exists: Example 876
Key Already Exists But the User Wants to Key In Interactively: Example 876
No Key Present But the User Wants to Key In Interactively: Example 876
Removal of the Password Encryption: Example 877
Where to Go Next 877
Additional References 877
Related Documents 877
Standards 877
MIBs 877
RFCs 878
Technical Assistance 878
Command Reference 878

Configuring Internet Key Exchange for IPSec VPNs 879

Contents 879
Prerequisites for IKE Configuration 880
Restrictions for IKE Configuration 880
Information About Configuring IKE for IPSec VPNs 880
Supported Standards for Use with IKE 880
IKE Benefits 882
IKE Main Mode and Aggressive Mode 882
How to Configure IKE for IPSec VPNs 882

Creating IKE Policies: Security Parameters for IKE Negotiation 883
About IKE Policies 883
IKE Peers Agreeing Upon a Matching IKE Policy 883
Restrictions 884
Examples 886
Troubleshooting Tips 886
What to Do Next 887
Configuring IKE Authentication 887
IKE Authentication Methods: Overview 887
Prerequisites 888
Configuring RSA Keys Manually for RSA Encrypted Nonces 888
Configuring Preshared Keys 891
Configuring IKE Mode Configuration 894
About IKE Mode Configuration 894
Restrictions 895
Configuration Examples for an IKE Configuration 896
Creating IKE Policies: Examples 896
Creating 3DES IKE Policies: Example 896
Creating an AES IKE Policy: Example 897
Configuring IKE Authentication: Example 897
Where to Go Next 898
Additional References 898
Related Documents 898
Standards 898
MIBs 899
RFCs 899
Technical Assistance 899
Glossary 900
Feature Information for Configuring IKE for IPSec VPNs 901

Security for VPNs with IPSec

Configuring Security for VPNs with IPSec 905
Contents 905
Prerequisites for Configuring Security for VPNs with IPSec 905
Restrictions for Configuring Security for VPNs with IPSec 906
Information About Configuring Security for VPNs with IPSec 906
Supported Standards 906
Supported Hardware, Switching Paths, and Encapsulation 907

Supported Hardware 908
Supported Switching Paths 910
Supported Encapsulation 910
IPSec Functionality Overview 910
IPSec Traffic Nested to Multiple Peers 912
How to Configure IPSec VPNs 912
Creating Crypto Access Lists 912
Crypto Access List Overview 913
When to Use the permit and deny Keywords in Crypto Access Lists 913
Mirror Image Crypto Access Lists at Each IPSec Peer 914
When to Use the **any** Keyword in Crypto Access Lists 915
What to Do Next 917
Defining Transform Sets: A Combination of Security Protocols and Algorithms 917
Restrictions 917
About Transform Sets 917
What to Do Next 920
Creating Crypto Map Sets 920
Prerequisites 920
About Crypto Maps 920
Load Sharing Among Crypto Maps 921
Crypto Map Guidelines 921
Creating Static Crypto Maps 922
Creating Dynamic Crypto Maps 924
Creating Crypto Map Entries to Establish Manual SAs 931
Applying Crypto Map Sets to Interfaces 933
Redundant Interfaces Sharing the Same Crypto Map 934
Configuration Examples for Configuring an IPSec VPN 935
AES-Based Static Crypto Map: Example 935
Additional References 937
Related Documents 937
Standards 937
MIBs 937
RFCs 937
Technical Assistance 938
Glossary 938
Feature Information for Security for VPNs with IPSec 939

Ability to Disable Extended Authentication for Static IPSec Peers 941

Feature Overview 941

Benefits 942
Restrictions 942
Related Documents 942
Supported Platforms 942
Supported Standards, MIBs, and RFCs 943
Prerequisites 943
Configuration Tasks 944
Disabling Xauth for Static IPSec Peers 944
Verifying Disabled Xauth for Static IPSec Peers 944
Configuration Examples 945
Disabling Xauth for Static IPSec Peers Configuration 945
Command Reference 945

Cisco Easy VPN Remote 947

Contents 948
Prerequisites for Cisco Easy VPN Remote 948
Restrictions for Cisco Easy VPN Remote 949
Information About Cisco Easy VPN Remote 950
Benefits of the Cisco Easy VPN Remote Feature 950
Cisco Easy VPN Remote Overview 950
Modes of Operation 951
Client Mode and Network Extension Mode Scenarios 952
Authentication 954
Using Preshared Keys 955
Using Digital Certificates 955
Using Xauth 955
Web-Based Activation 956
802.1x Authentication 962
Tunnel Activation Options 963
Automatic Activation 963
Manual Activation 963
Traffic-Triggered Activation 963
Dead Peer Detection Stateless Failover Support 964
Backup Server List Local Configuration 964
Backup Server List Auto Configuration 964
Cisco Easy VPN Remote Features 965
Default Inside Interface 966
Multiple Inside Interfaces 966
Multiple Outside Interfaces 967

VLAN Support 967
Multiple Subnet Support 967
NAT Interoperability Support 967
Local Address Support 968
Peer Hostname 968
Proxy DNS Server Support 968
Cisco IOS Firewall Support 969
Easy VPN Remote and Server on the Same Interface 969
Easy VPN Remote and Site to Site on the Same Interface 969
Cisco Easy VPN Remote Web Managers 969
Dead Peer Detection Periodic Message Option 970
Load Balancing 970
Management Enhancements 970
PFS Support 970
Dial Backup 971
How to Configure Cisco Easy VPN Remote 972
Remote Tasks 973
Configuring and Assigning the Easy VPN Remote Configuration 973
Verifying the Cisco Easy VPN Configuration 975
Configuring Save Password 976
Configuring Manual Tunnel Control 977
Configuring Automatic Tunnel Control 979
Configuring Multiple Inside Interfaces 980
Configuring Multiple Outside Interfaces 981
Configuring Multiple Subnet Support 982
Configuring Proxy DNS Server Support 984
Configuring Dial Backup 984
Configuring the DHCP Server Pool 985
Resetting a VPN Connection 985
Monitoring and Maintaining VPN and IKE Events 986
Easy VPN Server Tasks 987
Configuring a Cisco IOS Easy VPN Server 987
Configuring an Easy VPN Server on a VPN 3000 Series Concentrator 987
Configuring an Easy VPN Server on a Cisco PIX Firewall 989
Web Interface Tasks 990
Configuring Web-Based Activation 990
Monitoring and Maintaining Web-Based Activation 990
Using SDM As a Web Manager 994
Troubleshooting the VPN Connection 994
Troubleshooting a VPN Connection Using the Cisco Easy VPN Remote Feature 994

Troubleshooting the Client Mode of Operation 994
Troubleshooting Remote Management 995
Troubleshooting Dead Peer Detection 995
Configuration Examples for Cisco Easy VPN Remote 996
Easy VPN Remote Configuration Examples 996
Client Mode Configuration: Examples 996
Local Address Support for Easy VPN Remote: Example 1002
Network Extension Mode Configuration: Examples 1003
Save Password Configuration: Example 1007
PFS Support: Examples 1008
Dial Backup: Examples 1008
Web-Based Activation: Example 1014
Easy VPN Server Configuration Examples 1015
Cisco Easy VPN Server Without Split Tunneling: Example 1015
Cisco Easy VPN Server Configuration with Split Tunneling: Example 1016
Cisco Easy VPN Server Configuration with Xauth: Example 1018
Easy VPN Server Interoperability Support: Example 1020
Additional References 1021
Related Documents 1021
Standards 1024
MIBs 1024
RFCs 1025
Technical Assistance 1025
Command Reference 1025
Appendix A: Supported Mode Configuration Attributes 1026
Glossary 1027

Crypto Access Check on Clear-Text Packets 1029

Contents 1029
Prerequisites for Crypto Access Check on Clear-Text Packets 1029
Restrictions for Crypto Access Check on Clear-Text Packets 1030
Information About Crypto Access Check on Clear-Text Packets 1030
Crypto Access Check on Clear-Text Packets Overview 1030
Configuration Changes That Are Required for This Feature 1030
Prior to Upgrading 1030
After Upgrading 1031
How ACL Access Checking Worked Prior to This Feature 1031
ACL Checking Behavior After Upgrading to This Feature 1032
Backward Compatibility 1034

How to Configure Crypto Map Access ACLs 1034
Adding or Removing ACLs 1034
Verifying the Configured ACLs 1035
Configuration Examples for Crypto Access Check on Clear-Text Packets 1036
Previous IPSec ACL Configuration: Example 1036
New IPSec ACL Configuration Without Crypto Access ACLs: Example 1037
New IPSec ACL Configuration with Crypto Access ACLs: Example 1037
Additional References 1038
Related Documents 1038
Standards 1038
MIBs 1038
RFCs 1038
Technical Assistance 1039
Command Reference 1039

DF Bit Override Functionality with IPSec Tunnels 1041
Feature Overview 1041
Benefits 1042
Restrictions 1042
Related Documents 1042
Supported Platforms 1042
Supported Standards, MIBs, and RFCs 1043
Prerequisites 1043
Configuration Tasks 1043
Configuring the DF Bit for the Encapsulating Header in Tunnel Mode 1044
Verifying DF Bit Setting 1044
Configuration Examples 1044
DF Bit Setting Configuration Example 1044
Command Reference 1045

Distinguished Name Based Crypto Maps 1047
Feature Overview 1047
Benefits 1047
Restrictions 1048
Related Documents 1048
Supported Platforms 1048
Supported Standards, MIBs, and RFCs 1049
Prerequisites 1049

Configuration Tasks 1049
Configuring DN Based Crypto Maps (authenticated by DN) 1050
Configuring DN Based Crypto Maps (authenticated by hostname) 1050
Applying Identity to DN Based Crypto Maps 1051
Verifying DN Based Crypto Maps 1051
Troubleshooting Tips 1051
Configuration Examples 1051
DN Based Crypto Map Configuration Example 1051
Command Reference 1052

Dynamic Multipoint VPN (DMVPN) 1053

Contents 1054
Prerequisites for Dynamic Multipoint VPN (DMVPN) 1054
Restrictions for Dynamic Multipoint VPN (DMVPN) 1054
Information About Dynamic Multipoint VPN (DMVPN) 1054
Benefits of Dynamic Multipoint VPN (DMVPN) 1055
Feature Design of Dynamic Multipoint VPN (DMVPN) 1055
IPSec Profiles 1056
How to Configure DMVPN 1057
Configure an IPSec Profile 1057
Prerequisites 1057
What to Do Next 1058
Configure the Hub for DMVPN 1059
Configure the Spoke for DMVPN 1060
Verify Dynamic Multipoint VPN (DMVPN) 1063
Configuration Examples for Dynamic Multipoint VPN (DMVPN) Feature 1064
Hub Configuration for DMVPN Example 1064
Spoke Configuration for DMVPN Example 1065
Verify Dynamic Multipoint VPN (DMVPN) Example 1066
Additional References 1068
Related Documents 1068
Standards 1069
MIBs 1069
RFCs 1070
Technical Assistance 1070
Command Reference 1070
Glossary 1071

Easy VPN Remote RSA Signature Support 1073

Contents 1073

Prerequisites for Easy VPN Remote RSA Signature Support 1073

Restrictions for Easy VPN Remote RSA Signature Support 1074

Information About Easy VPN Remote RSA Signature Support 1074

Easy VPN Remote RSA Signature Support Overview 1074

How to Configure RSA Signatures 1074

Configuring Easy VPN Remote RSA Signature Support 1074

Additional References 1075

Related Documents 1075

Standards 1076

MIBs 1076

RFCs 1076

Technical Assistance 1076

Command Reference 1076

Easy VPN Server 1077

Contents 1077

Restrictions for Easy VPN Server 1078

Information About Easy VPN Server 1078

How It Works 1079

RADIUS Support for Group Profiles 1080

For a Cisco Secure Access Control Server 1080

For All Other RADIUS Servers 1082

RADIUS Support for User Profiles 1083

For All Other RADIUS Servers 1083

Supported Protocols 1084

Functions Supported by Easy VPN Server 1084

Mode Configuration Version 6 Support 1084

Xauth Version 6 Support 1085

IKE DPD 1085

Split Tunneling Control 1085

Initial Contact 1085

Group-Based Policy Control 1085

User-Based Policy Control 1085

Session Monitoring for VPN Group Access 1087

How to Configure Easy VPN Server 1087

Enabling Policy Lookup via AAA 1088

Defining Group Policy Information for Mode Configuration Push 1089

Enabling VPN Session Monitoring 1092
Verifying a VPN Session 1093
Applying Mode Configuration and Xauth 1094
Enabling Reverse Route Injection for the Client 1095
Enabling IKE Dead Peer Detection 1096
Configuring RADIUS Server Support 1097
Verifying Easy VPN Server 1097
Configuration Examples for Easy VPN Server 1098
Configuring Cisco IOS for Easy VPN Server: Example 1098
RADIUS Group Profile with IPSec AV Pairs: Example 1100
RADIUS User Profile with IPSec AV Pairs: Example 1100
Backup Gateway with Maximum Logins and Maximum Users: Example 1100
Additional References 1101
Related Documents 1101
Standards 1101
MIBs 1102
RFCs 1102
Technical Assistance 1102
Command Reference 1102
Glossary 1104

Invalid Security Parameter Index Recovery 1105

Contents 1105
Prerequisites for Invalid Security Parameter Index Recovery 1105
Restrictions for Invalid Security Parameter Index Recovery 1106
Information About Invalid Security Parameter Index Recovery 1106
How the Invalid Security Parameter Index Recovery Feature Works 1106
How to Configure Invalid Security Parameter Index Recovery 1107
Configuring Invalid Security Parameter Index Recovery 1107
Verifying an Invalid Security Parameter Index Recovery Configuration 1107
Configuration Examples for Invalid Security Parameter Index Recovery 1114
Invalid Security Parameter Index Recovery: Example 1114
Additional References 1119
Related Documents 1119
Standards 1120
MIBs 1120
RFCs 1120
Technical Assistance 1120
Command Reference 1121

IP Security VPN Monitoring 1123

Contents 1123

Prerequisites for IP Security VPN Monitoring 1124

Restrictions for IP Security VPN Monitoring 1124

Information About IPSec VPN Monitoring 1124

Background: Crypto Sessions 1124

Per-IKE Peer Description 1124

Summary Listing of Crypto Session Status 1125

Syslog Notification for Crypto Session Up or Down Status 1125

IKE and IPSec Security Exchange Clear Command 1125

How to Configure IP Security VPN Monitoring 1126

Adding the Description of an IKE Peer 1126

Verifying Peer Descriptions 1127

Examples 1127

Clearing a Crypto Session 1127

Configuration Examples for IP Security VPN Monitoring 1128

show crypto session Command Output: Examples 1128

Additional References 1129

Related Documents 1129

Standards 1129

MIBs 1129

RFCs 1129

Technical Assistance 1129

Command Reference 1130

IPSec and Quality of Service 1131

Contents 1131

Prerequisites for IPSec and Quality of Service 1131

Restrictions for IPSec and Quality of Service 1132

Information About IPSec and Quality of Service 1132

IPSec and Quality of Service Overview 1132

How to Configure IPSec and Quality of Service 1132

Configuring IPSec and Quality of Service 1132

Verifying IPSec and Quality of Service Sessions 1133

Troubleshooting Tips 1134

Configuration Examples for IPSec and Quality of Service 1134

QoS Policy Applied to Two Groups of Remote Users: Example 1134

show crypto isakmp profile Command: Example 1136

show crypto ipsec sa Command: Example 1136
Additional References 1137
Related Documents 1137
Standards 1137
MIBs 1137
RFCs 1138
Technical Assistance 1138
Command Reference 1138

IPSec Anti-Replay Window: Expanding and Disabling 1139

Contents 1139
Prerequisites for IPSec Anti-Replay Window: Expanding and Disabling 1140
Information About IPSec Anti-Replay Window: Expanding and Disabling 1140
IPSec Anti-Replay Window 1140
How to Configure IPSec Anti-Replay Window: Expanding and Disabling 1140
Configuring IPSec Anti-Replay Window: Expanding and Disabling Globally 1140
Configuring IPSec Anti-Replay Window: Expanding and Disabling on a Crypto Map 1141
Troubleshooting Tips 1142
Configuration Examples for IPSec Anti-Replay Window: Expanding and Disabling 1143
Global Expanding and Disabling of an Anti-Replay Window: Example 1143
Expanding and Disabling of an Anti-Replay Window for a Particular Crypto Map, Dynamic Crypto Map, or Crypto Profile: Example 1144
Additional References 1145
Related Documents 1145
Standards 1145
MIBs 1145
RFCs 1146
Technical Assistance 1146
Command Reference 1146

IPSec Dead Peer Detection Periodic Message Option 1147

Contents 1147
Prerequisites for IPSec Dead Peer Detection Periodic Message Option 1148

Restrictions for IPSec Dead Peer Detection Periodic Message Option 1148
Information About IPSec Dead Peer Detection Periodic Message Option 1148
How DPD and Cisco IOS Keepalive Features Work 1148
Using the IPSec Dead Peer Detection Periodic Message Option 1149
Using DPD and Cisco IOS Keepalive Features with Multiple Peers in the Crypto Map 1149
Using DPD in an Easy VPN Remote Configuration 1149
How to Configure IPSec Dead Peer Detection Periodic Message Option 1149
Configuring a Periodic DPD Message 1150
Configuring DPD and Cisco IOS Keepalives with Multiple Peers in the Crypto Map 1150
Configuring DPD for an Easy VPN Remote 1151
Verifying That DPD Is Enabled 1152
Configuration Examples for IPSec Dead Peer Detection Periodic Message Option 1153
Site-to-Site Setup with Periodic DPD Enabled: Example 1153
Easy VPN Remote with DPD Enabled: Example 1154
Verifying DPD Configuration Using the debug crypto isakmp Command: Example 1154
DPD and IOS Keepalives Used in Conjunction with Multiple Peers in a Crypto Map: Example 1157
DPD Used in Conjunction with Multiple Peers for an Easy VPN Remote: Example 1157
Additional References 1158
Related Documents 1158
Standards 1158
MIBs 1158
RFCs 1158
Technical Assistance 1159
Command Reference 1159

IPSec NAT Transparency 1161

Contents 1162
Restrictions for IPSec NAT Transparency 1162
Information About IPSec NAT Transparency 1162
Benefit of IPSec NAT Transparency 1163
Feature Design of IPSec NAT Traversal 1163
IKE Phase 1 Negotiation: NAT Detection 1163
IKE Phase 2 Negotiation: NAT Traversal Decision 1164
UDP Encapsulation of IPSec Packets for NAT Traversal 1164

UDP Encapsulated Process for Software Engines: Transport Mode and Tunnel Mode ESP Encapsulation 1166
NAT Keepalives 1166
How to Configure NAT and IPSec 1167
Configuring NAT Traversal 1167
Disabling NAT Traversal 1167
Configuring NAT Keepalives 1168
Verifying IPSec Configuration 1168
Configuration Examples for IPSec and NAT 1169
NAT Keepalives Configuration Example 1169
Additional References 1169
Related Documents 1170
Standards 1170
MIBs 1170
RFCs 1171
Technical Assistance 1171
Command Reference 1171
Glossary 1172

IPSec Preferred Peer 1173

Contents 1173
Prerequisites for IPSec Preferred Peer 1174
Restrictions for IPSec Preferred Peer 1174
Information About IPSec Preferred Peer 1174
IPSec 1174
Dead Peer Detection 1175
Default Peer Configuration 1175
Idle Timers 1176
IPSec Idle-Timer Usage with Default Peer 1176
Peers on Crypto Maps 1176
How to Configure IPSec Preferred Peer 1176
Configuring a Default Peer 1177
Configuring the Idle Timer 1178
Configuration Examples for IPSec Preferred Peer 1179
Configuring a Default Peer: Example 1179
Configuring the IPSec Idle Timer: Example 1179
Additional References 1179
Related Documents 1179
Standards 1179

MIBs 1180
RFCs 1180
Technical Assistance 1180
Command Reference 1180
Glossary 1181

IPSec Security Association Idle Timers 1183

Contents 1184
Prerequisites for IPSec Security Association Idle Timers 1184
Information About IPSec Security Association Idle Timers 1184
Lifetimes for IPSec Security Associations 1184
IPSec Security Association Idle Timers 1184
Benefits of IPSec Security Association Idle Timers 1185
How to Configure IPSec Security Association Idle Timers 1185
Configuring the IPSec SA Idle Timer Globally 1185
Configuring the IPSec SA Idle Timer per Crypto Map 1186
Configuration Examples for IPSec Security Association Idle Timers 1187
Configuring the IPSec SA Idle Timer Globally Example 1187
Configuring the IPSec SA Idle Timer per Crypto Map Example 1187
Additional References 1187
Related Documents 1188
Standards 1188
MIBs 1188
RFCs 1189
Technical Assistance 1189
Command Reference 1189

IPSec—SNMP Support 1191

Feature Overview 1192
Benefits 1192
Restrictions 1192
Related Features and Technologies 1193
Related Documents 1193
Supported Platforms 1193
Supported Standards, MIBs, and RFCs 1195
Configuration Tasks 1195
Enabling IPSec SNMP Notifications 1196
Configuring IPSec Failure History Table Size 1196
Configuring IPSec Tunnel History Table Size 1196

Verifying IPSec MIB Configuration 1196
Monitoring and Maintaining IPSec MIB 1197
Configuration Examples 1197
Enabling IPSec Notifications Examples 1197
Specifying History Table Size Examples 1198
Command Reference 1198
Glossary 1199

IPSec Virtual Tunnel Interface 1201

Contents 1201
Restrictions for IPSec Virtual Tunnel Interface 1202
Information About IPSec Virtual Tunnel Interfaces 1202
Routing with IPSec Virtual Tunnel Interfaces 1202
Traffic Encryption with the IPSec Virtual Tunnel Interface 1203
IPSec Packet Flow 1203
Dynamic Virtual Tunnel Interfaces 1205
Profile Definitions and Policy Define the Dynamic Virtual Tunnel Interface Life Cycle 1206
How to Configure IPSec Virtual Tunnels 1206
Configuring IPSec Static Tunnels 1206
Configuring Dynamic Virtual Tunnel Interfaces 1208
Configuration Examples for IPSec Virtual Tunnel Interfaces 1210
Static Virtual Tunnel Interface with IPSec: Example 1210
Verifying the Results for IPSec Virtual Tunnel Interface Example 1211
Dynamic Virtual Tunnel Interface with IPSec for Simple Hub-and-Spoke Configuration: Example 1213
VRF-Aware Ipsec with Dynamic VTI: Example 1213
QoS Service Policy Per Instance with Dynamic VTI: Example 1213
Additional References 1214
Related Documents 1214
Standards 1214
MIBs 1214
RFCs 1214
Technical Assistance 1215
Command Reference 1215

IPSec VPN Accounting 1217

Contents 1217
Prerequisites for IPSec VPN Accounting 1218
Information About IPSec VPN Accounting 1218

RADIUS Accounting 1218
RADIUS Start Accounting 1218
RADIUS Stop Accounting 1219
RADIUS Update Accounting 1220
IKE and IPSec Subsystem Interaction 1220
Accounting Start 1220
Accounting Stop 1221
Accounting Updates 1222
How to Configure IPSec VPN Accounting 1222
Configuring IPSec VPN Accounting 1223
Prerequisites 1223
Configuring Accounting Updates 1226
Prerequisites 1226
Troubleshooting for IPSec VPN Accounting 1227
Configuration Examples for IPSec VPN Accounting 1228
Accounting and ISAKMP-Profile Example 1228
Accounting Without ISAKMP Profiles Example 1230
Additional References 1232
Related Documents 1232
Standards 1232
MIBs 1233
RFCs 1233
Technical Assistance 1233
Command Reference 1233
Glossary 1235

IPSec VPN High Availability Enhancements 1237

Feature Overview 1237
Reverse Route Injection 1237
Hot Standby Router Protocol and IPSec 1238
Benefits 1239
Related Documents 1240
Supported Platforms 1240
Supported Standards, MIBs, and RFCs 1241
Configuration Tasks 1242
Configuring Reverse Route Injection on a Dynamic Crypto Map 1242
Configuring Reverse Route Injection on a Static Crypto Map 1242
Configuring HSRP with IPSec 1243
Verifying VPN IPSec Crypto Configuration 1244

Configuration Examples 1244
Reverse Route Injection on a Dynamic Crypto Map Example 1244
Reverse Route Injection on a Static Crypto Map Example 1245
HSRP and IPSec Example 1245
Command Reference 1246

L2TP Security 1247
Feature Overview 1247
Benefits 1248
Related Features and Technologies 1248
Related Documents 1248
Supported Platforms 1248
Supported Standards, MIBs, and RFCs 1249
Prerequisites 1250
Configuration Tasks 1250
Configuring NAS-Initiated VPDN Tunneling with L2TP Security 1251
Configuring the Client 1251
Configuring the LAC 1251
Configuring the LNS 1254
Configuring Client-Initiated VPDN Tunneling with L2TP Security 1256
Verifying Session Establishment 1258
Configuration Examples 1259
Configuring IPSec Protection of LAC-Initiated L2TP Tunnels Example 1259
Configuring IPSec Protection of Client-Initiated L2TP Tunnels Example 1261
Command Reference 1262

Low Latency Queueing (LLQ) for IPSec Encryption Engines 1263
Feature Overview 1263
Benefits 1264
Restrictions 1264
Related Features and Technologies 1264
Related Documents 1265
Supported Platforms 1265
Determining Platform Support Through Cisco Feature Navigator 1265
Availability of Cisco IOS Software Images 1266
Supported Standards, MIBs, and RFCs 1266
Prerequisites 1266
Configuration Tasks 1267
Defining Class Maps 1267

Configuring Class Policy in the Policy Map 1267
Configuring Class Policy for a Priority Queue 1268
Configuring Class Policy Using a Specified Bandwidth 1268
Configuring the Class-Default Class Policy 1269
Attaching the Service Policy 1269
Verifying Configuration of Policy Maps and Their Classes 1269
Monitoring and Maintaining LLQ for IPSec Encryption Engines 1270
Configuration Examples 1270
LLQ for IPSec Encryption Engines Example 1270
Command Reference 1271
Glossary 1271

L2TP—IPSec Support for NAT and PAT Windows Clients 1273

Contents 1273
Prerequisites for L2TP—IPSec Support for NAT and PAT Windows Clients 1274
Restrictions for L2TP—IPSec Support for NAT and PAT Windows Clients 1274
Information About L2TP—IPSec Support for NAT and PAT Windows Clients 1274
How L2TP—IPSec Support for NAT and PAT Windows Clients Works 1274
How to Enable L2TP—IPSec Support for NAT and PAT Windows Clients 1276
Enabling L2TP—IPSec Support 1276
Configuration Examples for L2TP—IPSec Support for NAT and PAT Windows Clients 1278
Dynamic Map Configuration: Example 1278
Additional References 1280
Related Documents 1280
Standards 1281
MIBs 1281
RFCs 1281
Technical Assistance 1281
Command Reference 1282

Pre-Fragmentation for IPSec VPNs 1283

Feature Overview 1283
Benefits 1284
Restrictions 1284
Supported Platforms 1285
Supported Standards, MIBs, and RFCs 1287
Configuration Tasks 1287
Configuring Pre-Fragmentation For IPSec VPNs 1287
Verifying Pre-Fragmentation For IPSec VPNs 1288

Configuration Examples 1289
Enabling Pre-Fragmentation For IPSec VPNs Example 1289
Command Reference 1290

Real-Time Resolution for IPSec Tunnel Peer 1291

Contents 1291
Restrictions for Real-Time Resolution for IPSec Tunnel Peer 1291
Information About Real-Time Resolution for IPSec Tunnel Peer 1292
Benefits of Real-Time Resolution Via Secure DNS 1292
How to Configure Real-Time Resolution 1292
Configuring Real-Time Resolution for IPSec Peers 1292
Prerequisites 1292
Troubleshooting Tips 1294
What to Do Next 1294
Configuration Examples for Real-Time Resolution 1294
Configuring Real-Time Resolution for an IPSec Peer: Example 1294
Additional References 1295
Related Documents 1295
Standards 1295
MIBs 1295
RFCs 1296
Technical Assistance 1296
Command Reference 1296

Reverse Route Injection 1297

Contents 1298
Prerequisites for Reverse Route Injection 1298
Restrictions for Reverse Route Injection 1298
Information About Reverse Route Injection 1298
Reverse Route Injection 1298
Reverse Route with Remote Peer Option 1299
Enhancements to Reverse Route Injection 1299
How to Configure Reverse Route Injection 1300
Configuring RRI for Cisco IOS Releases Before 12.3(14)T 1300
Configuring RRI Under a Static Crypto Map 1300
Configuring RRI Under a Dynamic Map Template 1301
Configuring RRI with Enhancements 1302
Configuring RRI with Enhancements Under a Static Crypto Map 1302
Configuring RRI with Enhancements Under a Dynamic Map Template 1303

Troubleshooting Tips 1304
Configuration Examples for Reverse Route Injection 1304
Configuring RRI Prior to Cisco IOS Release 12.3(14)T: Examples 1304
Configuring RRI When Crypto ACLs Exist: Example 1305
Configuring RRI When Two Routes Are Created, One for the Remote Endpoint and One for Route Recursion: Example 1305
Configuring RRI with the Enhancements Added in Cisco IOS Release 12.3(14)T: Examples 1305
Configuring RRI When Crypto ACLs Exist: Example 1306
Configuring RRI with Route Tags: Example 1306
Configuring RRI for One Route to the Remote Proxy Via a User-Defined Next Hop: Example 1306
Additional References 1307
Related Documents 1307
Standards 1307
MIBs 1307
RFCs 1307
Technical Assistance 1308
Command Reference 1308

SafeNet IPSec VPN Client Support 1309

Contents 1309
Prerequisites for SafeNet IPSec VPN Client Support 1309
Restrictions for SafeNet IPSec VPN Client Support 1310
Information About SafeNet IPSec VPN Client Support 1310
ISAKMP Profile and ISAKMP Keyring Configurations: Background 1310
Local Termination Address or Interface 1310
Benefit of SafeNet IPSec VPN Client Support 1310
How to Configure SafeNet IPSec VPN Client Support 1311
Limiting an ISAKMP Profile to a Local Termination Address or Interface 1311
Limiting a Keyring to a Local Termination Address or Interface 1312
Monitoring and Maintaining SafeNet IPSec VPN Client Support 1313
Examples 1314
Troubleshooting SafeNet IPSec VPN Client Support 1315
Configuration Examples for SafeNet IPSec VPN Client Support 1315
ISAKMP Profile Bound to a Local Interface: Example 1315
ISAKMP Keyring Bound to a Local Interface: Example 1315
ISAKMP Keyring Bound to a Local IP Address: Example 1316
ISAKMP Keyring Bound to an IP Address and Limited to a VRF: Example 1316
Additional References 1316

Related Documents 1316
Standards 1316
MIBs 1317
RFCs 1317
Technical Assistance 1317
Command Reference 1317

Stateful Failover for IPSec 1319

Contents 1319
Prerequisites for Stateful Failover for IPSec 1320
Restrictions for Stateful Failover for IPSec 1320
Information About Stateful Failover for IPSec 1321
Supported Deployment Scenarios: Stateful Failover for IPSec 1321
IPSec Stateful Failover for Remote Access Connections 1323
How to Use Stateful Failover for IPSec 1324
Enabling HSRP: IP Redundancy and a Virtual IP Address 1324
Prerequisites for Spanning Tree Protocol and HSRP Stability 1324
Restrictions 1324
Troubleshooting Tips 1327
Examples 1327
What to Do Next 1327
Enabling SSO 1327
SSO: Interacting with IPSec and IKE 1327
Prerequisites 1327
Troubleshooting Tips 1330
Examples 1330
What to Do Next 1331
Configuring Reverse Route Injection on a Crypto Map 1331
Configuring RRI on Dynamic Crypto Map 1331
Configuring RRI on a Static Crypto Map 1332
Examples 1333
What to Do Next 1333
Enabling Stateful Failover for IKE and IPSec 1333
Enabling Stateful Failover for IKE 1333
Enabling Stateful Failover for IPSec 1333
Enabling Stateful Failover for Tunnel Protection 1335
What to Do Next 1336
Protecting SSO Traffic 1336
Examples 1338

Managing and Verifying High Availability Information 1338
Managing Anti-Replay Intervals 1339
Examples 1339
Managing and Verifying HA Configurations 1340
Examples 1341
Configuration Examples for Stateful Failover 1345
Configuring IPSec Stateful Failover: Example 1345
Configuring IPSec Stateful Failover for an Easy VPN Server: Example 1349
Additional References 1354
Related Documents 1354
Standards 1354
MIBs 1354
RFCs 1355
Technical Assistance 1355
Command Reference 1355

VRF-Aware IPSec 1357

Contents 1357
Restrictions for VRF-Aware IPSec 1358
Information About VRF-Aware IPSec 1358
VRF Instance 1358
MPLS Distribution Protocol 1358
VRF-Aware IPSec Functional Overview 1358
Packet Flow into the IPSec Tunnel 1359
Packet Flow from the IPSec Tunnel 1359
How to Configure VRF-Aware IPSec 1360
Configuring Crypto Keyrings 1360
Configuring ISAKMP Profiles 1362
Restriction 1362
What to Do Next 1366
Configuring an ISAKMP Profile on a Crypto Map 1366
Prerequisites 1366
Configuring to Ignore Extended Authentication During IKE Phase 1 Negotiation 1367
Verifying VRF-Aware IPSec 1367
Clearing Security Associations 1368
Troubleshooting VRF-Aware IPSec 1369
Debug Examples for VRF-Aware IPSec 1369
Configuration Examples for VRF-Aware IPSec 1377
Static IPSec-to-MPLS VPN Example 1378

IPSec-to-MPLS VPN Using RSA Encryption Example 1379
IPSec-to-MPLS VPN with RSA Signatures Example 1381
IPSec Remote Access-to-MPLS VPN Example 1382
Upgrade from Previous Versions of the Cisco Network-Based IPSec VPN Solution 1383
Site-to-Site Configuration Upgrade 1384
Remote Access Configuration Upgrade 1385
Combination Site-to-Site and Remote Access Configuration Upgrade 1387
Additional References 1389
Related Documents 1389
Standards 1390
MIBs 1390
RFCs 1390
Technical Assistance 1391
Command Reference 1391
Glossary 1393

PART 5: PKI

Implementing and Managing PKI Features Roadmap 1397

Cisco IOS PKI Overview: Understanding and Planning a PKI 1403
Contents 1403
Information About Cisco IOS PKI 1403
What Is Cisco IOS PKI? 1404
RSA Keys Overview 1405
What Are CAs? 1405
Hierarchical PKI: Multiple CAs 1405
Certificate Enrollment: How It Works 1406
Certificate Enrollment Via Secure Device Provisioning 1407
Certificate Revocation: Why It Occurs 1407
Planning for a PKI 1407
Where to Go Next 1408
Additional References 1408
Related Documents 1408
Standards 1408
MIBs 1408
RFCs 1409
Technical Assistance 1409
Glossary 1409

Deploying RSA Keys Within a PKI 1411

Contents 1411

Prerequisites for Configuring RSA Keys for a PKI 1411

Information About RSA Keys Configuration 1412

RSA Keys Overview 1412

Usage RSA Keys Versus General-Purpose RSA Keys 1412

Reasons to Store Multiple RSA Keys on a Router 1412

Benefits of Exportable RSA Keys 1413

Passphrase Protection While Importing and Exporting RSA Keys 1413

How to Set Up and Deploy RSA Keys Within a PKI 1414

Generating an RSA Key Pair 1414

What to Do Next 1415

Generating and Storing Multiple RSA Key Pairs 1415

Prerequisites 1415

Exporting and Importing RSA Keys 1416

Exporting and Importing RSA Keys in PKCS12 Files 1416

Exporting and Importing RSA Keys in PEM-Formatted Files 1418

Encrypting and Locking Private Keys on a Router 1419

Prerequisites 1420

Restrictions for Encrypting and Locking Private Keys 1420

Removing RSA Key Pair Settings 1422

Configuration Examples for RSA Key Pair Deployment 1423

Generating and Specifying RSA Keys: Example 1423

Exporting and Importing RSA Keys: Examples 1423

Exporting and Importing RSA Keys in PKCS12 Files: Example 1424

Generating, Exporting, Importing, and Verifying RSA Keys in PEM Files: Example 1424

Exporting Router RSA Key Pairs and Certificates from PEM Files: Example 1425

Importing Router RSA Key Pairs and Certificate from PEM Files: Example 1427

Encrypting and Locking Private Keys on a Router: Examples 1427

Configuring and Verifying an Encrypted Key: Example 1427

Configuring and Verifying a Locked Key: Example 1428

Where to Go Next 1428

Additional References 1428

Related Documents 1428

Technical Assistance 1429

Feature Information for RSA Keys Within a PKI 1429

Configuring Authorization and Revocation of Certificates in a PKI 1433

Contents 1433

Prerequisites for Authorization and Revocation of Certificates 1433
Information About Authorization and Revocation of Certificates 1434
PKI Authorization 1434
PKI and AAA Server Integration for Certificate Status 1435
RADIUS or TACACS+: Choosing a AAA Server Protocol 1435
Attribute-Value Pairs for PKI and AAA Server Integration 1435
CRLs or OCSP Server: Choosing a Certificate Revocation Mechanism 1436
What Is a CRL? 1437
What Is OCSP? 1438
When to Use Certificate-Based ACLs for Authorization or Revocation 1438
Ignore Revocation Checks Using a Certificate-Based ACL 1439
How to Configure Authorization and Revocation of Certificates for Your PKI 1440
Configuring PKI Integration with a AAA Server 1440
Restrictions When Using the Entire Subject Name for PKI Authorization 1440
Troubleshooting Tips 1443
Configuring a Revocation Mechanism for Cisco IOS Certificate Status Checking 1444
The revocation-check Command 1444
Prerequisites 1444
Restrictions 1444
Examples 1446
Overriding Certificate Revocation and Authorization Settings 1447
Overview: Configuring Certificate-Based ACLs to Ignore Revocation Checks 1447
Manually Overriding CDPs in a Certificate 1447
Prerequisites 1447
Troubleshooting Tips 1450
Configuration Examples for Setting Up Authorization and Revocation of Certificates 1450
Configuring and Verifying PKI AAA Authorization: Examples 1450
Router Configuration: Example 1451
Debug of a Successful PKI AAA Authorization: Example 1453
Debugs of a Failed PKI AAA Authorization: Example 1454
Configuring a Revocation Mechanism: Examples 1455
Configuring an OCSP Server: Example 1455
Specifying a CRL and Then an OCSP Server: Example 1455
Specifying an OCSP Server: Example 1455
Configuring a Hub Router at a Central Site for Certificate Revocation Checks: Example 1456
Additional References 1460
Related Documents 1460
Technical Assistance 1460
Feature Information for Certificate Authorization and Revocation 1460

Configuring Certificate Enrollment for a PKI 1465

Contents 1465

Prerequisites for PKI Certificate Enrollment 1465

Information About Certificate Enrollment for a PKI 1466

What Are CAs? 1466

Hierarchical PKI: Multiple CAs 1466

Authentication of the CA 1467

Supported Certificate Enrollment Methods 1467

Registration Authorities 1468

Automatic Certificate Enrollment 1468

Certificate Enrollment Profiles 1469

How to Configure Certificate Enrollment for a PKI 1469

Configuring Certificate Enrollment or Autoenrollment 1469

Prerequisites for Autoenrollment 1469

Restrictions for Autoenrollment 1470

Configuring Manual Certificate Enrollment 1473

PEM-Formatted Files for Certificate Enrollment Requests 1473

Restrictions for Manual Certificate Enrollment 1474

Configuring Cut-and-Paste Certificate Enrollment 1474

Configuring TFTP Certificate Enrollment 1476

Configuring a Persistent Self-Signed Certificate for Enrollment via SSL 1478

Persistent Self-Signed Certificates Overview 1478

Restrictions 1478

Configuring a Trustpoint and Specifying Self-Signed Certificate Parameters 1478

Enabling the HTTPS Server 1480

Prerequisites 1480

Configuring a Certificate Enrollment Profile for Enrollment or Reenrollment 1481

Prerequisites 1481

Restrictions 1482

What to Do Next 1484

Configuration Examples for PKI Certificate Enrollment Requests 1485

Configuring Autoenrollment: Example 1485

Configuring Certificate Autoenrollment with Key Rollover: Example 1486

Configuring Manual Certificate Enrollment with Key Rollover: Example 1486

Creating and Verifying a Persistent Self-Signed Certificate: Example 1486

Configuring Direct HTTP Enrollment: Example 1488

Additional References 1489

Related Documents 1489

Technical AssistanceAssistance 1489

Feature Information for PKI Certificate Enrollment 1489

Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI 1493

Contents 1493

Prerequisites for Setting Up SDP for Enrollment in a PKI 1494

Information About Setting Up SDP for Enrollment in a PKI 1494

SDP Overview 1494

How SDP Works 1495

SDP Phase One—Welcome 1495

SDP Phase Two—Introduction 1496

SDP Phase Three—Completion 1497

How SDP Uses an External AAA Database 1498

Authentication and Authorization Lists for SDP 1498

Authentication and Authorization Lists for an Administrative Introducer 1499

How to Set Up SDP for a PKI 1499

Enabling the SDP Petitioner 1500

Prerequisites 1500

Troubleshooting Tips 1501

What to Do Next 1501

Enabling the SDP Registrar and Adding AAA Lists to the Server 1501

Prerequisites 1502

Restrictions 1502

The template config Command 1502

Examples 1504

Enabling the SDP Registrar for Certificate-Based Authorization 1504

Prerequisites 1504

Restrictions 1505

Configuring an Administrative Introducer 1506

Prerequisites 1506

Restrictions 1506

Examples 1507

Configuration Examples for Setting up a PKI via SDP 1509

Verifying the SDP Registrar: Example 1509

Verifying the SDP Petitioner: Example 1512

Adding AAA Lists to a RADIUS or TACACS+ Server: Examples 1515

TACACS+ AAA Server Database: Example 1515

RADIUS AAA Server Database: Example 1515

AAA List on a TACACS+ and a RADIUS AAA Server: Example 1515

Configuration Template File: Example 1516

Configuring the Petitioner and Registrar for Certificate-Based Authentication: Example 1516

Configuring an Administrative Introducer Using Authentication and Authorization Lists: Example 1517
Additional References 1517
Related Documents 1517
Technical Assistance 1517
Feature Information for SDP in a PKI 1518

Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment 1521

Contents 1521
Prerequisites for Configuring a Cisco IOS CS 1522
Restrictions for Configuring a Cisco IOS Certificate Server 1522
Information About Cisco IOS Certificate Servers 1522
RSA Key Pair and Certificate of the Certificate Server 1523
How the CA Certificate and CA Key Are Automatically Archived 1523
Trustpoint of the Certificate Server 1524
Certificate Revocation Lists (CRLs) 1524
Certificate Server Error Conditions 1525
Certificate Enrollment Using a Certificate Server 1525
SCEP Reenrollment 1526
How to Set Up and Deploy a Cisco IOS Certificate Server 1526
Generating and Exporting a Certificate Server RSA Key Pair 1526
Configuring a Certificate Server 1528
Examples 1529
What to Do Next 1529
Configuring Certificate Server Functionality 1530
Certificate Server Default Values and Recommended Values 1530
Examples 1532
Configuring a Proxy to Offload the Root Certificate Server 1533
Configuring a Certificate Server to Run in RA Mode by Configuring the RA Mode Certificate Server 1533
Configuring the Root Certificate Server to Delegate Enrollment Tasks to the RA Mode Certificate Server 1535
Configuring a Subordinate Certificate Server 1536
Maintaining, Verifying, and Troubleshooting the Certificate Server, Certificates, and the CA 1541
Managing the Enrollment Request Database 1542
Removing Requests from the Enrollment Request Database 1543
Deleting a Certificate Server 1544
Verifying and Troubleshooting Certificate Server, Certificate, and CA Status 1545
Configuration Examples for Using a Certificate Server 1546

Removing Enrollment Requests from the Enrollment Request Database: Examples 1546
Autoarchiving the Certificate Server Root Keys: Examples 1547
Restoring a Certificate Server from Certificate Server Backup Files: Examples 1549
RA Mode Certificate Server: Examples 1551
Subordinate Certificate Server: Example 1553
Root Certificate Server Differentiation: Example 1554
Show Output for a Subordinate Certificate Server: Example 1555
Where to Go Next 1555
Additional References 1555
Related Documents 1555
Technical Assistance 1556
Feature Information for the Cisco IOS Certificate Server 1556

Storing PKI Credentials 1559

Contents 1559
Prerequisites for Storing PKI Credentials 1559
Restrictions for Storing PKI Credentials 1560
Information About Storing PKI Credentials 1560
How a USB eToken Works 1560
Benefits of USB eTokens 1561
How to Configure PKI Storage 1561
Setting Up and Using USB eTokens on Cisco Routers 1561
Storing the Configuration on an External USB eToken 1562
Accessing and Setting Up the eToken 1562
Setting Administrative Functions on the eToken 1564
Troubleshooting USB eTokens 1566
The show file systems Command 1566
The show usb device Command 1567
The show usb controllers Command 1568
The dir Command 1570
Configuration Examples for PKI Storage 1571
Logging Into an eToken and Saving RSA Keys to the eToken: Example 1571
Additional References 1573
Related Documents 1573
Technical Assistance 1573
Feature Information for Storing PKI Credentials 1573

PART 6: OTHER SECURITY FEATURES

Neighbor Router Authentication: Overview and Guidelines 1577

In This Chapter 1577

About Neighbor Authentication 1577

Benefits of Neighbor Authentication 1577

Protocols That Use Neighbor Authentication 1578

When to Configure Neighbor Authentication 1578

How Neighbor Authentication Works 1578

Plain Text Authentication 1579

MD5 Authentication 1579

Key Management (Key Chains) 1580

Finding Neighbor Authentication Configuration Information 1581

Configuring IP Security Options 1583

In This Chapter 1583

IPSO Configuration Task List 1583

Configuring Basic IP Security Options 1584

Enabling IPSO and Setting the Security Classifications 1584

Specifying How IP Security Options Are Processed 1584

Configuring Extended IP Security Options 1585

Configuring Global Default Settings 1586

Attaching ESOs to an Interface 1586

Attaching AESOs to an Interface 1586

Configuring the DNSIX Audit Trail Facility 1586

Enabling the DNSIX Audit Trail Facility 1587

Specifying Hosts to Receive Audit Trail Messages 1587

Specifying Transmission Parameters 1587

IPSO Configuration Examples 1588

Example 1 1588

Example 2 1589

Example 3 1589

Unicast Reverse Path Forwarding

Configuring Unicast Reverse Path Forwarding 1593

In This Chapter 1593

About Unicast Reverse Path Forwarding 1593

How Unicast RPF Works 1594

Access Control Lists and Logging 1595
Per-Interface Statistics 1595
Implementing Unicast RPF 1597
Security Policy and Unicast RPF 1598
Where to Use Unicast RPF 1598
Routing Table Requirements 1601
Where Not to Use Unicast RPF 1601
Unicast RPF with BOOTP and DHCP 1602
Restrictions 1602
Related Features and Technologies 1603
Prerequisites to Configuring Unicast RPF 1604
Unicast RPF Configuration Task List 1604
Configuring Unicast RPF 1604
Verifying Unicast RPF 1606
Troubleshooting Tips 1606
HSRP Failure 1606
Dropped Boot Requests 1606
Monitoring and Maintaining Unicast RPF 1607
Unicast RPF Configuration Examples 1608
Unicast RPF on a Leased-Line Aggregation Router Example 1608
Unicast RPF on the Cisco AS5800 Using Dialup Ports Example 1608
Unicast RPF with Inbound and Outbound Filters Example 1609
Unicast RPF with ACLs and Logging Example 1609

Secure Shell

Configuring Secure Shell 1613
In This Chapter 1613
About Secure Shell 1613
How SSH Works 1614
SSH Server 1614
SSH Integrated Client 1614
Restrictions 1614
Related Features and Technologies 1615
Prerequisites to Configuring SSH 1615
SSH Configuration Task List 1616
Configuring SSH Server 1616
Verifying SSH 1617
Troubleshooting Tips 1618

Monitoring and Maintaining SSH 1618
SSH Configuration Examples 1618
SSH on a Cisco 7200 Series Router Example 1619
SSH on a Cisco 7500 Series Router Example 1620
SSH on a Cisco 1200 Gigabit Switch Router Example 1622

Reverse SSH Enhancements 1625
Contents 1625
Prerequisites for Reverse SSH Enhancements 1625
Restrictions for Reverse SSH Enhancements 1626
Information About Reverse SSH Enhancements 1626
Reverse Telnet 1626
Reverse SSH 1626
How to Configure Reverse SSH Enhancements 1626
Configuring Reverse SSH for Console Access 1626
Configuring Reverse SSH for Modem Access 1628
Troubleshooting Reverse SSH on the Client 1630
Troubleshooting Reverse SSH on the Server 1630
Configuration Examples for Reverse SSH Enhancements 1631
Reverse SSH Console Access: Example 1631
Reverse SSH Modem Access: Example 1632
Additional References 1632
Related Documents 1632
Standards 1632
MIBs 1633
RFCs 1633
Technical Assistance 1633
Command Reference 1633

Secure Copy 1635
Contents 1635
Prerequisites for Secure Copy 1635
Information About Secure Copy 1636
How SCP Works 1636
How to Configure SCP 1636
Configuring SCP 1636
Verifying SCP 1637
Troubleshooting SCP 1638
Configuration Examples for Secure Copy 1638

SCP Server-Side Configuration Using Local Authentication: Example 1639
SCP Server-Side Configuration Using Network-Based Authentication: Example 1639
Additional References 1639
Related Documents 1639
Standards 1639
MIBs 1640
RFCs 1640
Technical Assistance 1640
Command Reference 1640
Glossary 1641

Secure Shell Version 2 Support 1643

Contents 1643
Prerequisites for Secure Shell Version 2 Support 1644
Restrictions for Secure Shell Version 2 Support 1644
Information About Secure Shell Version 2 Support 1644
Secure Shell Version 2 1644
How to Configure Secure Shell Version 2 Support 1645
Configuring a Router for SSH Version 2 Using a Host Name and Domain Name 1645
Configuring a Router for SSH Version 2 Using RSA Key Pairs 1646
Starting an Encrypted Session with a Remote Device 1648
Troubleshooting Tips 1648
Verifying the Status of the Secure Shell Connection Using the show ssh Command 1649
Examples 1649
Verifying the Secure Shell Status Using the show ip ssh Command 1650
Examples 1650
Monitoring and Maintaining Secure Shell Version 2 1651
Example 1651
Configuration Examples for Secure Shell Version 2 Support 1654
Configuring Secure Shell Version 1: Example 1654
Configuring Secure Shell Version 2: Example 1654
Configuring Secure Shell Versions 1 and 2: Example 1654
Starting an Encrypted Session with a Remote Device: Example 1654
Where to Go Next 1654
Additional References 1655
Related Documents 1655
Standards 1655
MIBs 1655
RFCs 1655

Technical Assistance 1656
Command Reference 1656

SSH Terminal-Line Access 1657
Feature Overview 1657
Benefits 1658
Restrictions 1658
Related Documents 1658
Supported Platforms 1658
Supported Standards, MIBs, and RFCs 1659
Prerequisites 1659
Configuration Tasks 1659
Configuring SSH Terminal-Line Access 1660
Verifying SSH Terminal-Line Access 1661
Configuration Examples 1661
SSH Terminal-Line Access Configuration Example 1661
SSH Terminal-Line Access for a Console (Serial Line) Ports Configuration Example 1661
Command Reference 1662

802.1X Authentication Services

Remote Site IEEE 802.1X Local Authentication Service 1665
Contents 1665
Prerequisites for Configuring Remote Site IEEE 802.1X Local Authentication Service 1666
Restrictions for Configuring Remote Site IEEE 802.1X Local Authentication Service 1666
Information About Configuring Remote Site IEEE 802.1x Local Authentication Service 1666
How to Configure Remote Site IEEE 802.1X Local Authentication Service 1668
Configuring the Local Authentication Server 1668
Configuring User Groups on the Local Authentication Server 1669
Unblocking Usernames 1670
Creating the User List on the Local Authentication Server 1670
Saving the Configuration on the Local Authentication Server 1671
Configuring Access Points or Routers to Use the Local Authentication Server 1671
Verifying the Configuration for Local Authentication Service 1674
Monitoring and Maintaining 802.1X Local Authentication Service 1674
Configuration Examples for Remote Site IEEE 802.1X Local Authentication Service 1674
Setting Up a Local Authentication Server: Example 1674
Setting Up Two Main Servers and a Local Authentication Server: Example 1675
Displaying Local Authentication Server Configuration: Example 1676

Displaying Local Authentication Server Statistics: Example 1678
Additional References 1678
Related Documents 1678
MIBs 1679
Technical Assistance 1679
Command Reference 1679

VPN Access Control Using 802.1X Authentication 1681

Contents 1681
Prerequisites for VPN Access Control Using 802.1X Authentication 1682
Restrictions for VPN Access Control Using 802.1X Authentication 1682
Information About VPN Access Control Using 802.1X Authentication 1682
How VPN Control Using 802.1X Authentication Works 1682
802.1X Authentication Sample Topology and Configuration 1683
802.1X Supplicant Support 1684
Authentication Using Passwords and MD5 1684
How to Configure VPN Access Control Using 802.1X Authentication 1685
Configuring an AAA RADIUS Server 1685
Configuring a Router 1685
Enabling 802.1X Authentication 1685
Configuring Router and RADIUS Communication 1687
Configuring 802.1X Parameters (Retransmissions and Timeouts) 1688
Configuring the Identity Profile 1690
Configuring the Virtual Template and DHCP 1692
Configuring the Necessary Access Control Policies 1697
Configuring a Router As a Supplicant 1697
Configuring a PC 1699
Configuring a PC for VPN Access Control Using 802.1X Authentication 1699
Enabling 802.1X Authentication on a Windows 2000/XP PC 1699
Enabling 802.1X Authentication on a Windows 2000 PC 1699
Enabling 802.1X Authentication on a Windows XP PC 1700
Enabling 802.1X Authentication on Windows 2000 and Windows XP PCs 1700
Monitoring VPN Access Control Using 802.1X Authentication 1701
Verifying VPN Access Control Using 802.1X Authentication 1703
Configuration Examples for VPN Access Control Using 802.1X Authentication 1703
Typical VPN Access Control Using 802.1X Configuration: Example 1703

Access Control Policies: Example 1707
Router Acting As a Supplicant: Example 1708
Additional References 1710
Related Documents 1710
Standards 1710
MIBs 1711
RFCs 1711
Technical Assistance 1711
Command Reference 1711
Glossary 1713

WebVPN 1715
Contents 1715
Prerequisites for WebVPN 1715
Restrictions for WebVPN 1716
Information About WebVPN 1716
WebVPN 1716
Administrator Interface 1716
End User Interface 1716
How to Configure WebVPN 1724
Configuring WebVPN: Prerequisites 1724
AAA-Related Configuration 1725
DNS-Related Configuration 1726
Certificates and Trustpoints 1726
Configuring WebVPN 1728
Defining Encryption Algorithms for the SSL Protocol 1730
Displaying URL Entries on the Portal Page 1731
Maintaining and Monitoring Your WebVPN Functionality 1732
Examples 1733
Troubleshooting WebVPN 1736
Configuration Examples for WebVPN 1737
WebVPN Enabled Globally: Example 1737
WebVPN Enabled on a Specific IP Address: Example 1738
Additional References 1738
Related Documents 1738
Standards 1738
MIBs 1739
RFCs 1739
Technical Assistance 1739

Command Reference 1739

PART 7: SECURE INFRASTRUCTURE

AutoSecure 1743

Contents 1743

Information About AutoSecure 1744

Benefits of AutoSecure 1744

Secure Management Plane 1745

Secure Forwarding Plane 1748

How to Configure AutoSecure 1748

Configuring AutoSecure 1748

The auto secure Command 1748

Restrictions 1749

Configuring Additional Security 1749

Verifying AutoSecure 1750

Configuration Examples for AutoSecure 1751

AutoSecure Configuration Dialogue: Example 1751

Additional References 1754

Related Documents 1754

Standards 1754

MIBs 1755

RFCs 1755

Technical Assistance 1755

Command Reference 1755

Cisco IOS Login Enhancements 1757

Contents 1757

Information About Cisco IOS Login Enhancements 1757

Login Enhancements Functionality Overview 1758

Delays Between Successive Login Attempts 1758

Login Shutdown If DoS Attacks Are Suspected 1758

Generation of System Logging Messages for Login Detection 1758

How to Configure Cisco IOS Login Enhancements 1759

Configuring Login Parameters 1759

Login Parameter Defaults 1759

What to Do Next 1760

Verifying Login Parameters 1761

Examples 1761

Configuration Examples for Login Parameters 1762
Setting Login Parameters: Example 1762
Additional References 1763
Related Documents 1763
Standards 1763
MIBs 1763
RFCs 1763
Technical Assistance 1763
Command Reference 1764

Cisco IOS Resilient Configuration 1765

Contents 1765
Restrictions for Cisco IOS Resilient Configuration 1765
Information About Cisco IOS Resilient Configuration 1766
Feature Design of Cisco IOS Resilient Configuration 1766
How to Use Cisco IOS Resilient Configuration 1766
Archiving a Router Configuration 1767
Examples 1768
Restoring an Archived Router Configuration 1768
Additional References 1770
Related Documents 1770
Standards 1770
MIBs 1770
RFCs 1770
Technical Assistance 1770
Command Reference 1771

Image Verification 1773

Contents 1773
Restrictions for Image Verification 1773
Information About Image Verification 1774
Benefit of Image Verification 1774
How Image Verification Works 1774
How to Use Image Verification 1774
Globally Verifying the Integrity of an Image 1774
What to Do Next 1775
Verifying the Integrity of an Image That Is About to Be Copied 1775
Verifying the Integrity of an Image That Is About to Be Reloaded 1776
Configuration Examples for Image Verification 1777

Global Image Verification: Example 1777
Image Verification via the copy Command: Example 1778
Image Verification via the reload Command: Example 1778
verify Command Sample Output: Example 1778
Additional References 1779
Related Documents 1779
Standards 1779
MIBs 1779
RFCs 1779
Technical Assistance 1780
Command Reference 1780

IP Source Tracker 1781

Contents 1781
Restrictions for IP Source Tracker 1782
Information About IP Source Tracker 1782
Identifying and Tracking Denial of Service Attacks 1782
Using IP Source Tracker 1783
IP Source Tracker: Hardware Support 1783
Benefits of IP Source Tracker 1784
How to Configure IP Source Tracker 1784
Configuring IP Source Tracking 1784
What to Do Next 1785
Verifying IP Source Tracking 1785
Examples 1786
Configuration Examples for IP Source Tracker 1787
Configuring IP Source Tracking: Example 1787
Verifying Source Interface Statistics for All Tracked IP Addresses: Example 1787
Verifying a Flow Statistic Summary for All Tracked IP Addresses: Example 1788
Verifying Detailed Flow Statistics Collected by a Line Card: Example 1788
Verifying Flow Statistics Exported from Line Cards and Port Adapters: Example 1788
Additional References 1789
Related Documents 1789
Standards 1789
MIBs 1789
RFCs 1789
Technical Assistance 1790
Command Reference 1790

IP Traffic Export 1791

Contents 1791

Restrictions for IP Traffic Export 1791

Information About IP Traffic Export 1792

Benefits of IP Traffic Export 1792

How to Use IP Traffic Export 1792

Configuring IP Traffic Export 1793

IP Traffic Export Profiles Overview 1793

Troubleshooting Tips 1795

What to Do Next 1795

Displaying IP Traffic Export Configuration Data 1795

Examples 1796

Configuration Examples for IP Traffic Export 1796

Exporting IP Traffic Configuration: Example 1797

Additional References 1800

Related Documents 1800

Standards 1800

MIBs 1800

RFCs 1800

Technical Assistance 1801

Command Reference 1801

Role-Based CLI Access 1803

Contents 1803

Prerequisites for Role-Based CLI Access 1804

Restrictions for Role-Based CLI Access 1804

Information About Role-Based CLI Access 1804

Benefits of Using CLI Views 1804

Root View 1804

View Authentication via a New AAA Attribute 1805

How to Use Role-Based CLI Access 1805

Configuring a CLI View 1805

Prerequisites 1805

Troubleshooting Tips 1807

Configuring a Lawful Intercept View 1807

About Lawful Intercept Views 1807

Prerequisites 1808

Troubleshooting Tips 1809

Configuring a Superview 1809

About Superviews 1809
Monitoring Views and View Users 1811
Configuration Examples for Role-Based CLI Access 1811
Configuring a CLI View: Example 1811
Verifying a CLI View: Example 1812
Configuring a Lawful Intercept View: Example 1813
Configuring a Superview: Example 1814
Additional References 1814
Related Documents 1814
Standards 1814
MIBs 1814
RFCs 1815
Technical Assistance 1815
Command Reference 1815

Security with Passwords, Privilege Levels, and Login Usernames for CLI Sessions on Networking Devices

Configuring Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices 1819

Contents 1819
Restrictions for Configuring Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices 1820
Information About Configuring Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices 1820
Benefits of Creating a Security Scheme for Your Networking Device 1820
Cisco IOS CLI Modes 1821
User EXEC Mode 1822
Privileged EXEC Mode 1823
Global Configuration Mode 1825
Interface Configuration Mode 1826
Subinterface Configuration Mode 1827
Cisco IOS CLI Sessions 1828
Local CLI Sessions 1828
Remote CLI Sessions 1828
Terminal Lines are Used for Local and Remote CLI Sessions 1828
Protect Access to Cisco IOS EXEC Modes 1829
Protecting Access to User EXEC Mode 1829
Protecting Access to Privileged EXEC mode 1829
Cisco IOS Password Encryption Levels 1829

Cisco IOS CLI Session Usernames 1831
Cisco IOS Privilege Levels 1831
Cisco IOS Password Configuration 1832
How To Configure Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices 1832
Protecting Access to User Exec Mode 1833
Configuring and Verifying a Password for Remote CLI Sessions 1833
Configuring and Verifying a Password for Local CLI Sessions 1835
Protecting Access to Privileged Exec Mode 1837
Configuring and Verifying the Enable Password 1837
Configuring Password Encryption for Clear Text Passwords 1839
Configuring and Verifying the Enable Secret Password 1840
Configuring Security Options with Passwords, Privilege Levels and usernames to Manage Access to CLI Sessions and CLI Commands 1842
Configuring the Networking Device for the First-Line Technical Support Staff 1842
Verifying the Configuration for the First-Line Technical Support Staff 1845
Configuring and Verifying the Networking Device to Require a Username for the First-Line Technical Support Staff 1847
Recovering from a Lost or Misconfigured Password for Local CLI Sessions 1850
Networking Device Is Configured to Allow Remote CLI Sessions 1850
Networking Device Is Not Configured to Allow Remote CLI Sessions and the Local CLI Session Password Has Not Been Saved to the Startup Configuration File 1850
Networking Device is not Configured to Allow Remote CLI Sessions and the Local CLI Session Password Has Been Saved to the Startup Configuration File 1850
Recovering from a Lost or Misconfigured Password for Remote CLI Sessions 1851
Networking Device Is Configured to Allow Local CLI Sessions 1851
Networking Device Is Not Configured to Allow Local CLI Sessions and the Remote CLI Session Password Has Not Been Saved to the Startup Configuration File 1851
Networking Device Is Not Configured to Allow Local CLI Sessions and the Remote CLI Session Password Has Been Saved to the Startup Configuration File 1852
Recovering from a Lost or Misconfigured Passwords for Privileged EXEC Mode 1852
A Misconfigured Privileged EXEC Mode Password Has Not Been Saved to the Startup Configuration File 1852
A Misconfigured Privileged EXEC Mode Password Has Been Saved to the Startup Configuration File, or the Privileged EXEC Mode Password Has Been Lost 1853
Configuration Examples for Configuring Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices 1853
Configuring and Verifying a Networking Device to Allow Non Administrative Users to Clear Remote CLI Sessions: Example 1853
Configuring and Verifying a Networking Device to Allow Non Administrative Users to View the Running Configuration Automatically: Example 1855

Configuring and Verifying a Networking Device to Allow Non Administrative Users to Shutdown and Enable Interfaces: Example 1855
Where to Go Next 1856
Additional References 1857
Related Documents 1857
Standards 1857
MIBs 1857
RFCs 1857
Technical Assistance 1858
Feature Information for Configuring Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices 1858

No Service Password-Recovery 1861
Contents 1861
Prerequisites for No Service Password-Recovery 1861
Information About No Service Password-Recovery 1862
Cisco Password Recovery Procedure 1862
Configuration Registers and System Boot Configuration 1862
How to Enable No Service Password-Recovery 1862
Upgrading the ROMMON Version 1863
Verifying the Upgraded ROMMON Version 1865
Enabling No Service Password-Recovery 1865
Prerequisites 1865
Recovering a Device 1866
Examples 1867
Configuration Examples for No Service Password-Recovery 1870
Disabling Password Recovery: Example 1871
Additional References 1872
Related Documents 1872
Standards 1872
MIBs 1872
RFCs 1872
Technical Assistance 1873
Command Reference 1873

APPENDIXES

RADIUS Attributes

RADIUS Attributes Overview and RADIUS IETF Attributes 1879

In This Appendix 1879

RADIUS Attributes Overview 1879

IETF Attributes Versus VSAs 1879

RADIUS Packet Format 1880

RADIUS Packet Types 1881

RADIUS Files 1881

Dictionary File 1881

Clients File 1882

Users File 1882

Supporting Documentation 1883

RADIUS IETF Attributes 1883

Supported RADIUS IETF Attributes 1883

Comprehensive List of RADIUS Attribute Descriptions 1886

RADIUS Vendor-Proprietary Attributes 1895

Supported Vendor-Proprietary RADIUS Attributes 1895

Comprehensive List of Vendor-Proprietary RADIUS Attribute Descriptions 1900

RADIUS Vendor-Specific Attributes and RADIUS Disconnect-Cause Attribute Values 1907

RADIUS Disconnect-Cause Attribute Values 1913

Connect-Info RADIUS Attribute 77 1917

Feature Overview 1917

Benefits 1918

Related Documents 1918

Supported Platforms 1918

Supported Standards, MIBs, and RFCs 1919

Prerequisites 1919

Configuration Tasks 1919

Verifying Attribute 77 1919

Configuration Examples 1920

Configure NAS for AAA and Incoming Modem Calls Example 1920

Command Reference 1920

Encrypted Vendor-Specific Attributes 1921
Feature Overview 1921
Tagged String VSA 1922
Encrypted String VSA 1922
Tagged and Encrypted String VSA 1922
Benefits 1923
Related Documents 1923
Supported Platforms 1923
Supported Standards, MIBs, and RFCs 1924
Prerequisites 1925
Configuration Tasks 1925
Verifying Encrypted VSAs 1925
Configuration Examples 1925
NAS Configuration Example 1925
RADIUS User Profile with a Tagged and Encrypted VSA Example 1926
Command Reference 1926

Local AAA Server 1927
Contents 1927
Prerequisites for Local AAA Server 1927
Information About Local AAA Server 1928
Local Authorization Attributes: Overview 1928
Local AAA Attribute Support 1928
AAA Attribute Lists 1928
Converting from RADIUS Format to Cisco IOS AAA Format 1929
Validation of Attributes 1929
How to Configure Local AAA Server 1929
Defining a AAA Attribute List 1929
Defining a Subscriber Profile 1931
Monitoring and Troubleshooting a Local AAA Server 1932
Configuration Examples for Local AAA Server 1934
Local AAA Server: Example 1934
Mapping from the RADIUS Version of a Particular Attribute to the Cisco IOS AAA Version: Example 1935
Additional References 1935
Related Documents 1935
Standards 1935
MIBs 1936
RFCs 1936

Technical Assistance 1936
Command Reference 1936

Per-User QoS via AAA Policy Name 1937

Contents 1937
Prerequisites for Per-User QoS via AAA Policy Name 1937
Information About Per-User QoS via AAA Policy Name 1938
VSAs Added for Per-User QoS via AAA Policy Name 1938
How to Configure Per-User QoS via AAA Policy Name 1938
Monitoring and Maintaining Per-User QoS via AAA Policy Name 1938
Configuration Examples for Per-User QoS via AAA Policy Name 1939
Per-User QoS Using the AAA Policy Name 1939
Additional References 1940
Related Documents 1940
Standards 1940
MIBs 1940
RFCs 1941
Technical Assistance 1941
Command Reference 1941
Glossary 1941

RADIUS Attribute 5 (NAS-Port) Format Specified on a Per-Server Group Level 1943

Contents 1943
Prerequisites for RADIUS Attribute 5 (NAS-Port) Format Specified on a Per-Server Group Level 1944
Information About RADIUS Attribute 5 (NAS-Port) Format Specified on a Per-Server Group Level 1944
RADIUS Attribute 5 Format Customization 1944
How to Configure RADIUS Attribute 5 (NAS-Port) Format Specified on a Per-Server Group Level 1944
Configuring the RADIUS Attribute 5 Format on a Per-Server Group Level 1944
Prerequisites 1945
Monitoring and Maintaining RADIUS Attribute 5 Format on a Per-Server Group Level 1946
Configuration Examples for RADIUS Attribute 5 (NAS-Port) Format Specified on a Per-Server Group Level 1946
RADIUS Attribute 5 Format Specified on a Per-Server Level: Example 1946
Additional References 1947
Related Documents 1947
Standards 1947
MIBs 1947

RFCs 1947
Technical Assistance 1948
Command Reference 1948

RADIUS Attribute 8 (Framed-IP-Address) in Access Requests 1949
Feature Overview 1949
How It Works 1950
Benefits 1950
Related Documents 1950
Supported Platforms 1950
Supported Standards, MIBs, and RFCs 1951
Prerequisites 1951
Configuration Tasks 1951
Configuring RADIUS Attribute 8 in Access Requests 1951
Verifying RADIUS Attribute 8 in Access Requests 1952
Configuration Examples 1952
Command Reference 1952

RADIUS Attribute 82: Tunnel Assignment ID 1953
Contents 1953
Feature Overview 1953
Benefits 1954
Restrictions 1954
Related Documents 1954
Supported Platforms 1954
Supported Standards, MIBs, and RFCs 1955
Prerequisites 1955
Configuration Tasks 1955
Verifying RADIUS Attribute 82 1955
Configuration Examples 1956
LAC Configuration Example 1956
LNS Configuration Example 1956
RADIUS Configuration Example 1957
Command Reference 1957

RADIUS Attribute 104 1959
Contents 1959
Prerequisites for RADIUS Attribute 104 1959
Restrictions for RADIUS Attribute 104 1960

Information About RADIUS Attribute 104 1960
Policy-Based Routing: Background 1960
Attribute 104 and the Policy-Based Route Map 1961
RADIUS Attribute 104 Overview 1961
Permit Route Map 1961
Default Private Route 1961
Route Map Order 1961
How to Apply RADIUS Attribute 104 1961
Applying RADIUS Attribute 104 to Your User Profile 1961
Examples 1962
Verifying Route Maps 1962
Troubleshooting the RADIUS Profile 1963
Configuration Examples for RADIUS Attribute 104 1964
Route-Map Configuration in Which Attribute 104 Has Been Applied: Example 1964
Additional References 1965
Related Documents 1965
Standards 1965
MIBs 1965
RFCs 1965
Technical Assistance 1966
Command Reference 1966

RADIUS Progress Codes 1967

Feature Overview 1967
Benefits 1968
Related Documents 1968
Supported Platforms 1968
Supported Standards, MIBs, and RFCs 1969
Prerequisites 1969
Configuration Tasks 1969
Verifying Attribute 196 1969
Configuration Examples 1969
Sample Debug Output Example 1970
Command Reference 1970
Glossary 1971

RADIUS Timeout Set During Pre-Authentication 1973

Contents 1973
Prerequisites for the RADIUS Timeout Set During Pre-Authentication Feature 1974

Information About the RADIUS Timeout Set During Pre-Authentication Feature 1974
RADIUS Attribute 27 and the PPP Authentication Phase 1974
How to Configure the RADIUS Timeout Set During Pre-Authentication Feature 1974
Additional References 1975
Related Documents 1975
Standards 1975
MIBs 1975
RFCs 1976
Technical Assistance 1976
Command Reference 1976

RADIUS Tunnel Attribute Extensions 1977

Feature Overview 1977
How It Works 1977
Benefits 1978
Restrictions 1978
Related Documents 1978
Supported Platforms 1979
Supported Standards, MIBs, and RFCs 1980
Prerequisites 1980
Configuration Tasks 1980
Verifying RADIUS Attribute 90 and RADIUS Attribute 91 1981
Configuration Examples 1981
L2TP Network Server (LNS) Configuration Example 1981
RADIUS User Profile with RADIUS Tunneling Attributes 90 and 91 Example 1982
Command Reference 1982
Glossary 1983

V.92 Reporting Using RADIUS Attribute v.92-info 1985

Contents 1985
Prerequisites for V.92 Reporting Using RADIUS Attribute v.92-info 1986
Restrictions for V.92 Reporting Using RADIUS Attribute v.92-info 1986
Information About V.92 Reporting Using RADIUS Attribute v.92-info 1986
V.92 Standard Overview 1986
VSA v.92-info 1987
How to Monitor and Verify V.92 Call Information 1987
Monitoring V.92 Call Information 1987
Examples 1988

Verifying V.92 Call Information 1995
Examples 1995
Troubleshooting Tips 1998
Additional References 1999
Related Documents 1999
Standards 1999
MIBs 1999
RFCs 1999
Technical Assistance 2000
Command Reference 2000

TACACS+ Attribute-Value Pairs 2001
How to Use This Appendix 2001
TACACS+ Authentication and Authorization AV Pairs 2001
TACACS+ Accounting AV Pairs 2010

About Cisco IOS Software Documentation for Release 12.4

This chapter describes the objectives, audience, organization, and conventions of Cisco IOS software documentation. It also provides sources for obtaining documentation, technical assistance, and additional publications and information from Cisco Systems. It contains the following sections:

- Documentation Objectives, page xcvii
- Audience, page xcvii
- Documentation Organization for Cisco IOS Release 12.4, page xcviii
- Document Conventions, page civ
- Obtaining Documentation, page cv
- Documentation Feedback, page cvi
- Cisco Product Security Overview, page cvii
- Obtaining Technical Assistance, page cviii
- Obtaining Additional Publications and Information, page cix

Documentation Objectives

Cisco IOS software documentation describes the tasks and commands available to configure and maintain Cisco networking devices.

Audience

The Cisco IOS software documentation set is intended primarily for users who configure and maintain Cisco networking devices (such as routers and switches) but who may not be familiar with the configuration and maintenance tasks, the relationship among tasks, or the Cisco IOS software commands necessary to perform particular tasks. The Cisco IOS software documentation set is also intended for those users experienced with Cisco IOS software who need to know about new features, new configuration options, and new software characteristics in the current Cisco IOS software release.

Documentation Organization for Cisco IOS Release 12.4

The Cisco IOS Release 12.4 documentation set consists of the configuration guide and command reference pairs listed in Table 1 and the supporting documents listed in Table 2. The configuration guides and command references are organized by technology. For the configuration guides:

- Some technology documentation, such as that for DHCP, contains features introduced in Releases 12.2T and 12.3T and, in some cases, Release 12.2S. To assist you in finding a particular feature, a roadmap document is provided.
- Other technology documentation, such as that for OSPF, consists of a chapter and accompanying Release 12.2T and 12.3T feature documents.

Note In some cases, information contained in Release 12.2T and 12.3T feature documents augments or supersedes content in the accompanying documentation. Therefore it is important to review all feature documents for a particular technology.

Table 1 lists the Cisco IOS Release 12.4 configuration guides and command references.

Table 1 Cisco IOS Release 12.4 Configuration Guides and Command References

Configuration Guide and Command Reference Titles	Description
IP	
Cisco IOS IP Addressing Services Configuration Guide, Release 12.4 *Cisco IOS IP Addressing Services Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to configuring IP addressing and services, including Network Address Translation (NAT), Domain Name System (DNS), and Dynamic Host Configuration Protocol (DHCP). The command reference provides detailed information about the commands used in the configuration guide.
Cisco IOS IP Application Services Configuration Guide, Release 12.4 *Cisco IOS Application Services Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to configuring IP application services, including IP access lists, Web Cache Communication Protocol (WCCP), Gateway Load Balancing Protocol (GLBP), Server Load Balancing (SLB), Hot Standby Router Protocol (HSRP), and Virtual Router Redundancy Protocol (VRRP). The command reference provides detailed information about the commands used in the configuration guide.
Cisco IOS IP Mobility Configuration Guide, Release 12.4 *Cisco IOS IP Mobility Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to configuring Mobile IP and Cisco Mobile Networks. The command reference provides detailed information about the commands used in the configuration guide.
Cisco IOS IP Multicast Configuration Guide, Release 12.4 *Cisco IOS IP Multicast Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to configuring IP multicast, including Protocol Independent Multicast (PIM), Internet Group Management Protocol (IGMP), Distance Vector Multicast Routing Protocol (DVMRP), and Multicast Source Discovery Protocol (MSDP). The command reference provides detailed information about the commands used in the configuration guide.
Cisco IOS IP Routing Protocols Configuration Guide, Release 12.4 *Cisco IOS IP Routing Protocols Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to configuring IP routing protocols, including Border Gateway Protocol (BGP), Intermediate System-to-Intermediate System (IS-IS), and Open Shortest Path First (OSPF). The command reference provides detailed information about the commands used in the configuration guide.

Table 1 Cisco IOS Release 12.4 Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Description
Cisco IOS IP Switching Configuration Guide, Release 12.4 *Cisco IOS IP Switching Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to configuring IP switching features, including Cisco Express Forwarding (CEF), fast switching, and Multicast Distributed Switching (MDS). The command reference provides detailed information about the commands used in the configuration guide.
Cisco IOS IPv6 Configuration Guide, Release 12.4 *Cisco IOS IPv6 Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to configuring IP version 6 (IPv6), including IPv6 broadband access, IPv6 data-link layer, IPv6 multicast routing, IPv6 quality of service (QoS), IPv6 routing, IPv6 services and management, and IPv6 tunnel services. The command reference provides detailed information about the commands used in the configuration guide.
Cisco IOS Optimized Edge Routing Configuration Guide, Release 12.4 *Cisco IOS Optimized Edge Routing Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to configuring Optimized Edge Routing (OER) features, including OER prefix learning, OER prefix monitoring, OER operational modes, and OER policy configuration. The command reference provides detailed information about the commands used in the configuration guide.
Security and VPN	
Cisco IOS Security Configuration Guide, Release 12.4 *Cisco IOS Security Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to configuring various aspects of security, including terminal access security, network access security, accounting, traffic filters, router access, and network data encryption with router authentication. The command reference provides detailed information about the commands used in the configuration guide.
QoS	
Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.4 *Cisco IOS Quality of Service Solutions Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to configuring quality of service (QoS) features, including traffic classification and marking, traffic policing and shaping, congestion management, congestion avoidance, and signaling. The command reference provides detailed information about the commands used in the configuration guide.
LAN Switching	
Cisco IOS LAN Switching Configuration Guide, Release 12.4 *Cisco IOS LAN Switching Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to local-area network (LAN) switching features, including configuring routing between virtual LANs (VLANs) using Inter-Switch Link (ISL) encapsulation, IEEE 802.10 encapsulation, and IEEE 802.1Q encapsulation. The command reference provides detailed information about the commands used in the configuration guide.
Multiprotocol Label Switching (MPLS)	
Cisco IOS Multiprotocol Label Switching Configuration Guide, Release 12.4 *Cisco IOS Multiprotocol Label Switching Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to configuring Multiprotocol Label Switching (MPLS), including MPLS Label Distribution Protocol, MPLS traffic engineering, and MPLS Virtual Private Networks (VPNs). The command reference provides detailed information about the commands used in the configuration guide.
Network Management	
Cisco IOS IP SLAs Configuration Guide, Release 12.4 *Cisco IOS IP SLAs Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to configuring the Cisco IOS IP Service Level Assurances (IP SLAs) feature. The command reference provides detailed information about the commands used in the configuration guide.

Table 1 Cisco IOS Release 12.4 Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Description
Cisco IOS NetFlow Configuration Guide, Release 12.4 *Cisco IOS NetFlow Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to NetFlow features, including configuring NetFlow to analyze network traffic data, configuring NetFlow aggregation caches and export features, and configuring Simple Network Management Protocol (SNMP) and NetFlow MIB features. The command reference provides detailed information about the commands used in the configuration guide.
Cisco IOS Network Management Configuration Guide, Release 12.4 *Cisco IOS Network Management Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to network management features, including performing basic system management, performing troubleshooting and fault management, configuring Cisco Discovery Protocol (CDP), configuring Cisco Networking Services (CNS), configuring DistributedDirector, and configuring Simple Network Management Protocol (SNMP). The command reference provides detailed information about the commands used in the configuration guide.
Voice	
Cisco IOS Voice Configuration Library, Release 12.4 *Cisco IOS Voice Command Reference*, Release 12.4	The configuration library is a task-oriented collection of configuration guides, application guides, a troubleshooting guide, feature documents, a library preface, a voice glossary, and more. It also covers Cisco IOS support for voice call control protocols, interoperability, physical and virtual interface management, and troubleshooting. In addition, the library includes documentation for IP telephony applications. The command reference provides detailed information about the commands used in the configuration library.
Wireless / Mobility	
Cisco IOS Mobile Wireless Gateway GPRS Support Node Configuration Guide, Release 12.4 *Cisco IOS Mobile Wireless Gateway GPRS Support Node Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to understanding and configuring a Cisco IOS Gateway GPRS Support Node (GGSN) in a 2.5G General Packet Radio Service (GPRS) and 3G Universal Mobile Telecommunication System (UMTS) network. The command reference provides detailed information about the commands used in the configuration guide.
Cisco IOS Mobile Wireless Home Agent Configuration Guide, Release 12.4 *Cisco IOS Mobile Wireless Home Agent Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to understanding and configuring the Cisco Mobile Wireless Home Agent, which is an anchor point for mobile terminals for which Mobile IP or Proxy Mobile IP services are provided. The command reference provides detailed information about the commands used in the configuration guide.
Cisco IOS Mobile Wireless Packet Data Serving Node Configuration Guide, Release 12.4 *Cisco IOS Mobile Wireless Packet Data Serving Node Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to understanding and configuring the Cisco Packet Data Serving Node (PDSN), a wireless gateway between the mobile infrastructure and standard IP networks that enables packet data services in a Code Division Multiple Access (CDMA) environment. The command reference provides detailed information about the commands used in the configuration guide.

Table 1 Cisco IOS Release 12.4 Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Description
Cisco IOS Mobile Wireless Radio Access Networking Configuration Guide, Release 12.4 *Cisco IOS Mobile Wireless Radio Access Networking Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to understanding and configuring Cisco IOS Radio Access Network products. The command reference provides detailed information about the commands used in the configuration guide.
Long Reach Ethernet (LRE) and Digital Subscriber Line (xDSL)	
Cisco IOS Broadband and DSL Configuration Guide, Release 12.4 *Cisco IOS Broadband and DSL Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to configuring broadband access aggregation and digital subscriber line features. The command reference provides detailed information about the commands used in the configuration guide.
Cisco IOS Service Selection Gateway Configuration Guide, Release 12.4 *Cisco IOS Service Selection Gateway Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to configuring Service Selection Gateway (SSG) features, including subscriber authentication, service access, and accounting. The command reference provides detailed information about the commands used in the configuration guide.
Dial—Access	
Cisco IOS Dial Technologies Configuration Guide, Release 12.4 *Cisco IOS Dial Technologies Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to configuring lines, modems, and ISDN services. This guide also contains information about configuring dialup solutions, including solutions for remote sites dialing in to a central office, Internet service providers (ISPs), ISP customers at home offices, enterprise WAN system administrators implementing dial-on-demand routing, and other corporate environments. The command reference provides detailed information about the commands used in the configuration guide.
Asynchronous Transfer Mode (ATM)	
Cisco IOS Asynchronous Transfer Mode Configuration Guide, Release 12.4 *Cisco IOS Asynchronous Transfer Mode Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to configuring Asynchronous Transfer Mode (ATM), including WAN ATM, LAN ATM, and multiprotocol over ATM (MPOA). The command reference provides detailed information about the commands used in the configuration guide.
WAN	
Cisco IOS Wide-Area Networking Configuration Guide, Release 12.4 *Cisco IOS Wide-Area Networking Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to configuring wide-area network (WAN) features, including: Layer 2 Tunneling Protocol Version 3 (L2TPv3); Frame Relay; Link Access Procedure, Balanced (LAPB); and X.25. The command reference provides detailed information about the commands used in the configuration guide.

Table 1 Cisco IOS Release 12.4 Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Description
System Management	
Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.4 *Cisco IOS Configuration Fundamentals Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to using Cisco IOS software to configure and maintain Cisco routers and access servers, including information about using the Cisco IOS command-line interface (CLI), loading and maintaining system images, using the Cisco IOS file system, using the Cisco IOS Web browser user interface (UI), and configuring basic file transfer services. The command reference provides detailed information about the commands used in the configuration guide.
Cisco IOS Interface and Hardware Component Configuration Guide, Release 12.4 *Cisco IOS Interface and Hardware Component Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to configuring and managing interfaces and hardware components, including dial shelves, LAN interfaces, logical interfaces, serial interfaces, and virtual interfaces. The command reference provides detailed information about the commands used in the configuration guide.
IBM Technologies	
Cisco IOS Bridging and IBM Networking Configuration Guide, Release 12.4 *Cisco IOS Bridging Command Reference*, Release 12.4 *Cisco IOS IBM Networking Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to configuring: • Bridging features, including: transparent and source-route transparent (SRT) bridging, source-route bridging (SRB), Token Ring Inter-Switch Link (TRISL), and Token Ring Route Switch Module (TRRSM). • IBM network features, including: data-link switching plus (DLSw+), serial tunnel (STUN), and block serial tunnel (BSTUN); Logical Link Control, type 2 (LLC2), and Synchronous Data Link Control (SDLC); IBM Network Media Translation, including SDLC Logical Link Control (SDLLC) and Qualified Logical Link Control (QLLC); downstream physical unit (DSPU), Systems Network Architecture (SNA) service point, SNA Frame Relay Access, Advanced Peer-to-Peer Networking (APPN), native client interface architecture (NCIA) client/server topologies, and IBM Channel Attach. The two command references provide detailed information about the commands used in the configuration guide.
Additional and Legacy Protocols	
Cisco IOS AppleTalk Configuration Guide, Release 12.4 *Cisco IOS AppleTalk Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to configuring the AppleTalk protocol. The command reference provides detailed information about the commands used in the configuration guide.
Cisco IOS DECnet Configuration Guide, Release 12.4 *Cisco IOS DECnet Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to configuring the DECnet protocol. The command reference provides detailed information about the commands used in the configuration guide.
Cisco IOS ISO CLNS Configuration Guide, Release 12.4 *Cisco IOS ISO CLNS Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to configuring International Organization for Standardization (ISO) Connectionless Network Service (CLNS). The command reference provides detailed information about the commands used in the configuration guide.

Table 1 Cisco IOS Release 12.4 Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Description
Cisco IOS Novell IPX Configuration Guide, Release 12.4 *Cisco IOS Novell IPX Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to configuring the Novell Internetwork Packet Exchange (IPX) protocol. The command reference provides detailed information about the commands used in the configuration guide.
Cisco IOS Terminal Services Configuration Guide, Release 12.4 *Cisco IOS Terminal Services Command Reference*, Release 12.4	The configuration guide is a task-oriented guide to configuring terminal services, including DEC, local-area transport (LAT), and X.25 packet assembler/disassembler (PAD). The command reference provides detailed information about the commands used in the configuration guide.

Table 2 lists the documents and resources that support the Cisco IOS Release 12.4 software configuration guides and command references.

Table 2 Cisco IOS Release 12.4 Supporting Documents and Resources

Document Title	Description
Cisco IOS Master Commands List, Release 12.4	An alphabetical listing of all the commands documented in the Cisco IOS Release 12.4 command references.
Cisco IOS New, Modified, Replaced, and Removed Commands, Release 12.4	A listing of all the new, modified, replaced and removed commands since Cisco IOS Release 12.3, grouped by Release 12.3T maintenance release and ordered alphabetically within each group.
Cisco IOS New and Modified Commands, Release 12.3	A listing of all the new, modified, and replaced commands since Cisco IOS Release 12.2, grouped by Release 12.2T maintenance release and ordered alphabetically within each group.
Cisco IOS System Messages, Volume 1 of 2 *Cisco IOS System Messages, Volume 2 of 2*	Listings and descriptions of Cisco IOS system messages. Not all system messages indicate problems with your system. Some are purely informational, and others may help diagnose problems with communications lines, internal hardware, or the system software.
Cisco IOS Debug Command Reference, Release 12.4	An alphabetical listing of the **debug** commands and their descriptions. Documentation for each command includes a brief description of its use, command syntax, and usage guidelines.
Release Notes, Release 12.4	A description of general release information, including information about supported platforms, feature sets, platform-specific notes, and Cisco IOS software defects.
Dictionary of Internetworking Terms and Acronyms	Compilation and definitions of the terms and acronyms used in the internetworking industry.

Table 2 Cisco IOS Release 12.4 Supporting Documents and Resources (continued)

Document Title	Description
RFCs	RFCs are standards documents maintained by the Internet Engineering Task Force (IETF). Cisco IOS software documentation references supported RFCs when applicable. The full text of referenced RFCs may be obtained at the following URL: http://www.rfc-editor.org/
MIBs	MIBs are used for network monitoring. To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Document Conventions

Within Cisco IOS software documentation, the term *router* is generally used to refer to a variety of Cisco products (for example, routers, access servers, and switches). Routers, access servers, and other networking devices that support Cisco IOS software are shown interchangeably within examples. These products are used only for illustrative purposes; that is, an example that shows one product does not necessarily indicate that other products are not supported.

The Cisco IOS documentation set uses the following conventions:

Convention	Description
^ or Ctrl	The ^ and Ctrl symbols represent the Control key. For example, the key combination ^D or Ctrl-D means hold down the Control key while you press the D key. Keys are indicated in capital letters but are not case sensitive.
string	A string is a nonquoted set of characters shown in italics. For example, when setting an SNMP community string to *public*, do not use quotation marks around the string or the string will include the quotation marks.

Command syntax descriptions use the following conventions:

Convention	Description
bold	Bold text indicates commands and keywords that you enter literally as shown.
italics	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional element (keyword or argument).
|	A vertical line indicates a choice within an optional or required set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical line indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical line indicate a required choice.

Nested sets of square brackets or braces indicate optional or required choices within optional or required elements. For example:

Convention	Description
[x {y | z}]	Braces and a vertical line within square brackets indicate a required choice within an optional element.

Examples use the following conventions:

Convention	Description
`screen`	Examples of information displayed on the screen are set in Courier font.
`bold screen`	Examples of text that you must enter are set in Courier bold font.
< >	Angle brackets enclose text that is not printed to the screen, such as passwords, and are used in contexts in which the italic document convention is not available, such as ASCII text.
!	An exclamation point at the beginning of a line indicates a comment line. (Exclamation points are also displayed by the Cisco IOS software for certain processes.)
[ ]	Square brackets enclose default responses to system prompts.

The following conventions are used to attract the attention of the reader:

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Timesaver Means the *described action saves time*. You can save time by performing the action described in the paragraph.

Obtaining Documentation

Cisco documentation and additional literature are available on Cisco.com. Cisco also provides several ways to obtain technical assistance and other technical resources. These sections explain how to obtain technical information from Cisco Systems.

Cisco.com

You can access the most current Cisco documentation at this URL:

http://www.cisco.com/techsupport

You can access the Cisco website at this URL:

http://www.cisco.com

You can access international Cisco websites at this URL:

http://www.cisco.com/public/countries_languages.shtml

Product Documentation DVD

Cisco documentation and additional literature are available in the Product Documentation DVD package, which may have shipped with your product. The Product Documentation DVD is updated regularly and may be more current than printed documentation.

The Product Documentation DVD is a comprehensive library of technical product documentation on portable media. The DVD enables you to access multiple versions of hardware and software installation, configuration, and command guides for Cisco products and to view technical documentation in HTML. With the DVD, you have access to the same documentation that is found on the Cisco website without being connected to the Internet. Certain products also have .pdf versions of the documentation available.

The Product Documentation DVD is available as a single unit or as a subscription. Registered Cisco.com users (Cisco direct customers) can order a Product Documentation DVD (product number DOC-DOCDVD=) from Cisco Marketplace at this URL:

http://www.cisco.com/go/marketplace/

Ordering Documentation

Beginning June 30, 2005, registered Cisco.com users may order Cisco documentation at the Product Documentation Store in the Cisco Marketplace at this URL:

http://www.cisco.com/go/marketplace/

Nonregistered Cisco.com users can order technical documentation from 8:00 a.m. to 5:00 p.m. (0800 to 1700) PDT by calling 1 866 463-3487 in the United States and Canada, or elsewhere by calling 011 408 519-5055. You can also order documentation by e-mail at tech-doc-store-mkpl@external.cisco.com or by fax at 1 408 519-5001 in the United States and Canada, or elsewhere at 011 408 519-5001.

Documentation Feedback

You can rate and provide feedback about Cisco technical documents by completing the online feedback form that appears with the technical documents on Cisco.com.

You can send comments about Cisco documentation to bug-doc@cisco.com.

You can submit comments by using the response card (if present) behind the front cover of your document or by writing to the following address:

Cisco Systems
Attn: Customer Document Ordering
170 West Tasman Drive
San Jose, CA 95134-9883

We appreciate your comments.

Cisco Product Security Overview

Cisco provides a free online Security Vulnerability Policy portal at this URL:

http://www.cisco.com/en/US/products/products_security_vulnerability_policy.html

From this site, you can perform these tasks:

- Report security vulnerabilities in Cisco products.
- Obtain assistance with security incidents that involve Cisco products.
- Register to receive security information from Cisco.

A current list of security advisories and notices for Cisco products is available at this URL:

http://www.cisco.com/go/psirt

If you prefer to see advisories and notices as they are updated in real time, you can access a Product Security Incident Response Team Really Simple Syndication (PSIRT RSS) feed from this URL:

http://www.cisco.com/en/US/products/products_psirt_rss_feed.html

Reporting Security Problems in Cisco Products

Cisco is committed to delivering secure products. We test our products internally before we release them, and we strive to correct all vulnerabilities quickly. If you think that you might have identified a vulnerability in a Cisco product, contact PSIRT:

- Emergencies—security-alert@cisco.com

  An emergency is either a condition in which a system is under active attack or a condition for which a severe and urgent security vulnerability should be reported. All other conditions are considered nonemergencies.
- Nonemergencies—psirt@cisco.com

In an emergency, you can also reach PSIRT by telephone:

- 1 877 228-7302
- 1 408 525-6532

Tip We encourage you to use Pretty Good Privacy (PGP) or a compatible product to encrypt any sensitive information that you send to Cisco. PSIRT can work from encrypted information that is compatible with PGP versions 2.*x* through 8.*x*.

Never use a revoked or an expired encryption key. The correct public key to use in your correspondence with PSIRT is the one linked in the Contact Summary section of the Security Vulnerability Policy page at this URL:

http://www.cisco.com/en/US/products/products_security_vulnerability_policy.html

The link on this page has the current PGP key ID in use.

Obtaining Technical Assistance

Cisco Technical Support provides 24-hour-a-day award-winning technical assistance. The Cisco Technical Support & Documentation website on Cisco.com features extensive online support resources. In addition, if you have a valid Cisco service contract, Cisco Technical Assistance Center (TAC) engineers provide telephone support. If you do not have a valid Cisco service contract, contact your reseller.

Cisco Technical Support & Documentation Website

The Cisco Technical Support & Documentation website provides online documents and tools for troubleshooting and resolving technical issues with Cisco products and technologies. The website is available 24 hours a day, at this URL:

http://www.cisco.com/techsupport

Access to all tools on the Cisco Technical Support & Documentation website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register at this URL:

http://tools.cisco.com/RPF/register/register.do

Note Use the Cisco Product Identification (CPI) tool to locate your product serial number before submitting a web or phone request for service. You can access the CPI tool from the Cisco Technical Support & Documentation website by clicking the **Tools & Resources** link under Documentation & Tools. Choose **Cisco Product Identification Tool** from the Alphabetical Index drop-down list, or click the **Cisco Product Identification Tool** link under Alerts & RMAs. The CPI tool offers three search options: by product ID or model name; by tree view; or for certain products, by copying and pasting **show** command output. Search results show an illustration of your product with the serial number label location highlighted. Locate the serial number label on your product and record the information before placing a service call.

Submitting a Service Request

Using the online TAC Service Request Tool is the fastest way to open S3 and S4 service requests. (S3 and S4 service requests are those in which your network is minimally impaired or for which you require product information.) After you describe your situation, the TAC Service Request Tool provides recommended solutions. If your issue is not resolved using the recommended resources, your service request is assigned to a Cisco engineer. The TAC Service Request Tool is located at this URL:

http://www.cisco.com/techsupport/servicerequest

For S1 or S2 service requests or if you do not have Internet access, contact the Cisco TAC by telephone. (S1 or S2 service requests are those in which your production network is down or severely degraded.) Cisco engineers are assigned immediately to S1 and S2 service requests to help keep your business operations running smoothly.

To open a service request by telephone, use one of the following numbers:

Asia-Pacific: +61 2 8446 7411 (Australia: 1 800 805 227)
EMEA: +32 2 704 55 55
USA: 1 800 553-2447

For a complete list of Cisco TAC contacts, go to this URL:

http://www.cisco.com/techsupport/contacts

Definitions of Service Request Severity

To ensure that all service requests are reported in a standard format, Cisco has established severity definitions.

Severity 1 (S1)—Your network is "down," or there is a critical impact to your business operations. You and Cisco will commit all necessary resources around the clock to resolve the situation.

Severity 2 (S2)—Operation of an existing network is severely degraded, or significant aspects of your business operation are negatively affected by inadequate performance of Cisco products. You and Cisco will commit full-time resources during normal business hours to resolve the situation.

Severity 3 (S3)—Operational performance of your network is impaired, but most business operations remain functional. You and Cisco will commit resources during normal business hours to restore service to satisfactory levels.

Severity 4 (S4)—You require information or assistance with Cisco product capabilities, installation, or configuration. There is little or no effect on your business operations.

Obtaining Additional Publications and Information

Information about Cisco products, technologies, and network solutions is available from various online and printed sources.

- Cisco Marketplace provides a variety of Cisco books, reference guides, documentation, and logo merchandise. Visit Cisco Marketplace, the company store, at this URL:

  http://www.cisco.com/go/marketplace/

- *Cisco Press* publishes a wide range of general networking, training and certification titles. Both new and experienced users will benefit from these publications. For current Cisco Press titles and other information, go to Cisco Press at this URL:

  http://www.ciscopress.com

- *Packet* magazine is the Cisco Systems technical user magazine for maximizing Internet and networking investments. Each quarter, Packet delivers coverage of the latest industry trends, technology breakthroughs, and Cisco products and solutions, as well as network deployment and troubleshooting tips, configuration examples, customer case studies, certification and training information, and links to scores of in-depth online resources. You can access Packet magazine at this URL:

  http://www.cisco.com/packet

- *iQ Magazine* is the quarterly publication from Cisco Systems designed to help growing companies learn how they can use technology to increase revenue, streamline their business, and expand services. The publication identifies the challenges facing these companies and the technologies to help solve them, using real-world case studies and business strategies to help readers make sound technology investment decisions. You can access iQ Magazine at this URL:

  http://www.cisco.com/go/iqmagazine

  or view the digital edition at this URL:

  http://ciscoiq.texterity.com/ciscoiq/sample/

- *Internet Protocol Journal* is a quarterly journal published by Cisco Systems for engineering professionals involved in designing, developing, and operating public and private internets and intranets. You can access the Internet Protocol Journal at this URL:

  http://www.cisco.com/ipj

- Networking products offered by Cisco Systems, as well as customer support services, can be obtained at this URL:

  http://www.cisco.com/en/US/products/index.html

- Networking Professionals Connection is an interactive website for networking professionals to share questions, suggestions, and information about networking products and technologies with Cisco experts and other networking professionals. Join a discussion at this URL:

  http://www.cisco.com/discuss/networking

- World-class networking training is available from Cisco. You can view current offerings at this URL:

  http://www.cisco.com/en/US/learning/index.html

Using Cisco IOS Software for Release 12.4

This chapter provides helpful tips for understanding and configuring Cisco IOS software using the command-line interface (CLI). It contains the following sections:

- Understanding Command Modes, page cxi
- Getting Help, page cxii
- Using the no and default Forms of Commands, page cxv
- Saving Configuration Changes, page cxvi
- Filtering Output from the show and more Commands, page cxvi
- Finding Additional Feature Support Information, page cxvii

For an overview of Cisco IOS software configuration, see the *Cisco IOS Configuration Fundamentals Configuration Guide.*

For information on the conventions used in the Cisco IOS software documentation set, see the "About Cisco IOS Software Documentation for Release 12.4" chapter.

Understanding Command Modes

You use the CLI to access Cisco IOS software. Because the CLI is divided into many different modes, the commands available to you at any given time depend on the mode that you are currently in. Entering a question mark (**?**) at the CLI prompt allows you to obtain a list of commands available for each command mode.

When you log in to the CLI, you are in user EXEC mode. User EXEC mode contains only a limited subset of commands. To have access to all commands, you must enter privileged EXEC mode, normally by using a password. From privileged EXEC mode you can issue any EXEC command—user or privileged mode—or you can enter global configuration mode. Most EXEC commands are one-time commands. For example, **show** commands show important status information, and **clear** commands clear counters or interfaces. The EXEC commands are not saved when the software reboots.

Configuration modes allow you to make changes to the running configuration. If you later save the running configuration to the startup configuration, these changed commands are stored when the software is rebooted. To enter specific configuration modes, you must start at global configuration mode. From global configuration mode, you can enter interface configuration mode and a variety of other modes, such as protocol-specific modes.

ROM monitor mode is a separate mode used when the Cisco IOS software cannot load properly. If a valid software image is not found when the software boots or if the configuration file is corrupted at startup, the software might enter ROM monitor mode.

Table 1 describes how to access and exit various common command modes of the Cisco IOS software. It also shows examples of the prompts displayed for each mode.

Table 1 Accessing and Exiting Command Modes

Command Mode	Access Method	Prompt	Exit Method
User EXEC	Log in.	`Router>`	Use the **logout** command.
Privileged EXEC	From user EXEC mode, use the **enable** command.	`Router#`	To return to user EXEC mode, use the **disable** command.
Global configuration	From privileged EXEC mode, use the **configure terminal** command.	`Router(config)#`	To return to privileged EXEC mode from global configuration mode, use the **exit** or **end** command.
Interface configuration	From global configuration mode, specify an interface using an **interface** command.	`Router(config-if)#`	To return to global configuration mode, use the **exit** command. To return to privileged EXEC mode, use the **end** command.
ROM monitor	From privileged EXEC mode, use the **reload** command. Press the **Break** key during the first 60 seconds while the system is booting.	`>`	To exit ROM monitor mode, use the **continue** command.

For more information on command modes, see the "Using the Cisco IOS Command-Line Interface" chapter in the *Cisco IOS Configuration Fundamentals Configuration Guide*.

Getting Help

Entering a question mark (**?**) at the CLI prompt displays a list of commands available for each command mode. You can also get a list of keywords and arguments associated with any command by using the context-sensitive help feature.

To get help specific to a command mode, a command, a keyword, or an argument, use one of the following commands:

Command	Purpose
help	Provides a brief description of the help system in any command mode.
*abbreviated-command-entry***?**	Provides a list of commands that begin with a particular character string. (No space between command and question mark.)
abbreviated-command-entry<**Tab**>	Completes a partial command name.
?	Lists all commands available for a particular command mode.
command **?**	Lists the keywords or arguments that you must enter next on the command line. (Space between command and question mark.)

Example: How to Find Command Options

This section provides an example of how to display syntax for a command. The syntax can consist of optional or required keywords and arguments. To display keywords and arguments for a command, enter a question mark (**?**) at the configuration prompt or after entering part of a command followed by a space. The Cisco IOS software displays a list and brief description of available keywords and arguments. For example, if you were in global configuration mode and wanted to see all the keywords or arguments for the **arap** command, you would type **arap ?**.

The <cr> symbol in command help output stands for "carriage return." On older keyboards, the carriage return key is the Return key. On most modern keyboards, the carriage return key is the Enter key. The <cr> symbol at the end of command help output indicates that you have the option to press **Enter** to complete the command and that the arguments and keywords in the list preceding the <cr> symbol are optional. The <cr> symbol by itself indicates that no more arguments or keywords are available and that you must press **Enter** to complete the command.

Table 2 shows examples of how you can use the question mark (**?**) to assist you in entering commands. The table steps you through configuring an IP address on a serial interface on a Cisco 7206 router that is running Cisco IOS Release 12.0(3).

Table 2 How to Find Command Options

Command	Comment
`Router> enable` `Password: <password>` `Router#`	Enter the **enable** command and password to access privileged EXEC commands. You are in privileged EXEC mode when the prompt changes to `Router#`.
`Router# configure terminal` `Enter configuration commands, one per line. End with CNTL/Z.` `Router(config)#`	Enter the **configure terminal** privileged EXEC command to enter global configuration mode. You are in global configuration mode when the prompt changes to `Router(config)#`.
`Router(config)# interface serial ?` `  <0-6>  Serial interface number` `Router(config)# interface serial 4 ?` `  /` `Router(config)# interface serial 4/ ?` `  <0-3>  Serial interface number` `Router(config)# interface serial 4/0 ?` <cr> `Router(config)# interface serial 4/0` `Router(config-if)#`	Enter interface configuration mode by specifying the serial interface that you want to configure using the **interface serial** global configuration command. Enter **?** to display what you must enter next on the command line. In this example, you must enter the serial interface slot number and port number, separated by a forward slash. When the <cr> symbol is displayed, you can press **Enter** to complete the command. You are in interface configuration mode when the prompt changes to `Router(config-if)#`.

Table 2 How to Find Command Options (continued)

Command	Comment
`Router(config-if)# ?` `Interface configuration commands:` `  .` `  .` `  .` `  ip  Interface Internet Protocol config commands` `  keepalive  Enable keepalive` `  lan-name  LAN Name command` `  llc2  LLC2 Interface Subcommands` `  load-interval  Specify interval for load calculation for an interface` `  locaddr-priority  Assign a priority group` `  logging  Configure logging for interface` `  loopback  Configure internal loopback on an interface` `  mac-address  Manually set interface MAC address` `  mls  mls router sub/interface commands` `  mpoa  MPOA interface configuration commands` `  mtu  Set the interface Maximum Transmission Unit (MTU)` `  netbios  Use a defined NETBIOS access list or enable name-caching` `  no  Negate a command or set its defaults` `  nrzi-encoding  Enable use of NRZI encoding` `  ntp  Configure NTP` `  .` `  .` `  .` `Router(config-if)#`	Enter **?** to display a list of all the interface configuration commands available for the serial interface. This example shows only some of the available interface configuration commands.
`Router(config-if)# ip ?` `Interface IP configuration subcommands:` `  access-group  Specify access control for packets` `  accounting  Enable IP accounting on this interface` `  address  Set the IP address of an interface` `  authentication  authentication subcommands` `  bandwidth-percent  Set EIGRP bandwidth limit` `  broadcast-address  Set the broadcast address of an interface` `  cgmp  Enable/disable CGMP` `  directed-broadcast  Enable forwarding of directed broadcasts` `  dvmrp  DVMRP interface commands` `  hello-interval  Configures IP-EIGRP hello interval` `  helper-address  Specify a destination address for UDP broadcasts` `  hold-time  Configures IP-EIGRP hold time` `  .` `  .` `  .` `Router(config-if)# ip`	Enter the command that you want to configure for the interface. This example uses the **ip** command. Enter **?** to display what you must enter next on the command line. This example shows only some of the available interface IP configuration commands.

Table 2 How to Find Command Options (continued)

Command	Comment
`Router(config-if)# ip address ?` `  A.B.C.D  IP address` `  negotiated  IP Address negotiated over PPP` `Router(config-if)# ip address`	Enter the command that you want to configure for the interface. This example uses the **ip address** command. Enter **?** to display what you must enter next on the command line. In this example, you must enter an IP address or the **negotiated** keyword. A carriage return (<cr>) is not displayed; therefore, you must enter additional keywords or arguments to complete the command.
`Router(config-if)# ip address 172.16.0.1 ?` `  A.B.C.D  IP subnet mask` `Router(config-if)# ip address 172.16.0.1`	Enter the keyword or argument that you want to use. This example uses the 172.16.0.1 IP address. Enter **?** to display what you must enter next on the command line. In this example, you must enter an IP subnet mask. A <cr> is not displayed; therefore, you must enter additional keywords or arguments to complete the command.
`Router(config-if)# ip address 172.16.0.1 255.255.255.0 ?` `  secondary  Make this IP address a secondary address` `  <cr>` `Router(config-if)# ip address 172.16.0.1 255.255.255.0`	Enter the IP subnet mask. This example uses the 255.255.255.0 IP subnet mask. Enter **?** to display what you must enter next on the command line. In this example, you can enter the **secondary** keyword, or you can press **Enter**. A <cr> is displayed; you can press **Enter** to complete the command, or you can enter another keyword.
`Router(config-if)# ip address 172.16.0.1 255.255.255.0` `Router(config-if)#`	In this example, Enter is pressed to complete the command.

Using the no and default Forms of Commands

Almost every configuration command has a **no** form. In general, use the **no** form to disable a function. Use the command without the **no** keyword to reenable a disabled function or to enable a function that is disabled by default. For example, IP routing is enabled by default. To disable IP routing, use the **no ip routing** command; to reenable IP routing, use the **ip routing** command. The Cisco IOS software command reference publications provide the complete syntax for the configuration commands and describe what the **no** form of a command does.

Configuration commands can also have a **default** form, which returns the command settings to the default values. Most commands are disabled by default, so in such cases using the **default** form has the same result as using the **no** form of the command. However, some commands are enabled by default and

have variables set to certain default values. In these cases, the **default** form of the command enables the command and sets the variables to their default values. The Cisco IOS software command reference publications describe the effect of the **default** form of a command if the command functions differently than the **no** form.

Saving Configuration Changes

Use the **copy system:running-config nvram:startup-config** command or the **copy running-config startup-config** command to save your configuration changes to the startup configuration so that the changes will not be lost if the software reloads or a power outage occurs. For example:

```
Router# copy system:running-config nvram:startup-config
Building configuration...
```

It might take a minute or two to save the configuration. After the configuration has been saved, the following output appears:

```
[OK]
Router#
```

On most platforms, this task saves the configuration to NVRAM. On the Class A flash file system platforms, this task saves the configuration to the location specified by the CONFIG_FILE environment variable. The CONFIG_FILE variable defaults to NVRAM.

Filtering Output from the show and more Commands

You can search and filter the output of **show** and **more** commands. This functionality is useful if you need to sort through large amounts of output or if you want to exclude output that you need not see.

To use this functionality, enter a **show** or **more** command followed by the "pipe" character (|); one of the keywords **begin**, **include**, or **exclude**; and a regular expression on which you want to search or filter (the expression is case-sensitive):

command | {**begin** | **include** | **exclude**} *regular-expression*

The output matches certain lines of information in the configuration file. The following example illustrates how to use output modifiers with the **show interface** command when you want the output to include only lines in which the expression "protocol" appears:

```
Router# show interface | include protocol

FastEthernet0/0 is up, line protocol is up
Serial4/0 is up, line protocol is up
Serial4/1 is up, line protocol is up
Serial4/2 is administratively down, line protocol is down
Serial4/3 is administratively down, line protocol is down
```

For more information on the search and filter functionality, see the "Using the Cisco IOS Command-Line Interface" chapter in the *Cisco IOS Configuration Fundamentals Configuration Guide*.

Finding Additional Feature Support Information

If you want to use a specific Cisco IOS software feature, you will need to determine in which Cisco IOS software images that feature is supported. Feature support in Cisco IOS software images is dependant on three main factors: the software version (called the "Release"), the hardware model (the "Platform" or "Series"), and the "Feature Set" (collection of specific features designed for a certain network environment). Although the Cisco IOS software documentation set documents feature support information for Release 12.4 as a whole, it does not generally provide specific hardware and feature set information.

To determine the correct combination of Release (software version), Platform (hardware version), and Feature Set needed to run a particular feature (or any combination of features), use Feature Navigator.

Feature Navigator is a web-based tool available on Cisco.com at http://www.cisco.com/go/fn. Feature Navigator is available only for registered users of Cisco.com. If you do not have an account or have forgotten your username or password, click Cancel at the login dialog box and follow the instructions that appear.

Software features may also have additional limitations or restrictions. For example, a minimum amount of system memory may be required. Or there may be known issues for features on certain platforms that have not yet been resolved (called "Caveats"). For the latest information about these limitations, see the release notes for the appropriate Cisco IOS software release. Release notes provide detailed installation instructions, new feature descriptions, system requirements, limitations and restrictions, caveats, and troubleshooting information for a particular software release.

Security Overview

This chapter contains the following sections:

- About This Guide

  Preview the topics in this guide.

- Creating Effective Security Policies

  Learn tips and hints for creating a security policy for your organization. A security policy should be finalized and up to date *before* you configure any security features.

- Identifying Security Risks and Cisco IOS Solutions

  Identify common security risks that might be present in your network, and find the right Cisco IOS security feature to prevent security break-ins.

About This Guide

The *Cisco IOS Security Configuration Guide* describes how to configure Cisco IOS security features for your Cisco networking devices. These security features can protect your network against degradation or failure and also against data loss or compromise resulting from intentional attacks and from unintended but damaging mistakes by well-meaning network users.

This guide is divided into seven parts:

- Authentication, Authorization, and Accounting (AAA)
- Security Server Protocols
- Traffic Filtering, Firewalls, and Virus Detection
- IP Security (IPSec) and Internet Key Exchange (IKE)
- Public Key Infrastructure (PKI)
- Other Security Features
- Cisco IOS Secure Infrastructure

Appendixes follow the seven main divisions.

The following sections briefly describe each of these sections and the appendixes.

Authentication, Authorization, and Accounting (AAA)

This part describes how to configure Cisco's authentication, authorization, and accounting (AAA) paradigm. AAA is an architectural framework for configuring a set of three independent security functions in a consistent, modular manner.

- Authentication—Provides the method of identifying users, including login and password dialog, challenge and response, messaging support, and, depending on the security protocol you select, encryption. Authentication is the way a user is identified prior to being allowed access to the network and network services. You configure AAA authentication by defining a named list of authentication methods and then applying that list to various interfaces.
- Authorization—Provides the method for remote access control, including one-time authorization or authorization for each service, per-user account list and profile, user group support, and support of IP, IPX, ARA, and Telnet.

  Remote security servers, such as RADIUS and TACACS+, authorize users for specific rights by associating attribute-value (AV) pairs, which define those rights, with the appropriate user. AAA authorization works by assembling a set of attributes that describe what the user is authorized to perform. These attributes are compared with the information contained in a database for a given user, and the result is returned to AAA to determine the user's actual capabilities and restrictions.
- Accounting—Provides the method for collecting and sending security server information used for billing, auditing, and reporting, such as user identities, start and stop times, executed commands (such as PPP), number of packets, and number of bytes. Accounting enables you to track the services users are accessing, as well as the amount of network resources they are consuming.

Note You can configure authentication outside of AAA. However, you must configure AAA if you want to use RADIUS, TACACS+, or Kerberos or if you want to configure a backup authentication method.

Security Server Protocols

In many circumstances, AAA uses security protocols to administer its security functions. If your router or access server is acting as a network access server, AAA is the means through which you establish communication between your network access server and your RADIUS, TACACS+, or Kerberos security server.

The chapters in this part describe how to configure the following security server protocols:

- RADIUS—A distributed client/server system implemented through AAA that secures networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco routers and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.
- TACACS+—A security application implemented through AAA that provides centralized validation of users attempting to gain access to a router or network access server. TACACS+ services are maintained in a database on a TACACS+ daemon running, typically, on a UNIX or Windows NT workstation. TACACS+ provides for separate and modular authentication, authorization, and accounting facilities.
- Kerberos—A secret-key network authentication protocol implemented through AAA that uses the Data Encryption Standard (DES) cryptographic algorithm for encryption and authentication. Kerberos was designed to authenticate requests for network resources. Kerberos is based on the concept of a trusted third party that performs secure verification of users and services. The primary use of Kerberos is to verify that users and the network services they use are really who and what

they claim to be. To accomplish this, a trusted Kerberos server issues tickets to users. These tickets, which have a limited lifespan, are stored in a user's credential cache and can be used in place of the standard username-and-password authentication mechanism.

Traffic Filtering, Firewalls, and Virus Detection

This part describes how to configure your networking devices to filter traffic, function as a firewall, or detect potential viruses.

- Cisco implements traffic filters with access control lists (also called access lists). Access lists determine what traffic is blocked and what traffic is forwarded at router interfaces. Cisco provides both basic and advanced access list capabilities.
  - Basic access lists

    An overview of basic access lists is in the chapter "Access Control Lists: Overview and Guidelines." This chapter describes tips, cautions, considerations, recommendations, and general guidelines for configuring access lists for the various network protocols. You should configure basic access lists for all network protocols that will be routed through your networking device, such as IP, IPX, AppleTalk, and so forth.
  - Advanced access lists

    The advanced access list capabilities and configuration are described in the remaining chapters in the "Traffic Filtering, Firewalls, and Virus Detection" part of this document. The advanced access lists provide sophisticated and dynamic traffic filtering capabilities for stronger, more flexible network security.
- Cisco IOS Firewall provides an extensive set of security features, allowing you to configure a simple or elaborate firewall, according to your particular requirements. The following features are key components of Cisco IOS Firewall:
  - Context-based Access Control (CBAC)

    CBAC intelligently filters TCP and UDP packets based on application-layer protocol session information. You can configure CBAC to permit specified TCP and UDP traffic through a firewall only when the connection is initiated from within the network you want to protect. CBAC can inspect traffic for sessions that originate from either side of the firewall, and CBAC can be used for intranet, extranet, and Internet perimeters of your network.
  - Cisco IOS Intrusion Prevention System (IPS)

    Cisco IOS IPS acts as an in-line intrusion detection sensor, "watching" packets and sessions as they flow through the router and scanning each packet to match any of the Cisco IOS IPS signatures. When Cisco IOS IPS detects suspicious activity, it responds before network security can be compromised and logs the event through Cisco IOS syslog messages or Security Device Event Exchange (SDEE).

    Customers can download the Cisco IOS IPS (via a signature detection file [SDF]) to their router from Cisco.com via the VPN and Security Management Solution (VMS) IDS Management Console (MC) 2.3 network management device or via the Cisco Router and Security Device Manager (SDM). Thus VMS IDS MC or SDM can immediately begin scanning for new signatures.
  - Authentication Proxy

    The Cisco IOS Firewall authentication proxy feature allows network administrators to apply specific security policies on a per-user basis. Previously, user identity and related authorized access were associated with a user's IP address, or a single security policy had to be applied to

an entire user group or sub network. Now, users can be identified and authorized on the basis of their per-user policy, and access privileges tailored on an individual basis are possible, as opposed to general policy applied across multiple users.

– Port to Application Mapping (PAM)

  Port to Application Mapping (PAM) is a feature of Cisco Secure Integrated Software. PAM allows you to customize TCP or UDP port numbers for network services or applications. PAM uses this information to support network environments that run services using ports that are different from the registered or well-known ports associated with an application. For example, the information in the PAM table enables Context-based Access Control (CBAC) supported services to run on non-standard ports.

Firewalls are discussed in the chapters "Cisco IOS Firewall Overview" and "Configuring Context-Based Access Control."

- Cisco addresses the increased threat and impact of worms and viruses to networked businesses with Cisco Network Admission Control (NAC). NAC enables Cisco routers to enforce access privileges when an endpoint attempts to connect to a network. This access decision is made on the basis of information about the endpoint device, such as its current antivirus state. The antivirus state includes information such as the version of antivirus software, virus definitions, and version of the scan engine.

  NAC systems allow noncompliant devices to be denied access, placed in a quarantined area, or given restricted access to computing resources, thus keeping insecure nodes from infecting the network.

IP Security (IPSec) and Internet Key Exchange (IKE)

This section describes how to configure security for VPNs via IPSec and IKE:

- Configuring Security for VPNs with IPSec

  This module describes how to configure IPSec. IPSec provides security for transmission of sensitive information over unprotected networks such as the Internet. IPSec provides data authentication and anti-replay services in addition to data confidentiality services.

- Configuring Internet Key Exchange for IPSec VPNs

  This module describes how to configure IKE for use with IPSec VPNs. IKE is a key management protocol standard that is used in conjunction with the IPSec standard. IPSec can be configured without IKE, but IKE enhances IPSec by providing additional features, flexibility, and ease of configuration for the IPSec standard.

Public Key Infrastructure (PKI)

This section describes how to implement and manage a Cisco IOS PKI, which provides certificate management to support security protocols such as IPSec, secure shell (SSH), and secure socket layer (SSL). This section is divided into the following modules:

- Cisco IOS PKI Overview: Understanding and Planning a PKI

  This module identifies general concepts necessary to understand how a PKI functions.

- Deploying RSA Keys Within a PKI

  This module explains how to set up and deploy Rivest, Shamir, and Adelman (RSA) keys within a PKI. An RSA key pair is required before you can obtain a certificate for your router; that is, the end host must generate a pair of RSA keys and exchange the public key with the certification authority (CA) to obtain a certificate and enroll in a PKI.

- Configuring Revocation and Authorization of Certificates in a PKI

  This module describes how to configure revocation and authorization of certificates in a PKI. After a certificate is validated as a properly signed certificate, it is authorized (via methods such as, certificate maps, PKI-AAA, or a certificate-based ACL) and the revocation status is checked by the issuing CA to ensure that the certificate has not been revoked.

- Configuring Certificate Enrollment for a PKI

  Certificate enrollment, which is the process of obtaining a certificate from a CA, occurs between the end host requesting the certificate and the CA. Each peer that participates in the PKI must enroll with a CA. This module describes the different methods available for certificate enrollment and describes how to set up each method for a participating PKI peer.

- Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI

  This module describes how to use SDP in a PKI. SDP is a web-based certificate enrollment interface that can be used to easily deploy PKI between two end devices, such as a Cisco IOS client and a Cisco IOS certificate server. SDP provides a solution for users deploying a large number of peer devices, including certificates and configurations.

- Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment

  This module describes how to set up and manage a Cisco IOS Certificate Server (CS) for PKI deployment. A CS embeds a simple certificate server, with limited CA functionality, into the Cisco IOS software.

- Storing PKI Credentials External to the Router

  This module explains how to store RSA keys on device external to the router via a USB eToken. eTokens provide secure configuration distribution and allow users to store PKI credentials, such as RSA keys, for deployment.

Other Security Features

This section describes six security features in the following chapters:

- Neighbor Router Authentication: Overview and Guidelines

  This chapter briefly describes the security benefits and operation of neighbor router authentication.

  When neighbor authentication is configured on a router, the router authenticates its neighbor router before accepting any route updates from that neighbor. This ensures that a router always receives reliable routing update information from a trusted source.

- Configuring IP Security Options

  This chapter describes how to configure IP Security Options (IPSO) as described in RFC 1108. IPSO is generally used to comply with the security policy of the U.S. government's Department of Defense.

- Configuring Unicast Reverse Path Forwarding

  This chapter describes the Unicast Reverse Path Forwarding (Unicast RPF) feature, which helps mitigate problems caused by the introduction of malformed or forged (spoofed) IP source addresses into a network by discarding IP packets that lack a verifiable IP source address. For example, a number of common types of denial-of-service (DoS) attacks, including Smurf and Tribe Flood Network (TFN), can take advantage of forged or rapidly changing source IP addresses to allow attackers to thwart efforts to locate or filter the attacks. For Internet service providers (ISPs) that provide public access, Unicast RPF deflects such attacks by forwarding only packets that have source addresses that are valid and consistent with the IP routing table. This action protects the network of the ISP, its customer, and the rest of the Internet.

- Configuring Secure Shell

  This chapter describes the Secure Shell (SSH) feature. SSH is an application and a protocol that provides a secure replacement to a suite of Unix r-commands such as rsh, rlogin and rcp. (Cisco IOS supports rlogin.) The protocol secures the sessions using standard cryptographic mechanisms, and the application can be used similarly to the Berkeley rexec and rsh tools. There are currently two versions of SSH available: SSH Version 1 and SSH Version 2. Only SSH Version 1 is implemented in the Cisco IOS software.

- Configuring 802.1x Authentication Services

  This section describes how to configure local authentication and VPN access via the Institute of Electrical and Electronics Engineers (IEEE) 802.1X protocol framework.

- WebVPN

  WebVPN provides end users with unrestricted, secure remote access to enterprise sites without having VPN installed on their end devices. Users can access the enterprise sites from anywhere on the Internet and can access enterprise applications such as e-mail and web browsing.

Cisco IOS Secure Infrastructure

- This section contains features that help users secure their network infrastructure. Some of the available features are as follows: Autosecure (which simplifies the security configuration of a router and hardens the router configuration); Image Verification (which enables routers to automatically detect when the integrity of an image is accidentally corrupted as a result of transmission errors or disk corruption); Role-Based CLI Access (which allows network administrators to exercise better control over access to Cisco networking devices), and "Configuring Security with Passwords, Privilege Levels, and Login Usernames for CLI Sessions on Networking Devices" (which is a guide to implementing a baseline level of security for your networking devices).

Appendixes

The appendixes describe the supported RADIUS attributes and TACACS+ attribute-value pairs as follows:

- RADIUS Attributes

  RADIUS attributes are used to define specific AAA elements in a user profile, which is stored on the RADIUS daemon. This appendix lists the RADIUS attributes currently supported.

- TACACS+ Attribute-Value Pairs

  TACACS+ attribute-value pairs are used to define specific AAA elements in a user profile, which is stored on the TACACS+ daemon. This appendix lists the TACACS+ attribute-value pairs currently supported.

Creating Effective Security Policies

An effective security policy works to ensure that your organization's network assets are protected from sabotage and from inappropriate access—both intentional and accidental.

All network security features should be configured in compliance with your organization's security policy. If you do not have a security policy, or if your policy is out of date, you should ensure that the policy is created or updated before you decide how to configure security on your Cisco device.

The following sections provide guidelines to help you create an effective security policy:

- The Nature of Security Policies
- Two Levels of Security Policies
- Tips for Developing an Effective Security Policy

The Nature of Security Policies

You should recognize these aspects of security policies:

- Security policies represent trade-offs.

  With all security policies, there is some trade-off between user productivity and security measures that can be restrictive and time consuming. The goal of any security design is to provide maximum security with minimum impact on user access and productivity. Some security measures, such as network data encryption, do not restrict access and productivity. On the other hand, cumbersome or unnecessarily redundant verification and authorization systems can frustrate users and even prevent access to critical network resources.

- Security policies should be determined by business needs.

  Business needs should dictate the security policy; a security policy should not determine how a business operates.

- Security policies are living documents.

  Because organizations are constantly subject to change, security policies must be systematically updated to reflect new business directions, technological changes, and resource allocations.

Two Levels of Security Policies

You can think of a security policy as having two levels: a requirements level and an implementation level.

- At the requirements level, a policy defines the degree to which your network assets must be protected against intrusion or destruction and also estimates the cost (consequences) of a security breach. For example, the policy could state that only human resources personnel should be able to access personnel records, or that only IS personnel should be able to configure the backbone routers. The policy could also address the consequences of a network outage (due to sabotage), and the consequences of inadvertently making sensitive information public.
- At the implementation level, a policy defines guidelines to implement the requirements-level policy, using specific technology in a predefined way. For example, the implementation-level policy could require access lists to be configured so that only traffic from human resources host computers can access the server containing personnel records.

When creating a policy, define security requirements before defining security implementations so that you do not end up merely justifying particular technical solutions that might not actually be required.

Tips for Developing an Effective Security Policy

To develop an effective security policy, consider the recommendations in the following sections:

- Identifying Your Network Assets to Protect
- Determining Points of Risk
- Limiting the Scope of Access
- Identifying Assumptions
- Determining the Cost of Security Measures
- Considering Human Factors
- Keeping a Limited Number of Secrets
- Implementing Pervasive and Scalable Security
- Understanding Typical Network Functions
- Remembering Physical Security

Identifying Your Network Assets to Protect

The first step to developing a security policy is to understand and identify your organization's network assets. Network assets include the following:

- Networked hosts (such as PCs; includes the hosts' operating systems, applications, and data)
- Networking devices (such as routers)
- Network data (data that travels across the network)

You must both identify your network's assets and determine the degree to which each of these assets must be protected. For example, one subnetwork of hosts might contain extremely sensitive data that should be protected at all costs, while a different subnetwork of hosts might require only modest protection against security risks because there is less cost involved if the subnetwork is compromised.

Determining Points of Risk

You must understand how potential intruders can enter your organization's network or sabotage network operation. Special areas of consideration are network connections, dial-up access points, and misconfigured hosts. Misconfigured hosts, frequently overlooked as points of network entry, can be systems with unprotected login accounts (guest accounts), employ extensive trust in remote commands (such as rlogin and rsh), have illegal modems attached to them, and use easy-to-break passwords.

Limiting the Scope of Access

Organizations can create multiple barriers within networks, so that unlawful entry to one part of the system does not automatically grant entry to the entire infrastructure. Although maintaining a high level of security for the entire network can be prohibitively expensive (in terms of systems and equipment as well as productivity), you can often provide higher levels of security to the more sensitive areas of your network.

Identifying Assumptions

Every security system has underlying assumptions. For example, an organization might assume that its network is not tapped, that intruders are not very knowledgeable, that intruders are using standard software, or that a locked room is safe. It is important to identify, examine, and justify your assumptions: any hidden assumption is a potential security hole.

Determining the Cost of Security Measures

In general, providing security comes at a cost. This cost can be measured in terms of increased connection times or inconveniences to legitimate users accessing the assets, or in terms of increased network management requirements, and sometimes in terms of actual dollars spent on equipment or software upgrades.

Some security measures inevitably inconvenience some sophisticated users. Security can delay work, create expensive administrative and educational overhead, use significant computing resources, and require dedicated hardware.

When you decide which security measures to implement, you must understand their costs and weigh these against potential benefits. If the security costs are out of proportion to the actual dangers, it is a disservice to the organization to implement them.

Considering Human Factors

If security measures interfere with essential uses of the system, users resist these measures and sometimes even circumvent them. Many security procedures fail because their designers do not take this fact into account. For example, because automatically generated "nonsense" passwords can be difficult to remember, users often write them on the undersides of keyboards. A "secure" door that leads to a system's only tape drive is sometimes propped open. For convenience, unauthorized modems are often connected to a network to avoid cumbersome dial-in security procedures. To ensure compliance with your security measures, users must be able to get their work done as well as understand and accept the need for security.

Any user can compromise system security to some degree. For example, an intruder might learn passwords by simply calling legitimate users on the telephone claiming to be a system administrator and asking for them. If users understand security issues and understand the reasons for them, they are far less likely to compromise security in this way.

Defining such human factors and any corresponding policies needs to be included as a formal part of your complete security policy.

At a minimum, users must be taught never to release passwords or other secrets over unsecured telephone lines (especially through cordless or cellular telephones) or electronic mail. They should be wary of questions asked by people who call them on the telephone. Some companies have implemented formalized network security training for their employees in which employees are not allowed access to the network until they have completed a formal training program.

Keeping a Limited Number of Secrets

Most security is based on secrets; for example, passwords and encryption keys are secrets. But the more secrets there are, the harder it is to keep all of them. It is prudent, therefore, to design a security policy that relies on a limited number of secrets. Ultimately, the most important secret an organization has is the information that can help someone circumvent its security.

Implementing Pervasive and Scalable Security

Use a systematic approach to security that includes multiple, overlapping security methods.

Almost any change that is made to a system can affect security. This is especially true when new services are created. System administrators, programmers, and users need to consider the security implications of every change they make. Understanding the security implications of a change takes practice; it requires lateral thinking and a willingness to explore every way that a service could potentially be manipulated. The goal of any security policy is to create an environment that is not susceptible to every minor change.

Understanding Typical Network Functions

Understand how your network system normally functions, know what is expected and unexpected behavior, and be familiar with how devices are usually used. This kind of awareness helps the organization detect security problems. Noticing unusual events can help catch intruders before they can damage the system. Software auditing tools can help detect, log, and track unusual events. In addition, an organization should know exactly what software it relies on to provide auditing trails, and a security system should not operate on the assumption that all software is bug free.

Remembering Physical Security

The physical security of your network devices and hosts cannot be neglected. For example, many facilities implement physical security by using security guards, closed circuit television, card-key entry systems, or other means to control physical access to network devices and hosts. Physical access to a computer or router usually gives a sophisticated user complete control over that device. Physical access to a network link usually allows a person to tap into that link, jam it, or inject traffic into it. Software security measures can often be circumvented when access to the hardware is not controlled.

Identifying Security Risks and Cisco IOS Solutions

Cisco IOS software provides a comprehensive set of security features to guard against specific security risks. This section describes a few common security risks that might be present in your network, and describes how to use Cisco IOS software to protect against each of these risks:

- Preventing Unauthorized Access into Networking Devices
- Preventing Unauthorized Access into Networks
- Preventing Network Data Interception
- Preventing Fraudulent Route Updates

Preventing Unauthorized Access into Networking Devices

If someone were to gain console or terminal access into a networking device, such as a router, switch, or network access server, that person could do significant damage to your network—perhaps by reconfiguring the device, or even by simply viewing the device's configuration information.

Typically, you want administrators to have access to your networking device; you do not want other users on your local-area network or those dialing in to the network to have access to the router.

Users can access Cisco networking devices by dialing in from outside the network through an asynchronous port, connecting from outside the network through a serial port, or connecting via a terminal or workstation from within the local network.

To prevent unauthorized access into a networking device, you should configure one or more of the following security features:

- At a minimum, you should configure passwords and privileges at each networking device for all device lines and ports, as described in the module "Configuring Security with Passwords, Privilege Levels, and Login Usernames for CLI Sessions on Networking Devices." These passwords are stored on the networking device. When users attempt to access the device through a particular line or port, they must enter the password applied to the line or port before they can access the device.
- For an additional layer of security, you can also configure username/password pairs, stored in a database on the networking device, as described in the module "Configuring Security with Passwords, Privilege Levels, and Login Usernames for CLI Sessions on Networking Devices." These pairs are assigned to lines or interfaces and authenticate each user before that user can access the device. If you have defined privilege levels, you can also assign a specific privilege level (with associated rights and privileges) to each username/password pair.
- If you want to use username/password pairs, but you want to store them centrally instead of locally on each individual networking device, you can store them in a database on a security server. Multiple networking devices can then use the same database to obtain user authentication (and, if necessary, authorization) information. Cisco supports a variety of security server protocols, such as RADIUS, TACACS+, and Kerberos. If you decide to use the database on a security server to store login username/password pairs, you must configure your router or access server to support the applicable protocol; in addition, because most supported security protocols must be administered through the AAA security services, you will probably need to enable AAA. For more information about security protocols and AAA, refer to the chapters in the "Authentication, Authorization, and Accounting (AAA)" part of this document.

Note Cisco recommends that, whenever possible, AAA be used to implement authentication.

- If you want to authorize individual users for specific rights and privileges, you can implement AAA's authorization feature, using a security protocol such as TACACS+ or RADIUS. For more information about security protocol features and AAA, refer to the chapters in the "Authentication, Authorization, and Accounting (AAA)" part of this document.
- If you want to have a backup authentication method, you must configure AAA. AAA allows you to specify the primary method for authenticating users (for example, a username/password database stored on a TACACS+ server) and then specify backup methods (for example, a locally stored username/password database.) The backup method is used if the primary method's database cannot be accessed by the networking device. To configure AAA, refer to the chapters in the "Authentication, Authorization, and Accounting (AAA)" part of this document. You can configure up to four sequential backup methods.

  Note If you do not have backup methods configured, you will be denied access to the device if the username/password database cannot be accessed for any reason.

- If you want to keep an audit trail of user access, configure AAA accounting as described in the chapter "Configuring Accounting."

Preventing Unauthorized Access into Networks

If someone were to gain unauthorized access to your organization's internal network, that person could cause damage in many ways, perhaps by accessing sensitive files from a host, by planting a virus, or by hindering network performance by flooding your network with illegitimate packets.

This risk can also apply to a person within your network attempting to access another internal network such as a Research and Development subnetwork with sensitive and critical data. That person could intentionally or inadvertently cause damage; for example, that person might access confidential files or tie up a time-critical printer.

To prevent unauthorized access through a networking device into a network, you should configure one or more of these security features:

- Traffic Filtering

  Cisco uses access lists to filter traffic at networking devices. Basic access lists allow only specified traffic through the device; other traffic is simply dropped. You can specify individual hosts or subnets that should be allowed into the network, and you can specify what type of traffic should be allowed into the network. Basic access lists generally filter traffic based on source and destination addresses, and protocol type of each packet.

  Advanced traffic filtering is also available, providing additional filtering capabilities; for example, the Lock-and-Key Security feature requires each user to be authenticated via a username/password before that user's traffic is allowed onto the network.

  All the Cisco IOS traffic filtering capabilities are described in the chapters in the "Traffic Filtering, Firewalls, and Virus Detection" part of this document.

- Authentication

  You can require users to be authenticated before they gain access into a network. When users attempt to access a service or host (such as a web site or file server) within the protected network, they must first enter certain data such as a username and password, and possibly additional information such as their date of birth or mother's maiden name. After successful authentication (depending on the method of authentication), users will be assigned specific privileges, allowing them to access

specific network assets. In most cases, this type of authentication would be facilitated by using CHAP or PAP over a serial PPP connection in conjunction with a specific security protocol, such as TACACS+ or RADIUS.

Just as in preventing unauthorized access to specific network devices, you need to decide whether or not you want the authentication database to reside locally or on a separate security server. In this case, a local security database is useful if you have very few routers providing network access. A local security database does not require a separate (and costly) security server. A remote, centralized security database is convenient when you have a large number of routers providing network access because it prevents you from having to update each router with new or changed username authentication and authorization information for potentially hundreds of thousands of dial-in users. A centralized security database also helps establish consistent remote access policies throughout a corporation.

Cisco IOS software supports a variety of authentication methods. Although AAA is the primary (and recommended) method for access control, Cisco IOS software provides additional features for simple access control that are outside the scope of AAA. For more information, refer to the chapter "Configuring Authentication."

Preventing Network Data Interception

When packets travel across a network, they are susceptible to being read, altered, or "hijacked." (Hijacking occurs when a hostile party intercepts a network traffic session and poses as one of the session endpoints.)

If the data is traveling across an unsecured network such as the Internet, the data is exposed to a fairly significant risk. Sensitive or confidential data could be exposed, critical data could be modified, and communications could be interrupted if data is altered.

To protect data as it travels across a network, configure network data encryption, as described in the chapter "Configuring IPSec Network Security."

IPSec provides the following network security services. These services are optional. In general, local security policy will dictate the use of one or more of the following services:

- Data Confidentiality—The IPSec sender can encrypt packets before transmitting them across a network.
- Data Integrity—The IPSec receiver can authenticate packets sent by the IPSec sender to ensure that the data has not been altered during transmission.
- Data Origin Authentication—The IPSec receiver can authenticate the source of the IPSec packets sent. This service is dependent upon the data integrity service.
- Anti-Replay—The IPSec receiver can detect and reject replayed packets.

  Cisco IPSec prevents routed traffic from being examined or tampered with while it travels across a network. This feature causes IP packets to be encrypted at a Cisco router, routed across a network as encrypted information, and decrypted at the destination Cisco router. In between the two routers, the packets are in encrypted form and therefore the packets' contents cannot be read or altered. You define what traffic should be encrypted between the two routers, according to what data is more sensitive or critical.

  If you want to protect traffic for protocols other than IP, you can encapsulate those other protocols into IP packets using GRE encapsulation, and then encrypt the IP packets.

Typically, you do not use IPSec for traffic that is routed through networks that you consider secure. Consider using IPSec for traffic that is routed across unsecured networks, such as the Internet, if your organization could be damaged if the traffic is examined or tampered with by unauthorized individuals.

Preventing Fraudulent Route Updates

All routing devices determine where to route individual packets by using information stored in route tables. This route table information is created using route updates obtained from neighboring routers.

If a router receives a fraudulent update, the router could be tricked into forwarding traffic to the wrong destination. This could cause sensitive data to be exposed, or could cause network communications to be interrupted.

To ensure that route updates are received only from known, trusted neighbor routers, configure neighbor router authentication as described in the chapter "Neighbor Router Authentication: Overview and Guidelines."

Cisco Systems

Part 1: Authentication, Authorization, and Accounting (AAA)

AAA Overview

Access control is the way you control who is allowed access to the network server and what services they are allowed to use once they have access. Authentication, authorization, and accounting (AAA) network security services provide the primary framework through which you set up access control on your router or access server.

In This Chapter

This chapter includes the following sections:

- About AAA Security Services
- Where to Begin
- What to Do Next

About AAA Security Services

AAA is an architectural framework for configuring a set of three independent security functions in a consistent manner. AAA provides a modular way of performing the following services:

- Authentication—Provides the method of identifying users, including login and password dialog, challenge and response, messaging support, and, depending on the security protocol you select, encryption.

  Authentication is the way a user is identified prior to being allowed access to the network and network services. You configure AAA authentication by defining a named list of authentication methods, and then applying that list to various interfaces. The method list defines the types of authentication to be performed and the sequence in which they will be performed; it must be applied to a specific interface before any of the defined authentication methods will be performed. The only exception is the default method list (which is named "default"). The default method list is automatically applied to all interfaces if no other method list is defined. A defined method list overrides the default method list.

  All authentication methods, except for local, line password, and enable authentication, must be defined through AAA. For information about configuring all authentication methods, including those implemented outside of the AAA security services, refer to the chapter "Configuring Authentication."

- Authorization—Provides the method for remote access control, including one-time authorization or authorization for each service, per-user account list and profile, user group support, and support of IP, IPX, ARA, and Telnet.

  AAA authorization works by assembling a set of attributes that describe what the user is authorized to perform. These attributes are compared to the information contained in a database for a given user and the result is returned to AAA to determine the user's actual capabilities and restrictions. The database can be located locally on the access server or router or it can be hosted remotely on a RADIUS or TACACS+ security server. Remote security servers, such as RADIUS and TACACS+, authorize users for specific rights by associating attribute-value (AV) pairs, which define those rights with the appropriate user. All authorization methods must be defined through AAA.

  As with authentication, you configure AAA authorization by defining a named list of authorization methods, and then applying that list to various interfaces. For information about configuring authorization using AAA, refer to the chapter "Configuring Authorization."

- Accounting—Provides the method for collecting and sending security server information used for billing, auditing, and reporting, such as user identities, start and stop times, executed commands (such as PPP), number of packets, and number of bytes.

  Accounting enables you to track the services users are accessing as well as the amount of network resources they are consuming. When AAA accounting is activated, the network access server reports user activity to the RADIUS or TACACS+ security server (depending on which security method you have implemented) in the form of accounting records. Each accounting record is comprised of accounting AV pairs and is stored on the access control server. This data can then be analyzed for network management, client billing, and/or auditing. All accounting methods must be defined through AAA. As with authentication and authorization, you configure AAA accounting by defining a named list of accounting methods, and then applying that list to various interfaces. For information about configuring accounting using AAA, refer to the chapter "Configuring Accounting."

In many circumstances, AAA uses protocols such as RADIUS, TACACS+, or Kerberos to administer its security functions. If your router or access server is acting as a network access server, AAA is the means through which you establish communication between your network access server and your RADIUS, TACACS+, or Kerberos security server.

Although AAA is the primary (and recommended) method for access control, Cisco IOS software provides additional features for simple access control that are outside the scope of AAA, such as local username authentication, line password authentication, and enable password authentication. However, these features do not provide the same degree of access control that is possible by using AAA.

This section includes the following sections:

- Benefits of Using AAA
- AAA Philosophy
- Method Lists

Benefits of Using AAA

AAA provides the following benefits:

- Increased flexibility and control of access configuration
- Scalability
- Standardized authentication methods, such as RADIUS, TACACS+, and Kerberos
- Multiple backup systems

Note The deprecated protocols, TACACS and extended TACACS, are not compatible with AAA; if you select these security protocols, you will not be able to take advantage of the AAA security services.

AAA Philosophy

AAA is designed to enable you to dynamically configure the type of authentication and authorization you want on a per-line (per-user) or per-service (for example, IP, IPX, or VPDN) basis. You define the type of authentication and authorization you want by creating method lists, then applying those method lists to specific services or interfaces.

For information about applications that use AAA, such as per-user configuration and virtual profiles, refer to the chapters "Configuring Per-User Configuration" and "Configuring Virtual Profiles" in the *Cisco IOS Dial Technologies Configuration Guide,* Release 12.2.

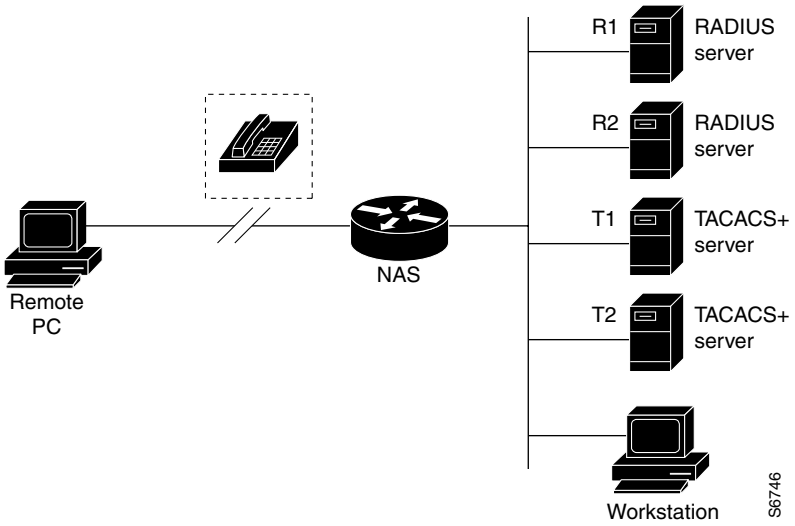
Method Lists

A method list is a sequential list that defines the authentication methods used to authenticate a user. Method lists enable you to designate one or more security protocols to be used for authentication, thus ensuring a backup system for authentication in case the initial method fails. Cisco IOS software uses the first method listed to authenticate users; if that method does not respond, Cisco IOS software selects the next authentication method in the method list. This process continues until there is successful communication with a listed authentication method or the authentication method list is exhausted, in which case authentication fails.

Note Cisco IOS software attempts authentication with the next listed authentication method only when there is no response from the previous method. If authentication fails at any point in this cycle—meaning that the security server or local username database responds by denying the user access—the authentication process stops and no other authentication methods are attempted.

Figure 1 shows a typical AAA network configuration that includes four security servers: R1 and R2 are RADIUS servers, and T1 and T2 are TACACS+ servers.

Figure 1 Typical AAA Network Configuration



Suppose the system administrator has defined a method list where R1 will be contacted first for authentication information, then R2, T1, T2, and finally the local username database on the access server itself. When a remote user attempts to dial in to the network, the network access server first queries R1 for authentication information. If R1 authenticates the user, it issues a PASS response to the network access server and the user is allowed to access the network. If R1 returns a FAIL response, the user is denied access and the session is terminated. If R1 does not respond, then the network access server processes that as an ERROR and queries R2 for authentication information. This pattern continues through the remaining designated methods until the user is either authenticated or rejected, or until the session is terminated. If all of the authentication methods return errors, the network access server will process the session as a failure, and the session will be terminated.

Note A FAIL response is significantly different from an ERROR. A FAIL means that the user has not met the criteria contained in the applicable authentication database to be successfully authenticated. Authentication ends with a FAIL response. An ERROR means that the security server has not responded to an authentication query. Because of this, no authentication has been attempted. Only when an ERROR is detected will AAA select the next authentication method defined in the authentication method list.

Where to Begin

You must first decide what kind of security solution you want to implement. You need to assess the security risks in your particular network and decide on the appropriate means to prevent unauthorized entry and attack. For more information about assessing your security risks and possible security solutions, refer to the chapter “Security Overview.” Cisco recommends that you use AAA, no matter how minor your security needs might be.

This section includes the following subsections:

- Overview of the AAA Configuration Process
- Enabling AAA
- Disabling AAA

Overview of the AAA Configuration Process

Configuring AAA is relatively simple after you understand the basic process involved. To configure security on a Cisco router or access server using AAA, follow this process:

1. Enable AAA by using the **aaa new-model** global configuration command.
2. If you decide to use a separate security server, configure security protocol parameters, such as RADIUS, TACACS+, or Kerberos.
3. Define the method lists for authentication by using an AAA authentication command.
4. Apply the method lists to a particular interface or line, if required.
5. (Optional) Configure authorization using the **aaa authorization** command.
6. (Optional) Configure accounting using the **aaa accounting** command.

For a complete description of the commands used in this chapter, refer to the chapter "Authentication Commands" of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Enabling AAA

Before you can use any of the services AAA network security services provide, you must enable AAA.

Note When you enable AAA, you can no longer access the commands to configure the older protocols, TACACS or extended TACACS. If you decided to use TACACS or extended TACACS in your security solution, do not enable AAA.

To enable AAA, use the following command in global configuration mode:

Command	Purpose
`Router (config)# aaa new-model`	Enables AAA.

Disabling AAA

You can disable AAA functionality with a single command if you decide that your security needs cannot be met by AAA but can be met by using TACACS, extended TACACS, or a line security method that can be implemented without AAA. To disable AAA, use the following command in global configuration mode:

Command	Purpose
`Router(config)# no aaa new-model`	Disables AAA.

What to Do Next

Once you have enabled AAA, you are ready to configure the other elements relating to your selected security solution. Table 3 describes AAA configuration tasks and where to find more information.

Table 3 AAA Access Control Security Solutions Methods

Task	Chapter in the *Cisco IOS Security Configuration Guide*
Configuring local login authentication	"Configuring Authentication"
Controlling login using security server authentication	"Configuring Authentication"
Defining method lists for authentication	"Configuring Authentication"
Applying method lists to a particular interface or line	"Configuring Authentication"
Configuring RADIUS security protocol parameters	"Configuring RADIUS"
Configuring TACACS+ security protocol parameters	"Configuring TACACS+"
Configuring Kerberos security protocol parameters	"Configuring Kerberos"
Enabling TACACS+ authorization	"Configuring Authorization"
Enabling RADIUS authorization	"Configuring Authorization"
Viewing supported IETF RADIUS attributes	"RADIUS Attributes" (Appendix)
Viewing supported vendor-specific RADIUS attributes	"RADIUS Attributes" (Appendix)
Viewing supported TACACS+ AV pairs	"TACACS+ AV Pairs" (Appendix)
Enabling accounting	"Configuring Accounting"

If you have elected not to use the AAA security services, see the "Configuring Authentication" chapter for the non-AAA configuration task "Configuring Login Authentication."

Authentication

This part consists of the following:

- Configuring Authentication
- AAA Double Authentication Secured by Absolute Timeout
- Login Password Retry Lockout
- MSCHAP Version 2
- RADIUS EAP Support
- RADIUS Packet of Disconnect

Configuring Authentication

Authentication verifies users before they are allowed access to the network and network services. The Cisco IOS software implementation of authentication is divided into two main categories:

- AAA Authentication Methods Configuration Task List
- Non-AAA Authentication Methods

Authentication, for the most part, is implemented through the AAA security services. Cisco recommends that, whenever possible, AAA be used to implement authentication.

This chapter describes both AAA and non-AAA authentication methods. For authentication configuration examples, refer to the "Authentication Examples" section at the end of this chapter. For a complete description of the AAA commands used in this chapter, refer to the "Authentication, Authorization, and Accounting (AAA)" part of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature, or refer to the software release notes for a specific release. For more information, see the "Finding Additional Feature Support Information" section on page cxvii in the chapter Using Cisco IOS Software for Release 12.4.

In This Chapter

This chapter contains the following sections:

- Named Method Lists for Authentication
- AAA Authentication Methods Configuration Task List
- Non-AAA Authentication Methods
- Authentication Examples

Named Method Lists for Authentication

To configure AAA authentication, you must first define a named list of authentication methods, and then apply that list to various interfaces. The method list defines the types of authentication to be performed and the sequence in which they will be performed; it must be applied to a specific interface before any

of the defined authentication methods will be performed. The only exception is the default method list (which is named "default"). The default method list is automatically applied to all interfaces except those that have a named method list explicitly defined. A defined method list overrides the default method list.

A method list is a sequential list describing the authentication methods to be queried in order to authenticate a user. Method lists enable you to designate one or more security protocols to be used for authentication, thus ensuring a backup system for authentication in case the initial method fails. Cisco IOS software uses the first listed method to authenticate users. If that method fails to respond, the Cisco IOS software selects the next authentication method listed in the method list. This process continues until there is successful communication with a listed authentication method, or all methods defined in the method list are exhausted.

It is important to note that the Cisco IOS software attempts authentication with the next listed authentication method only when there is no response from the previous method. If authentication fails at any point in this cycle—meaning that the security server or local username database responds by denying the user access—the authentication process stops and no other authentication methods are attempted.

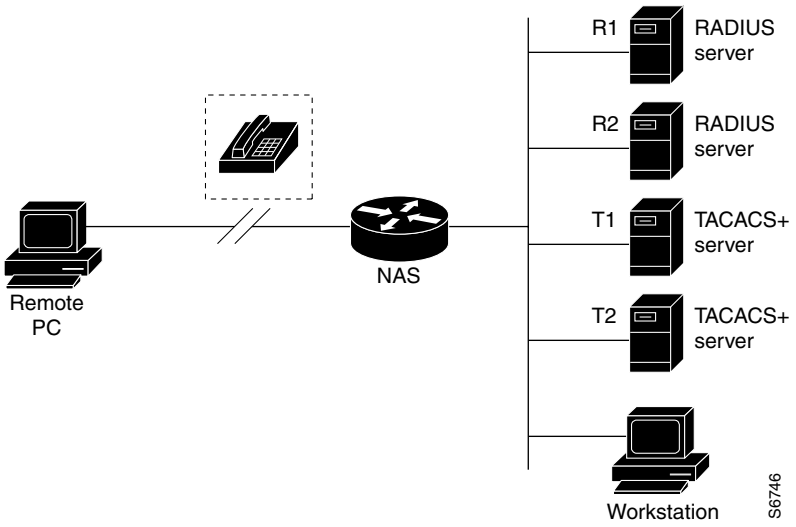
This section contains the following subsections:

- Method Lists and Server Groups
- Method List Examples
- AAA Authentication General Configuration Procedure

Method Lists and Server Groups

A server group is a way to group existing RADIUS or TACACS+ server hosts for use in method lists. Figure 2 shows a typical AAA network configuration that includes four security servers: R1 and R2 are RADIUS servers and T1 and T2 are TACACS+ servers. R1 and R2 make up the group of RADIUS server. T1 and T2 make up the group of TACACS+ servers.

Figure 2 Typical AAA Network Configuration



Using server groups, you can specify a subset of the configured server hosts and use them for a particular service. For example, server groups allow you to define R1 and R2 as a server group, and define T1 and T2 as a separate server group. For example, you can specify R1 and T1 in the method list for authentication login, while specifying R2 and T2 in the method list for PPP authentication.

Server groups also can include multiple host entries for the same server, as long as each entry has a unique identifier. The combination of an IP address and a UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. In other words, this unique identifier enables RADIUS requests to be sent to different UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service—for example, authentication—the second host entry configured acts as failover backup to the first one. Using this example, if the first host entry fails to provide accounting services, the network access server will try the second host entry configured on the same device for accounting services. (The RADIUS host entries will be tried in the order in which they are configured.)

For more information about configuring server groups and about configuring server groups based on Dialed Number Identification Service (DNIS) numbers, refer to the "Configuring RADIUS" or "Configuring TACACS+" chapter.

Method List Examples

Suppose the system administrator has decided on a security solution where all interfaces will use the same authentication methods to authenticate PPP connections. In the RADIUS group, R1 is contacted first for authentication information, then if there is no response, R2 is contacted. If R2 does not respond, T1 in the TACACS+ group is contacted; if T1 does not respond, T2 is contacted. If all designated servers fail to respond, authentication falls to the local username database on the access server itself. To implement this solution, the system administrator would create a default method list by entering the following command:

```
aaa authentication ppp default group radius group tacacs+ local
```

In this example, "default" is the name of the method list. The protocols included in this method list are listed after the name, in the order they are to be queried. The default list is automatically applied to all interfaces.

When a remote user attempts to dial in to the network, the network access server first queries R1 for authentication information. If R1 authenticates the user, it issues a PASS response to the network access server and the user is allowed to access the network. If R1 returns a FAIL response, the user is denied access and the session is terminated. If R1 does not respond, then the network access server processes that as an ERROR and queries R2 for authentication information. This pattern would continue through the remaining designated methods until the user is either authenticated or rejected, or until the session is terminated.

It is important to remember that a FAIL response is significantly different from an ERROR. A FAIL means that the user has not met the criteria contained in the applicable authentication database to be successfully authenticated. Authentication ends with a FAIL response. An ERROR means that the security server has not responded to an authentication query. Because of this, no authentication has been attempted. Only when an ERROR is detected will AAA select the next authentication method defined in the authentication method list.

Suppose the system administrator wants to apply a method list only to a particular interface or set of interfaces. In this case, the system administrator creates a named method list and then applies this named list to the applicable interfaces. The following example shows how the system administrator can implement an authentication method that will be applied only to interface 3:

```
aaa authentication ppp default group radius group tacacs+ local
aaa authentication ppp apple group radius group tacacs+ local none
 interface async 3
 ppp authentication chap apple
```

In this example, “apple” is the name of the method list, and the protocols included in this method list are listed after the name in the order in which they are to be performed. After the method list has been created, it is applied to the appropriate interface. Note that the method list name (apple) in both the AAA and PPP authentication commands must match.

In the following example, the system administrator uses server groups to specify that only R2 and T2 are valid servers for PPP authentication. To do this, the administrator must define specific server groups whose members are R2 (172.16.2.7) and T2 (172.16.2.77), respectively. In this example, the RADIUS server group “rad2only” is defined as follows using the **aaa group server** command:

```
aaa group server radius rad2only
 server 172.16.2.7
```

The TACACS+ server group “tac2only” is defined as follows using the **aaa group server** command:

```
aaa group server tacacs+ tac2only
 server 172.16.2.77
```

The administrator then applies PPP authentication using the server groups. In this example, the default methods list for PPP authentication follows this order: **group rad2only**, **group tac2only**, and **local**:

```
aaa authentication ppp default group rad2only group tac2only local
```

AAA Authentication General Configuration Procedure

To configure AAA authentication, perform the following tasks:

1. Enable AAA by using the **aaa new-model** global configuration command. For more information about configuring AAA, refer to the chapter “AAA Overview”.
2. Configure security protocol parameters, such as RADIUS, TACACS+, or Kerberos if you are using a security server. For more information about RADIUS, refer to the chapter “Configuring RADIUS”. For more information about TACACS+, refer to the chapter “Configuring TACACS+”. For more information about Kerberos, refer to the chapter “Configuring Kerberos”.
3. Define the method lists for authentication by using an AAA authentication command.
4. Apply the method lists to a particular interface or line, if required.

AAA Authentication Methods Configuration Task List

This section discusses the following AAA authentication methods:

- Configuring Login Authentication Using AAA
- Configuring PPP Authentication Using AAA
- Configuring AAA Scalability for PPP Requests
- Configuring ARAP Authentication Using AAA

- Configuring NASI Authentication Using AAA
- Specifying the Amount of Time for Login Input
- Enabling Password Protection at the Privileged Level
- Changing the Text Displayed at the Password Prompt
- Configuring Message Banners for AAA Authentication
- Configuring AAA Packet of Disconnect
- Enabling Double Authentication
- Enabling Automated Double Authentication

Note AAA features are not available for use until you enable AAA globally by issuing the **aaa new-model** command. For more information about enabling AAA, refer to the "AAA Overview" chapter.

For authentication configuration examples using the commands in this chapter, refer to the section "Authentication Examples" at the end of the this chapter.

Configuring Login Authentication Using AAA

The AAA security services facilitate a variety of login authentication methods. Use the **aaa authentication login** command to enable AAA authentication no matter which of the supported login authentication methods you decide to use. With the **aaa authentication login** command, you create one or more lists of authentication methods that are tried at login. These lists are applied using the **login authentication** line configuration command.

To configure login authentication by using AAA, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables AAA globally.
Step 2	`Router(config)# aaa authentication login {default | list-name} method1 [method2...]`	Creates a local authentication list.
Step 3	`Router(config)# line [aux | console | tty | vty] line-number [ending-line-number]`	Enters line configuration mode for the lines to which you want to apply the authentication list.
Step 4	`Router(config-line)# login authentication {default | list-name}`	Applies the authentication list to a line or set of lines.

The *list-name* is a character string used to name the list you are creating. The method argument refers to the actual method the authentication algorithm tries. The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

For example, to specify that authentication should succeed even if (in this example) the TACACS+ server returns an error, enter the following command:

```
aaa authentication login default group tacacs+ none
```

Note Because the **none** keyword enables *any* user logging in to successfully authenticate, it should be used only as a backup method of authentication.

To create a default list that is used when a named list is *not* specified in the **login authentication** command, use the **default** keyword followed by the methods that are to be used in default situations. The default method list is automatically applied to all interfaces.

For example, to specify RADIUS as the default method for user authentication during login, enter the following command:

```
aaa authentication login default group radius
```

Table 4 lists the supported login authentication methods.

Table 4 AAA Authentication Login Methods

Keyword	Description
enable	Uses the enable password for authentication.
krb5	Uses Kerberos 5 for authentication.
krb5-telnet	Uses Kerberos 5 Telnet authentication protocol when using Telnet to connect to the router. If selected, this keyword must be listed as the first method in the method list.
line	Uses the line password for authentication.
local	Uses the local username database for authentication.
local-case	Uses case-sensitive local username authentication.
none	Uses no authentication.
group radius	Uses the list of all RADIUS servers for authentication.
group tacacs+	Uses the list of all TACACS+ servers for authentication.
group *group-name*	Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the **aaa group server radius** or **aaa group server tacacs+** command.

Note The **login** command only changes username and privilege level but does not execute a shell; therefore autocommands will not be executed. To execute autocommands under this circumstance, you need to establish a Telnet session back into the router (loop-back). Make sure that the router has been configured for secure Telnet sessions if you choose to implement autocommands this way.

This section includes the following sections:

- Login Authentication Using Enable Password
- Login Authentication Using Kerberos
- Login Authentication Using Line Password
- Login Authentication Using Local Password
- Login Authentication Using Group RADIUS
- Login Authentication Using Group TACACS+
- Login Authentication Using group group-name

Login Authentication Using Enable Password

Use the **aaa authentication login** command with the **enable** *method* keyword to specify the enable password as the login authentication method. For example, to specify the enable password as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default enable
```

Before you can use the enable password as the login authentication method, you need to define the enable password. For more information about defining enable passwords, refer to the chapter "Configuring Passwords and Privileges."

Login Authentication Using Kerberos

Authentication via Kerberos is different from most other authentication methods: the user's password is never sent to the remote access server. Remote users logging in to the network are prompted for a username. If the key distribution center (KDC) has an entry for that user, it creates an encrypted ticket granting ticket (TGT) with the password for that user and sends it back to the router. The user is then prompted for a password, and the router attempts to decrypt the TGT with that password. If it succeeds, the user is authenticated and the TGT is stored in the user's credential cache on the router.

While **krb5** does use the KINIT program, a user does not need to run the KINIT program to get a TGT to authenticate to the router. This is because KINIT has been integrated into the login procedure in the Cisco IOS implementation of Kerberos.

Use the **aaa authentication login** command with the **krb5** *method* keyword to specify Kerberos as the login authentication method. For example, to specify Kerberos as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default krb5
```

Before you can use Kerberos as the login authentication method, you need to enable communication with the Kerberos security server. For more information about establishing communication with a Kerberos server, refer to the chapter "Configuring Kerberos."

Login Authentication Using Line Password

Use the **aaa authentication login** command with the **line** *method* keyword to specify the line password as the login authentication method. For example, to specify the line password as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default line
```

Before you can use a line password as the login authentication method, you need to define a line password. For more information about defining line passwords, refer to the section "Configuring Line Password Protection" in this chapter.

Login Authentication Using Local Password

Use the **aaa authentication login** command with the **local** *method* keyword to specify that the Cisco router or access server will use the local username database for authentication. For example, to specify the local username database as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default local
```

For information about adding users into the local username database, refer to the section "Establishing Username Authentication" in this chapter.

Login Authentication Using Group RADIUS

Use the **aaa authentication login** command with the **group radius** *method* to specify RADIUS as the login authentication method. For example, to specify RADIUS as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default group radius
```

Before you can use RADIUS as the login authentication method, you need to enable communication with the RADIUS security server. For more information about establishing communication with a RADIUS server, refer to the chapter "Configuring RADIUS."

Configuring RADIUS Attribute 8 in Access Requests

Once you have used the **aaa authentication login** command to specify RADIUS and your login host has been configured to request its IP address from the NAS, you can send attribute 8 (Framed-IP-Address) in access-request packets by using the **radius-server attribute 8 include-in-access-req** command in global configuration mode. This command makes it possible for a NAS to provide the RADIUS server with a hint of the user IP address in advance of user authentication. For more information about attribute 8, refer to the appendix "RADIUS Attributes" at the end of the book.

Login Authentication Using Group TACACS+

Use the **aaa authentication login** command with the **group tacacs+** *method* to specify TACACS+ as the login authentication method. For example, to specify TACACS+ as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default group tacacs+
```

Before you can use TACACS+ as the login authentication method, you need to enable communication with the TACACS+ security server. For more information about establishing communication with a TACACS+ server, refer to the chapter "Configuring TACACS+."

Login Authentication Using group group-name

Use the **aaa authentication login** command with the **group** *group-name* method to specify a subset of RADIUS or TACACS+ servers to use as the login authentication method. To specify and define the group name and the members of the group, use the **aaa group server** command. For example, use the **aaa group server** command to first define the members of **group loginrad**:

```
aaa group server radius loginrad
 server 172.16.2.3
 server 172.16.2 17
 server 172.16.2.32
```

This command specifies RADIUS servers 172.16.2.3, 172.16.2.17, and 172.16.2.32 as members of the group *loginrad*.

To specify **group loginrad** as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication login default group loginrad
```

Before you can use a group name as the login authentication method, you need to enable communication with the RADIUS or TACACS+ security server. For more information about establishing communication with a RADIUS server, refer to the chapter "Configuring RADIUS." For more information about establishing communication with a TACACS+ server, refer to the chapter "Configuring TACACS+."

Configuring PPP Authentication Using AAA

Many users access network access servers through dialup via async or ISDN. Dialup via async or ISDN bypasses the CLI completely; instead, a network protocol (such as PPP or ARA) starts as soon as the connection is established.

The AAA security services facilitate a variety of authentication methods for use on serial interfaces running PPP. Use the **aaa authentication ppp** command to enable AAA authentication no matter which of the supported PPP authentication methods you decide to use.

To configure AAA authentication methods for serial lines using PPP, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables AAA globally.
Step 2	`Router(config)# aaa authentication ppp {default | list-name} method1 [method2...]`	Creates a local authentication list.
Step 3	`Router(config)# interface interface-type interface-number`	Enters interface configuration mode for the interface to which you want to apply the authentication list.
Step 4	`Router(config-if)# ppp authentication {protocol1 [protocol2...]} [if-needed] {default | list-name} [callin] [one-time][optional]`	Applies the authentication list to a line or set of lines. In this command, *protocol1* and *protocol2* represent the following protocols: CHAP, MS-CHAP, and PAP. PPP authentication is attempted first using the first authentication method, specified by *protocol1*. If *protocol1* is unable to establish authentication, the next configured protocol is used to negotiate authentication.

With the **aaa authentication ppp** command, you create one or more lists of authentication methods that are tried when a user tries to authenticate via PPP. These lists are applied using the **ppp authentication** line configuration command.

To create a default list that is used when a named list is *not* specified in the **ppp authentication** command, use the **default** keyword followed by the methods you want used in default situations.

For example, to specify the local username database as the default method for user authentication, enter the following command:

```
aaa authentication ppp default local
```

The *list-name* is any character string used to name the list you are creating. The method argument refers to the actual method the authentication algorithm tries. The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

For example, to specify that authentication should succeed even if (in this example) the TACACS+ server returns an error, enter the following command:

```
aaa authentication ppp default group tacacs+ none
```

Note Because **none** allows all users logging in to authenticate successfully, it should be used as a backup method of authentication.

Table 5 lists the supported login authentication methods.

Table 5 AAA Authentication PPP Methods

Keyword	Description
if-needed	Does not authenticate if user has already been authenticated on a TTY line.
krb5	Uses Kerberos 5 for authentication (can only be used for PAP authentication).
local	Uses the local username database for authentication.
local-case	Uses case-sensitive local username authentication.
none	Uses no authentication.
group radius	Uses the list of all RADIUS servers for authentication.
group tacacs+	Uses the list of all TACACS+ servers for authentication.
group *group-name*	Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the **aaa group server radius** or **aaa group server tacacs+** command.

This section includes the following sections:

- PPP Authentication Using Kerberos
- PPP Authentication Using Local Password
- PPP Authentication Using Group RADIUS
- PPP Authentication Using Group TACACS+
- PPP Authentication Using group group-name

PPP Authentication Using Kerberos

Use the **aaa authentication ppp** command with the **krb5** *method* keyword to specify Kerberos as the authentication method for use on interfaces running PPP. For example, to specify Kerberos as the method of user authentication when no other method list has been defined, enter the following command:

```
aaa authentication ppp default krb5
```

Before you can use Kerberos as the PPP authentication method, you need to enable communication with the Kerberos security server. For more information about establishing communication with a Kerberos server, refer to the chapter “Configuring Kerberos”.

Note Kerberos login authentication works only with PPP PAP authentication.

PPP Authentication Using Local Password

Use the **aaa authentication ppp** command with the *method* keyword **local** to specify that the Cisco router or access server will use the local username database for authentication. For example, to specify the local username database as the method of authentication for use on lines running PPP when no other method list has been defined, enter the following command:

```
aaa authentication ppp default local
```

For information about adding users into the local username database, refer to the section “Establishing Username Authentication” in this chapter.

PPP Authentication Using Group RADIUS

Use the **aaa authentication ppp** command with the **group radius** *method* to specify RADIUS as the login authentication method. For example, to specify RADIUS as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication ppp default group radius
```

Before you can use RADIUS as the PPP authentication method, you need to enable communication with the RADIUS security server. For more information about establishing communication with a RADIUS server, refer to the chapter "Configuring RADIUS."

Configuring RADIUS Attribute 44 in Access Requests

Once you have used the **aaa authentication ppp** command with the **group radius** *method* to specify RADIUS as the login authentication method, you can configure your router to send attribute 44 (Acct-Seccion-ID) in access-request packets by using the **radius-server attribute 44 include-in-access-req** command in global configuration mode. This command allows the RADIUS daemon to track a call from the beginning of the call to the end of the call. For more information on attribute 44, refer to the appendix "RADIUS Attributes" at the end of the book.

PPP Authentication Using Group TACACS+

Use the **aaa authentication ppp** command with the **group tacacs+** *method* to specify TACACS+ as the login authentication method. For example, to specify TACACS+ as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication ppp default group tacacs+
```

Before you can use TACACS+ as the PPP authentication method, you need to enable communication with the TACACS+ security server. For more information about establishing communication with a TACACS+ server, refer to the chapter "Configuring TACACS+."

PPP Authentication Using group group-name

Use the **aaa authentication ppp** command with the **group** *group-name* method to specify a subset of RADIUS or TACACS+ servers to use as the login authentication method. To specify and define the group name and the members of the group, use the **aaa group server** command. For example, use the **aaa group server** command to first define the members of **group pppprad**:

```
aaa group server radius pppprad
 server 172.16.2.3
 server 172.16.2 17
 server 172.16.2.32
```

This command specifies RADIUS servers 172.16.2.3, 172.16.2.17, and 172.16.2.32 as members of the group *pppprad*.

To specify **group pppprad** as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication ppp default group pppprad
```

Before you can use a group name as the PPP authentication method, you need to enable communication with the RADIUS or TACACS+ security server. For more information about establishing communication with a RADIUS server, refer to the chapter "Configuring RADIUS". For more information about establishing communication with a TACACS+ server, refer to the chapter "Configuring TACACS+."

Configuring AAA Scalability for PPP Requests

You can configure and monitor the number of background processes allocated by the PPP manager in the network access server (NAS) to deal with AAA authentication and authorization requests. In previous Cisco IOS releases, only one background process was allocated to handle all AAA requests for PPP. This meant that parallelism in AAA servers could not be fully exploited. The AAA Scalability feature enables you to configure the number of processes used to handle AAA requests for PPP, thus increasing the number of users that can be simultaneously authenticated or authorized.

To allocate a specific number of background processes to handle AAA requests for PPP, use the following command in global configuration mode:

Command	Purpose
`Router(config)# aaa processes number`	Allocates a specific number of background processes to handle AAA authentication and authorization requests for PPP.

The argument *number* defines the number of background processes earmarked to process AAA authentication and authorization requests for PPP and can be configured for any value from 1 to 2147483647. Because of the way the PPP manager handles requests for PPP, this argument also defines the number of new users that can be simultaneously authenticated. This argument can be increased or decreased at any time.

Note Allocating additional background processes can be expensive. You should configure the minimum number of background processes capable of handling the AAA requests for PPP.

Configuring ARAP Authentication Using AAA

With the **aaa authentication arap** command, you create one or more lists of authentication methods that are tried when AppleTalk Remote Access Protocol (ARAP) users attempt to log in to the router. These lists are used with the **arap authentication** line configuration command.

Use the following commands starting in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables AAA globally.
Step 2	`Router(config)# aaa authentication arap {default | list-name} method1 [method2...]`	Enables authentication for ARAP users.
Step 3	`Router(config)# line number`	(Optional) Changes to line configuration mode.
Step 4	`Router(config-line)# autoselect arap`	(Optional) Enables autoselection of ARAP.
Step 5	`Router(config-line)# autoselect during-login`	(Optional) Starts the ARAP session automatically at user login.
Step 6	`Router(config-line)# arap authentication list-name`	(Optional—not needed if **default** is used in the **aaa authentication arap** command) Enables TACACS+ authentication for ARAP on a line.

The *list-name* is any character string used to name the list you are creating. The method argument refers to the actual list of methods the authentication algorithm tries, in the sequence entered.

To create a default list that is used when a named list is *not* specified in the **arap authentication** command, use the **default** keyword followed by the methods you want to be used in default situations.

The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

Note Because **none** allows all users logging in to authenticate successfully, it should be used as a backup method of authentication.

Table 6 lists the supported login authentication methods.

Table 6 AAA Authentication ARAP Methods

Keyword	Description
auth-guest	Allows guest logins only if the user has already logged in to EXEC.
guest	Allows guest logins.
line	Uses the line password for authentication.
local	Uses the local username database for authentication.
local-case	Uses case-sensitive local username authentication.
group radius	Uses the list of all RADIUS servers for authentication.
group tacacs+	Uses the list of all TACACS+ servers for authentication.
group *group-name*	Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the **aaa group server radius** or **aaa group server tacacs+** command.

For example, to create a default AAA authentication method list used with ARAP, enter the following command:

```
aaa authentication arap default if-needed none
```

To create the same authentication method list for ARAP but name the list *MIS-access,* enter the following command:

```
aaa authentication arap MIS-access if-needed none
```

This section includes the following sections:

- ARAP Authentication Allowing Authorized Guest Logins
- ARAP Authentication Allowing Guest Logins
- ARAP Authentication Using Line Password
- ARAP Authentication Using Local Password
- ARAP Authentication Using Group RADIUS
- ARAP Authentication Using Group TACACS+
- ARAP Authentication Using Group group-name

ARAP Authentication Allowing Authorized Guest Logins

Use the **aaa authentication arap** command with the **auth-guest** keyword to allow guest logins only if the user has already successfully logged in to the EXEC. This method must be the first listed in the ARAP authentication method list but it can be followed by other methods if it does not succeed. For example, to allow all authorized guest logins—meaning logins by users who have already successfully logged in to the EXEC—as the default method of authentication, using RADIUS only if that method fails, enter the following command:

```
aaa authentication arap default auth-guest group radius
```

For more information about ARAP authorized guest logins, refer to the chapter "Configuring AppleTalk" in the *Cisco IOS AppleTalk and Novell IPX Configuration Guide*.

Note By default, guest logins through ARAP are disabled when you initialize AAA. To allow guest logins, you must use the **aaa authentication arap** command with either the **guest** or the **auth-guest** keyword.

ARAP Authentication Allowing Guest Logins

Use the **aaa authentication arap** command with the **guest** keyword to allow guest logins. This method must be the first listed in the ARAP authentication method list but it can be followed by other methods if it does not succeed. For example, to allow all guest logins as the default method of authentication, using RADIUS only if that method fails, enter the following command:

```
aaa authentication arap default guest group radius
```

For more information about ARAP guest logins, refer to the chapter "Configuring AppleTalk" in the *Cisco IOS AppleTalk and Novell IPX Configuration Guide*.

ARAP Authentication Using Line Password

Use the **aaa authentication arap** command with the *method* keyword **line** to specify the line password as the authentication method. For example, to specify the line password as the method of ARAP user authentication when no other method list has been defined, enter the following command:

```
aaa authentication arap default line
```

Before you can use a line password as the ARAP authentication method, you need to define a line password. For more information about defining line passwords, refer to the section "Configuring Line Password Protection" in this chapter.

ARAP Authentication Using Local Password

Use the **aaa authentication arap** command with the *method* keyword **local** to specify that the Cisco router or access server will use the local username database for authentication. For example, to specify the local username database as the method of ARAP user authentication when no other method list has been defined, enter the following command:

```
aaa authentication arap default local
```

For information about adding users to the local username database, refer to the section "Establishing Username Authentication" in this chapter.

ARAP Authentication Using Group RADIUS

Use the **aaa authentication arap** command with the **group radius** *method* to specify RADIUS as the ARAP authentication method. For example, to specify RADIUS as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication arap default group radius
```

Before you can use RADIUS as the ARAP authentication method, you need to enable communication with the RADIUS security server. For more information about establishing communication with a RADIUS server, refer to the chapter "Configuring RADIUS."

ARAP Authentication Using Group TACACS+

Use the **aaa authentication arap** command with the **group tacacs+** method to specify TACACS+ as the ARAP authentication method. For example, to specify TACACS+ as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication arap default group tacacs+
```

Before you can use TACACS+ as the ARAP authentication method, you need to enable communication with the TACACS+ security server. For more information about establishing communication with a TACACS+ server, refer to the chapter "Configuring TACACS+."

ARAP Authentication Using Group group-name

Use the **aaa authentication arap** command with the **group** *group-name* method to specify a subset of RADIUS or TACACS+ servers to use as the ARAP authentication method. To specify and define the group name and the members of the group, use the **aaa group server** command. For example, use the **aaa group server** command to first define the members of **group araprad**:

```
aaa group server radius araprad
 server 172.16.2.3
 server 172.16.2 17
 server 172.16.2.32
```

This command specifies RADIUS servers 172.16.2.3, 172.16.2.17, and 172.16.2.32 as members of the group *araprad*.

To specify **group araprad** as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication arap default group araprad
```

Before you can use a group name as the ARAP authentication method, you need to enable communication with the RADIUS or TACACS+ security server. For more information about establishing communication with a RADIUS server, refer to the chapter "Configuring RADIUS." For more information about establishing communication with a TACACS+ server, refer to the chapter "Configuring TACACS+."

Configuring NASI Authentication Using AAA

With the **aaa authentication nasi** command, you create one or more lists of authentication methods that are tried when NetWare Asynchronous Services Interface (NASI) users attempt to log in to the router. These lists are used with the **nasi authentication** line configuration command.

To configure NASI authentication using AAA, use the following commands starting in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables AAA globally.
Step 2	`Router(config)# aaa authentication nasi {default | list-name} method1 [method2...]`	Enables authentication for NASI users.
Step 3	`Router(config)# line number`	(Optional—not needed if **default** is used in the **aaa authentication nasi** command) Enters line configuration mode.
Step 4	`Router(config-line)# nasi authentication list-name`	(Optional—not needed if **default** is used in the **aaa authentication nasi** command) Enables authentication for NASI on a line.

The *list-name* is any character string used to name the list you are creating. The method argument refers to the actual list of methods the authentication algorithm tries, in the sequence entered.

To create a default list that is used when a named list is *not* specified in the **aaa authentication nasi** command, use the **default** keyword followed by the methods you want to be used in default situations.

The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

Note Because **none** allows all users logging in to authenticate successfully, it should be used as a backup method of authentication.

Table 7 lists the supported NASI authentication methods.

Table 7 AAA Authentication NASI Methods

Keyword	Description
enable	Uses the enable password for authentication.
line	Uses the line password for authentication.
local	Uses the local username database for authentication.
local-case	Uses case-sensitive local username authentication.
none	Uses no authentication.
group radius	Uses the list of all RADIUS servers for authentication.
group tacacs+	Uses the list of all TACACS+ servers for authentication.
group *group-name*	Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the **aaa group server radius** or **aaa group server tacacs+** command.

This section includes the following sections:

- NASI Authentication Using Enable Password
- NASI Authentication Using Line Password
- NASI Authentication Using Local Password
- NASI Authentication Using Group RADIUS
- NASI Authentication Using Group TACACS+
- NASI Authentication Using group group-name

NASI Authentication Using Enable Password

Use the **aaa authentication nasi** command with the *method* keyword **enable** to specify the enable password as the authentication method. For example, to specify the enable password as the method of NASI user authentication when no other method list has been defined, enter the following command:

```
aaa authentication nasi default enable
```

Before you can use the enable password as the authentication method, you need to define the enable password. For more information about defining enable passwords, refer to the chapter "Configuring Passwords and Privileges."

NASI Authentication Using Line Password

Use the **aaa authentication nasi** command with the *method* keyword **line** to specify the line password as the authentication method. For example, to specify the line password as the method of NASI user authentication when no other method list has been defined, enter the following command:

```
aaa authentication nasi default line
```

Before you can use a line password as the NASI authentication method, you need to define a line password. For more information about defining line passwords, refer to the section "Configuring Line Password Protection" in this chapter.

NASI Authentication Using Local Password

Use the **aaa authentication nasi** command with the *method* keyword **local** to specify that the Cisco router or access server will use the local username database for authentication information. For example, to specify the local username database as the method of NASI user authentication when no other method list has been defined, enter the following command:

```
aaa authentication nasi default local
```

For information about adding users to the local username database, refer to the section "Establishing Username Authentication" in this chapter.

NASI Authentication Using Group RADIUS

Use the **aaa authentication nasi** command with the **group radius** *method* to specify RADIUS as the NASI authentication method. For example, to specify RADIUS as the method of NASI user authentication when no other method list has been defined, enter the following command:

```
aaa authentication nasi default group radius
```

Before you can use RADIUS as the NASI authentication method, you need to enable communication with the RADIUS security server. For more information about establishing communication with a RADIUS server, refer to the chapter "Configuring RADIUS."

NASI Authentication Using Group TACACS+

Use the **aaa authentication nasi** command with the **group tacacs+** *method* keyword to specify TACACS+ as the NASI authentication method. For example, to specify TACACS+ as the method of NASI user authentication when no other method list has been defined, enter the following command:

```
aaa authentication nasi default group tacacs+
```

Before you can use TACACS+ as the authentication method, you need to enable communication with the TACACS+ security server. For more information about establishing communication with a TACACS+ server, refer to the chapter "Configuring TACACS+."

NASI Authentication Using group group-name

Use the **aaa authentication nasi** command with the **group** *group-name* method to specify a subset of RADIUS or TACACS+ servers to use as the NASI authentication method. To specify and define the group name and the members of the group, use the **aaa group server** command. For example, use the **aaa group server** command to first define the members of **group nasirad**:

```
aaa group server radius nasirad
 server 172.16.2.3
 server 172.16.2 17
 server 172.16.2.32
```

This command specifies RADIUS servers 172.16.2.3, 172.16.2.17, and 172.16.2.32 as members of the group *nasirad*.

To specify **group nasirad** as the method of user authentication at login when no other method list has been defined, enter the following command:

```
aaa authentication nasi default group nasirad
```

Before you can use a group name as the NASI authentication method, you need to enable communication with the RADIUS or TACACS+ security server. For more information about establishing communication with a RADIUS server, refer to the chapter "Configuring RADIUS". For more information about establishing communication with a TACACS+ server, refer to the chapter "Configuring TACACS+."

Specifying the Amount of Time for Login Input

The **timeout login response** command allows you to specify how long the system will wait for login input (such as username and password) before timing out. The default login value is 30 seconds; with the **timeout login response** command, you can specify a timeout value from 1 to 300 seconds. To change the login timeout value from the default of 30 seconds, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# timeout login response seconds`	Specifies how long the system will wait for login information before timing out.

Enabling Password Protection at the Privileged Level

Use the **aaa authentication enable default** command to create a series of authentication methods that are used to determine whether a user can access the privileged EXEC command level. You can specify up to four authentication methods. The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

Use the following command in global configuration mode:

Command	Purpose
`Router(config)# aaa authentication enable default method1 [method2...]`	Enables user ID and password checking for users requesting privileged EXEC level. **Note** All **aaa authentication enable default** requests sent by the router to a RADIUS server include the username "$enab15$." Requests sent to a TACACS+ server will include the username that is entered for login authentication.

The method argument refers to the actual list of methods the authentication algorithm tries, in the sequence entered. Table 8 lists the supported enable authentication methods.

Table 8 AAA Authentication Enable Default Methods

Keyword	Description
enable	Uses the enable password for authentication.
line	Uses the line password for authentication.
none	Uses no authentication.
group radius	Uses the list of all RADIUS hosts for authentication. **Note** The RADIUS method does not work on a per-username basis.
group tacacs+	Uses the list of all TACACS+ hosts for authentication.
group *group-name*	Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the **aaa group server radius** or **aaa group server tacacs+** command.

Changing the Text Displayed at the Password Prompt

Use the **aaa authentication password-prompt** command to change the default text that the Cisco IOS software displays when prompting a user to enter a password. This command changes the password prompt for the enable password as well as for login passwords that are not supplied by remote security servers. The **no** form of this command returns the password prompt to the following default value:

```
Password:
```

The **aaa authentication password-prompt** command does not change any dialog that is supplied by a remote TACACS+ or RADIUS server.

The **aaa authentication password-prompt** command works when RADIUS is used as the login method. You will be able to see the password prompt defined in the command shown even when the RADIUS server is unreachable. The **aaa authentication password-prompt** command does not work with TACACS+. TACACS+ supplies the NAS with the password prompt to display to the users. If the TACACS+ server is reachable, the NAS gets the password prompt from the server and uses that prompt instead of the one defined in the **aaa authentication password-prompt** command. If the TACACS+ server is not reachable, the password prompt defined in the **aaa authentication password-prompt** command may be used.

Use the following command in global configuration mode:

Command	Purpose
`Router(config)# aaa authentication password-prompt` *text-string*	Changes the default text displayed when a user is prompted to enter a password.

Configuring Message Banners for AAA Authentication

AAA supports the use of configurable, personalized login and failed-login banners. You can configure message banners that will be displayed when a user logs in to the system to be authenticated using AAA and when, for whatever reason, authentication fails.

This section includes the following sections:

- Configuring a Login Banner
- Configuring a Failed-Login Banner

Configuring a Login Banner

To create a login banner, you need to configure a delimiting character, which notifies the system that the following text string is to be displayed as the banner, and then the text string itself. The delimiting character is repeated at the end of the text string to signify the end of the banner. The delimiting character can be any single character in the extended ASCII character set, but once defined as the delimiter, that character cannot be used in the text string making up the banner.

To configure a banner that will be displayed whenever a user logs in (replacing the default message for login), use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables AAA.
Step 2	`Router(config)# aaa authentication banner` *`delimiter string delimiter`*	Creates a personalized login banner.

The maximum number of characters that can be displayed in the login banner is 2996 characters.

Configuring a Failed-Login Banner

To create a failed-login banner, you need to configure a delimiting character, which notifies the system that the following text string is to be displayed as the banner, and then the text string itself. The delimiting character is repeated at the end of the text string to signify the end of the failed-login banner. The delimiting character can be any single character in the extended ASCII character set, but once defined as the delimiter, that character cannot be used in the text string making up the banner.

To configure a message that will be displayed whenever a user fails login (replacing the default message for failed login), use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables AAA.
Step 2	`Router(config)# aaa authentication fail-message` *`delimiter string delimiter`*	Creates a message to be displayed when a user fails login.

The maximum number of characters that can be displayed in the failed-login banner is 2996 characters.

Configuring AAA Packet of Disconnect

Packet of disconnect (POD) terminates connections on the network access server (NAS) when particular session attributes are identified. By using session information obtained from AAA, the POD client residing on a UNIX workstation sends disconnect packets to the POD server running on the network access server. The NAS terminates any inbound user session with one or more matching key attributes. It rejects requests when required fields are missing or when an exact match is not found.

To configure POD, perform the following tasks in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa accounting network default start-stop radius`	Enables AAA accounting records.
Step 2	`Router(config)# aaa accounting delay-start`	(Optional) Delays generation of the start accounting record until the Framed-IP-Address is assigned, allowing its use in the POD packet.
Step 3	`Router(config)# aaa pod server server-key` *string*	Enables POD reception.
Step 4	`Router(config)# radius-server host` *IP address* `non-standard`	Declares a RADIUS host that uses a vendor-proprietary version of RADIUS.

Enabling Double Authentication

Previously, PPP sessions could only be authenticated by using a single authentication method: either PAP or CHAP. Double authentication requires remote users to pass a second stage of authentication—after CHAP or PAP authentication—before gaining network access.

This second ("double") authentication requires a password that is known to the user but *not* stored on the user's remote host. Therefore, the second authentication is specific to a user, not to a host. This provides an additional level of security that will be effective even if information from the remote host is stolen. In addition, this also provides greater flexibility by allowing customized network privileges for each user.

The second stage authentication can use one-time passwords such as token card passwords, which are not supported by CHAP. If one-time passwords are used, a stolen user password is of no use to the perpetrator.

This section includes the following subsections:

- How Double Authentication Works
- Configuring Double Authentication
- Accessing the User Profile After Double Authentication

How Double Authentication Works

With double authentication, there are two authentication/authorization stages. These two stages occur after a remote user dials in and a PPP session is initiated.

In the first stage, the user logs in using the remote host name; CHAP (or PAP) authenticates the remote host, and then PPP negotiates with AAA to authorize the remote host. In this process, the network access privileges associated with the remote host are assigned to the user.

Note We suggest that the network administrator restrict authorization at this first stage to allow only Telnet connections to the local host.

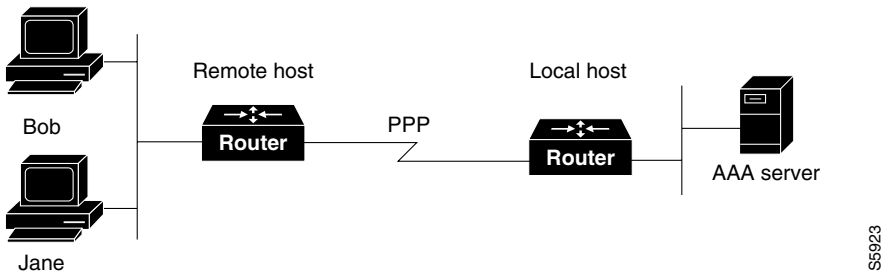
In the second stage, the remote user must Telnet to the network access server to be authenticated. When the remote user logs in, the user must be authenticated with AAA login authentication. The user then must enter the **access-profile** command to be reauthorized using AAA. When this authorization is complete, the user has been double authenticated, and can access the network according to per-user network privileges.

The system administrator determines what network privileges remote users will have after each stage of authentication by configuring appropriate parameters on a security server. To use double authentication, the user must activate it by issuing the **access-profile** command.

Caution Double authentication can cause certain undesirable events if multiple hosts share a PPP connection to a network access server, as shown in Figure 3.

First, if a user, Bob, initiates a PPP session and activates double authentication at the network access server (per Figure 3), any other user will automatically have the same network privileges as Bob until Bob's PPP session expires. This happens because Bob's authorization profile is applied to the network access server's interface during the PPP session and any PPP traffic from other users will use the PPP session Bob established.
Second, if Bob initiates a PPP session and activates double authentication, and then—before Bob's PPP session has expired—another user, Jane, executes the **access-profile** command (or, if Jane Telnets to the network access server and **autocommand access-profile** is executed), a reauthorization will occur and Jane's authorization profile will be applied to the interface—replacing Bob's profile. This can disrupt or halt Bob's PPP traffic, or grant Bob additional authorization privileges Bob should not have.

Figure 3 Possibly Risky Topology: Multiple Hosts Share a PPP Connection to a Network Access Server



Configuring Double Authentication

To configure double authentication, you must complete the following steps:

1. Enable AAA by using the **aaa-new model** global configuration command. For more information about enabling AAA, refer to the chapter "AAA Overview."
2. Use the **aaa authentication** command to configure your network access server to use login and PPP authentication method lists, then apply those method lists to the appropriate lines or interfaces.

3. Use the **aaa authorization** command to configure AAA network authorization at login. For more information about configuring network authorization, refer to the "Configuring Authorization" chapter.
4. Configure security protocol parameters (for example, RADIUS or TACACS+). For more information about RADIUS, refer to the chapter "Configuring RADIUS". For more information about TACACS+, refer to the chapter "Configuring TACACS+."
5. Use access control list AV pairs on the security server that the user can connect to the local host only by establishing a Telnet connection.
6. (Optional) Configure the **access-profile** command as an autocommand. If you configure the autocommand, remote users will not have to manually enter the **access-profile** command to access authorized rights associated with their personal user profile. To learn about configuring autocommands, refer to the **autocommand** command in the *Cisco IOS Dial Technologies Command Reference: Network Services.*

Note If the **access-profile** command is configured as an autocommand, users will still have to Telnet to the local host and log in to complete double authentication.

Follow these rules when creating the user-specific authorization statements (These rules relate to the default behavior of the **access-profile** command):

- Use valid AV pairs when configuring access control list AV pairs on the security server. For a list of valid AV pairs, refer to the chapter "Authentication Commands" in the *Cisco IOS Security Command Reference*.
- If you want remote users to use the interface's existing authorization (that which existed prior to the second stage authentication/authorization), but you want them to have different access control lists (ACLs), you should specify *only* ACL AV pairs in the user-specific authorization definition. This might be desirable if you set up a default authorization profile to apply to the remote host, but want to apply specific ACLs to specific users.
- When these user-specific authorization statements are later applied to the interface, they can either be *added to* the existing interface configuration or they can *replace* the existing interface configuration—depending on which form of the **access-profile** command is used to authorize the user. You should understand how the **access-profile** command works before configuring the authorization statements.
- If you will be using ISDN or Multilink PPP, you must also configure virtual templates at the local host.

To troubleshoot double authentication, use the **debug aaa per-user** debug command. For more information about this command, refer to the *Cisco IOS Debug Command Reference*.

Accessing the User Profile After Double Authentication

In double authentication, when a remote user establishes a PPP link to the local host using the local host name, the remote host is CHAP (or PAP) authenticated. After CHAP (or PAP) authentication, PPP negotiates with AAA to assign network access privileges associated with the remote host to the user. (We suggest that privileges at this stage be restricted to allow the user to connect to the local host only by establishing a Telnet connection.)

When the user needs to initiate the second phase of double authentication, establishing a Telnet connection to the local host, the user enters a personal username and password (different from the CHAP or PAP username and password). This action causes AAA reauthentication to occur according to the

personal username/password. The initial rights associated with the local host, though, are still in place. By using the **access-profile** command, the rights associated with the local host are replaced by or merged with those defined for the user in the user's profile.

To access the user profile after double authentication, use the following command in EXEC configuration mode:

Command	Purpose
`Router> access-profile [merge | replace] [ignore-sanity-checks]`	Accesses the rights associated for the user after double authentication.

If you configured the **access-profile** command to be executed as an autocommand, it will be executed automatically after the remote user logs in.

Enabling Automated Double Authentication

You can make the double authentication process easier for users by implementing automated double authentication. Automated double authentication provides all of the security benefits of double authentication, but offers a simpler, more user-friendly interface for remote users. With double authentication, a second level of user authentication is achieved when the user Telnets to the network access server or router and enters a username and password. With automated double authentication, the user does not have to Telnet to the network access server; instead the user responds to a dialog box that requests a username and password or personal identification number (PIN). To use the automated double authentication feature, the remote user hosts must be running a companion client application. As of Cisco IOS Release 12.0, the only client application software available is the Glacier Bay application server software for PCs.

Note Automated double authentication, like the existing double authentication feature, is for Multilink PPP ISDN connections only. Automated double authentication cannot be used with other protocols such as X.25 or SLIP.

Automated double authentication is an enhancement to the existing double authentication feature. To configure automated double authentication, you must first configure double authentication by completing the following steps:

1. Enable AAA by using the **aaa-new model** global configuration command. For more information about enabling AAA, refer to the chapter "AAA Overview."
2. Use the **aaa authentication** command to configure your network access server to use login and PPP authentication method lists, then apply those method lists to the appropriate lines or interfaces.
3. Use the **aaa authorization** command to configure AAA network authorization at login. For more information about configuring network authorization, refer to the chapter "Configuring Authorization."
4. Configure security protocol parameters (for example, RADIUS or TACACS+). For more information about RADIUS, refer to the chapter "Configuring RADIUS". For more information about TACACS+, refer to the chapter "Configuring TACACS+."

5. Use access control list AV pairs on the security server that the user can connect to the local host only by establishing a Telnet connection.
6. Configure the **access-profile** command as an autocommand. If you configure the autocommand, remote users will not have to manually enter the **access-profile** command to access authorized rights associated with their personal user profile. To learn about configuring autocommands, refer to the **autocommand** command in the *Cisco IOS Dial Technologies Command Reference*, Release 12.2.

Note If the **access-profile** command is configured as an autocommand, users will still have to Telnet to the local host and log in to complete double authentication.

Follow these rules when creating the user-specific authorization statements (These rules relate to the default behavior of the **access-profile** command):

- Use valid AV pairs when configuring access control list AV pairs on the security server. For a list of valid AV pairs, refer to the chapter "Authentication Commands" in the *Cisco IOS Security Command Reference*.
- If you want remote users to use the interface's existing authorization (that which existed prior to the second stage authentication/authorization), but you want them to have different access control lists (ACLs), you should specify *only* ACL AV pairs in the user-specific authorization definition. This might be desirable if you set up a default authorization profile to apply to the remote host, but want to apply specific ACLs to specific users.
- When these user-specific authorization statements are later applied to the interface, they can either be *added to* the existing interface configuration, or *replace* the existing interface configuration—depending on which form of the **access-profile** command is used to authorize the user. You should understand how the **access-profile** command works before configuring the authorization statements.
- If you will be using ISDN or Multilink PPP, you must also configure virtual templates at the local host.

To troubleshoot double authentication, use the **debug aaa per-user** debug command. For more information about this command, refer to the *Cisco IOS Debug Command Reference*.

After you have configured double authentication, you are ready to configure the automation enhancement.

To configure automated double authentication, use the following commands, starting in global configuration mode.

:

	Command	Purpose
Step 1	`Router(config)# ip trigger-authentication [timeout seconds] [port number]`	Enables automation of double authentication.
Step 2	`Router(config)# interface bri number` or `Router(config)# interface serial number:23`	Selects an ISDN BRI or ISDN PRI interface and enter the interface configuration mode.
Step 3	`Router(config-if)# ip trigger-authentication`	Applies automated double authentication to the interface.

To troubleshoot automated double authentication, use the following commands in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# show ip trigger-authentication`	Displays the list of remote hosts for which automated double authentication has been attempted (successfully or unsuccessfully).
Step 2	`Router# clear ip trigger-authentication`	Clears the list of remote hosts for which automated double authentication has been attempted. (This clears the table displayed by the **show ip trigger-authentication** command.)
Step 3	`Router# debug ip trigger-authentication`	Displays **debug** output related to automated double authentication.

Non-AAA Authentication Methods

This section discusses the following non-AAA authentication tasks:

- Configuring Line Password Protection
- Establishing Username Authentication
- Enabling CHAP or PAP Authentication
- Using MS-CHAP

Configuring Line Password Protection

You can provide access control on a terminal line by entering the password and establishing password checking. To do so, use the following commands in line configuration mode:

	Command	Purpose
Step 1	`Router(config-line)# password` *password*	Assigns a password to a terminal or other device on a line.
Step 2	`Router(config-line)# login`	Enables password checking at login.

The password checker is case sensitive and can include spaces; for example, the password “Secret” is different from the password “secret,” and “two words” is an acceptable password.

You can disable line password verification by disabling password checking. To do so, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# no login`	Disables password checking or allow access to a line without password verification.

If you configure line password protection and then configure TACACS or extended TACACS, the TACACS username and password take precedence over line passwords. If you have not yet implemented a security policy, we recommend that you use AAA.

Note The **login** command only changes username and privilege level but it does not execute a shell; therefore autocommands will not be executed. To execute autocommands under this circumstance, you need to establish a Telnet session back into the router (loop-back). Make sure that the router has been configured for secure Telnet sessions if you choose to implement autocommands this way.

Establishing Username Authentication

You can create a username-based authentication system, which is useful in the following situations:

- To provide a TACACS-like username and encrypted password-authentication system for networks that cannot support TACACS
- To provide special-case logins: for example, access list verification, no password verification, autocommand execution at login, and "no escape" situations

To establish username authentication, use the following commands in global configuration mode as needed for your system configuration:

	Command	Purpose
Step 1	`Router(config)# username` *name* [**nopassword** | **password** *password* | **password** *encryption-type encrypted password*]	Establishes username authentication with encrypted passwords.
	or	or
	`Router(config)# username` *name* [**access-class** *number*]	(Optional) Establishes username authentication by access list.
Step 2	`Router(config)# username` *name* [**privilege** *level*]	(Optional) Sets the privilege level for the user.
Step 3	`Router(config)# username` *name* [**autocommand** *command*]	(Optional) Specifies a command to be executed automatically.
Step 4	`Router(config)# username` *name* [**noescape**] [**nohangup**]	(Optional) Sets a "no escape" login environment.

The keyword **noescape** prevents users from using escape characters on the hosts to which they are connected. The **nohangup** feature does not disconnect after using the autocommand.

Caution Passwords will be displayed in clear text in your configuration unless you enable the **service password-encryption** command. For more information about the **service password-encryption** command, refer to the chapter "Passwords and Privileges Commands" in the *Cisco IOS Security Command Reference*.

Enabling CHAP or PAP Authentication

One of the most common transport protocols used in Internet service providers' (ISPs') dial solutions is the Point-to-Point Protocol (PPP). Traditionally, remote users dial in to an access server to initiate a PPP session. After PPP has been negotiated, remote users are connected to the ISP network and to the Internet.

Because ISPs want only customers to connect to their access servers, remote users are required to authenticate to the access server before they can start up a PPP session. Normally, a remote user authenticates by typing in a username and password when prompted by the access server. Although this is a workable solution, it is difficult to administer and awkward for the remote user.

A better solution is to use the authentication protocols built into PPP. In this case, the remote user dials in to the access server and starts up a minimal subset of PPP with the access server. This does not give the remote user access to the ISP's network—it merely allows the access server to talk to the remote device.

PPP currently supports two authentication protocols: Password Authentication Protocol (PAP) and Challenge Handshake Authentication Protocol (CHAP). Both are specified in RFC 1334 and are supported on synchronous and asynchronous interfaces. Authentication via PAP or CHAP is equivalent to typing in a username and password when prompted by the server. CHAP is considered to be more secure because the remote user's password is never sent across the connection.

PPP (with or without PAP or CHAP authentication) is also supported in dialout solutions. An access server utilizes a dialout feature when it initiates a call to a remote device and attempts to start up a transport protocol such as PPP.

See the chapter "Configuring Interfaces" in the *Cisco IOS Configuration Fundamentals Configuration Guide* for more information about CHAP and PAP.

Note To use CHAP or PAP, you must be running PPP encapsulation.

When CHAP is enabled on an interface and a remote device attempts to connect to it, the access server sends a CHAP packet to the remote device. The CHAP packet requests or "challenges" the remote device to respond. The challenge packet consists of an ID, a random number, and the host name of the local router.

When the remote device receives the challenge packet, it concatenates the ID, the remote device's password, and the random number, and then encrypts all of it using the remote device's password. The remote device sends the results back to the access server, along with the name associated with the password used in the encryption process.

When the access server receives the response, it uses the name it received to retrieve a password stored in its user database. The retrieved password should be the same password the remote device used in its encryption process. The access server then encrypts the concatenated information with the newly retrieved password—if the result matches the result sent in the response packet, authentication succeeds.

The benefit of using CHAP authentication is that the remote device's password is never transmitted in clear text. This prevents other devices from stealing it and gaining illegal access to the ISP's network.

CHAP transactions occur only at the time a link is established. The access server does not request a password during the rest of the call. (The local device can, however, respond to such requests from other devices during a call.)

When PAP is enabled, the remote router attempting to connect to the access server is required to send an authentication request. If the username and password specified in the authentication request are accepted, the Cisco IOS software sends an authentication acknowledgment.

After you have enabled CHAP or PAP, the access server will require authentication from remote devices dialing in to the access server. If the remote device does not support the enabled protocol, the call will be dropped.

To use CHAP or PAP, you must perform the following tasks:

1. Enable PPP encapsulation.
2. Enable CHAP or PAP on the interface.
3. For CHAP, configure host name authentication and the secret or password for each remote system with which authentication is required.

This section includes the following sections:

- Enabling PPP Encapsulation
- Enabling PAP or CHAP
- Inbound and Outbound Authentication
- Enabling Outbound PAP Authentication
- Refusing PAP Authentication Requests
- Creating a Common CHAP Password
- Refusing CHAP Authentication Requests
- Delaying CHAP Authentication Until Peer Authenticates

Enabling PPP Encapsulation

To enable PPP encapsulation, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# encapsulation ppp`	Enables PPP on an interface.

Enabling PAP or CHAP

To enable CHAP or PAP authentication on an interface configured for PPP encapsulation, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp authentication {protocol1 [protocol2...]} [if-needed] {default | list-name} [callin] [one-time]`	Defines the authentication protocols supported and the order in which they are used. In this command, *protocol1, protocol2* represent the following protocols: CHAP, MS-CHAP, and PAP. PPP authentication is attempted first using the first authentication method, which is *protocol1*. If *protocol1* is unable to establish authentication, the next configured protocol is used to negotiate authentication.

If you configure **ppp authentication chap** on an interface, all incoming calls on that interface that initiate a PPP connection will have to be authenticated using CHAP; likewise, if you configure **ppp authentication pap**, all incoming calls that start a PPP connection will have to be authenticated via PAP. If you configure **ppp authentication chap pap**, the access server will attempt to authenticate all

incoming calls that start a PPP session with CHAP. If the remote device does not support CHAP, the access server will try to authenticate the call using PAP. If the remote device does not support either CHAP or PAP, authentication will fail and the call will be dropped. If you configure **ppp authentication pap chap**, the access server will attempt to authenticate all incoming calls that start a PPP session with PAP. If the remote device does not support PAP, the access server will try to authenticate the call using CHAP. If the remote device does not support either protocol, authentication will fail and the call will be dropped. If you configure the **ppp authentication** command with the **callin** keyword, the access server will only authenticate the remote device if the remote device initiated the call.

Authentication method lists and the **one-time** keyword are only available if you have enabled AAA—they will not be available if you are using TACACS or extended TACACS. If you specify the name of an authentication method list with the **ppp authentication** command, PPP will attempt to authenticate the connection using the methods defined in the specified method list. If AAA is enabled and no method list is defined by name, PPP will attempt to authenticate the connection using the methods defined as the default. The **ppp authentication** command with the **one-time** keyword enables support for one-time passwords during authentication.

The **if-needed** keyword is only available if you are using TACACS or extended TACACS. The **ppp authentication** command with the **if-needed** keyword means that PPP will only authenticate the remote device via PAP or CHAP if they have not yet authenticated during the life of the current call. If the remote device authenticated via a standard login procedure and initiated PPP from the EXEC prompt, PPP will not authenticate via CHAP if **ppp authentication chap if-needed** is configured on the interface.

Caution If you use a *list-name* that has not been configured with the **aaa authentication ppp** command, you disable PPP on the line.

For information about adding a **username** entry for each remote system from which the local router or access server requires authentication, see the section "Establishing Username Authentication."

Inbound and Outbound Authentication

PPP supports two-way authentication. Normally, when a remote device dials in to an access server, the access server requests that the remote device prove that it is allowed access. This is known as inbound authentication. At the same time, the remote device can also request that the access server prove that it is who it says it is. This is known as outbound authentication. An access server also does outbound authentication when it initiates a call to a remote device.

Enabling Outbound PAP Authentication

To enable outbound PAP authentication, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp pap sent-username username password password`	Enables outbound PAP authentication.

The access server uses the username and password specified by the **ppp pap sent-username** command to authenticate itself whenever it initiates a call to a remote device or when it has to respond to a remote device's request for outbound authentication.

Refusing PAP Authentication Requests

To refuse PAP authentication from peers requesting it, meaning that PAP authentication is disabled for all calls, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp pap refuse`	Refuses PAP authentication from peers requesting PAP authentication.

If the **refuse** keyword is not used, the router will not refuse any PAP authentication challenges received from the peer.

Creating a Common CHAP Password

For remote CHAP authentication only, you can configure your router to create a common CHAP secret password to use in response to challenges from an unknown peer; for example, if your router calls a rotary of routers (either from another vendor, or running an older version of the Cisco IOS software) to which a new (that is, unknown) router has been added. The **ppp chap password** command allows you to replace several username and password configuration commands with a single copy of this command on any dialer interface or asynchronous group interface.

To enable a router calling a collection of routers to configure a common CHAP secret password, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp chap password` *secret*	Enables a router calling a collection of routers to configure a common CHAP secret password.

Refusing CHAP Authentication Requests

To refuse CHAP authentication from peers requesting it, meaning that CHAP authentication is disabled for all calls, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp chap refuse [callin]`	Refuses CHAP authentication from peers requesting CHAP authentication.

If the **callin** keyword is used, the router will refuse to answer CHAP authentication challenges received from the peer, but will still require the peer to answer any CHAP challenges the router sends.

If outbound PAP has been enabled (using the **ppp pap sent-username** command), PAP will be suggested as the authentication method in the refusal packet.

Delaying CHAP Authentication Until Peer Authenticates

To specify that the router will not authenticate to a peer requesting CHAP authentication until after the peer has authenticated itself to the router, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp chap wait secret`	Configures the router to delay CHAP authentication until after the peer has authenticated itself to the router.

This command (which is the default) specifies that the router will not authenticate to a peer requesting CHAP authentication until the peer has authenticated itself to the router. The **no ppp chap wait** command specifies that the router will respond immediately to an authentication challenge.

Using MS-CHAP

Microsoft Challenge Handshake Authentication Protocol (MS-CHAP) is the Microsoft version of CHAP and is an extension of RFC 1994. Like the standard version of CHAP, MS-CHAP is used for PPP authentication; in this case, authentication occurs between a PC using Microsoft Windows NT or Microsoft Windows 95 and a Cisco router or access server acting as a network access server.

MS-CHAP differs from the standard CHAP as follows:

- MS-CHAP is enabled by negotiating CHAP Algorithm 0x80 in LCP option 3, Authentication Protocol.
- The MS-CHAP Response packet is in a format designed to be compatible with Microsoft Windows NT 3.5 and 3.51, Microsoft Windows 95, and Microsoft LAN Manager 2.*x*. This format does not require the authenticator to store a clear or reversibly encrypted password.
- MS-CHAP provides an authenticator-controlled authentication retry mechanism.
- MS-CHAP provides an authenticator-controlled change password mechanism.
- MS-CHAP defines a set of “reason-for failure” codes returned in the Failure packet message field.

Depending on the security protocols you have implemented, PPP authentication using MS-CHAP can be used with or without AAA security services. If you have enabled AAA, PPP authentication using MS-CHAP can be used in conjunction with both TACACS+ and RADIUS. Table 9 lists the vendor-specific RADIUS attributes (IETF Attribute 26) that enable RADIUS to support MS-CHAP.

Table 9 Vendor-Specific RADIUS Attributes for MS-CHAP

Vendor-ID Number	Vendor-Type Number	Vendor-Proprietary Attribute	Description
311	11	MSCHAP-Challenge	Contains the challenge sent by a network access server to an MS-CHAP user. It can be used in both Access-Request and Access-Challenge packets.
211	11	MSCHAP-Response	Contains the response value provided by a PPP MS-CHAP user in response to the challenge. It is only used in Access-Request packets. This attribute is identical to the PPP CHAP Identifier.

To define PPP authentication using MS-CHAP, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 2	`Router(config-if)# ppp authentication ms-chap [if-needed] [list-name | default] [callin] [one-time]`	Defines PPP authentication using MS-CHAP.

If you configure **ppp authentication ms-chap** on an interface, all incoming calls on that interface that initiate a PPP connection will have to be authenticated using MS-CHAP. If you configure the **ppp authentication** command with the **callin** keyword, the access server will only authenticate the remote device if the remote device initiated the call.

Authentication method lists and the **one-time** keyword are only available if you have enabled AAA—they will not be available if you are using TACACS or extended TACACS. If you specify the name of an authentication method list with the **ppp authentication** command, PPP will attempt to authenticate the connection using the methods defined in the specified method list. If AAA is enabled and no method list is defined by name, PPP will attempt to authenticate the connection using the methods defined as the default. The **ppp authentication** command with the **one-time** keyword enables support for one-time passwords during authentication.

The **if-needed** keyword is only available if you are using TACACS or extended TACACS. The **ppp authentication** command with the **if-needed** keyword means that PPP will only authenticate the remote device via MS-CHAP if that device has not yet authenticated during the life of the current call. If the remote device authenticated through a standard login procedure and initiated PPP from the EXEC prompt, PPP will not authenticate through MS-CHAP if **ppp authentication chap if-needed** is configured.

Note If PPP authentication using MS-CHAP is used with username authentication, you must include the MS-CHAP secret in the local username/password database. For more information about username authentication, refer to the "Establish Username Authentication" section.

Authentication Examples

The following sections provide authentication configuration examples:

- RADIUS Authentication Examples
- TACACS+ Authentication Examples
- Kerberos Authentication Examples
- AAA Scalability Example
- Login and Failed Banner Examples
- AAA Packet of Disconnect Server Key Example
- Double Authentication Examples
- Automated Double Authentication Example
- MS-CHAP Example

RADIUS Authentication Examples

This section provides two sample configurations using RADIUS.

The following example shows how to configure the router to authenticate and authorize using RADIUS:

```
aaa authentication login radius-login group radius local
aaa authentication ppp radius-ppp if-needed group radius
aaa authorization exec default group radius if-authenticated
aaa authorization network default group radius
line 3
login authentication radius-login
interface serial 0
ppp authentication radius-ppp
```

The lines in this sample RADIUS authentication and authorization configuration are defined as follows:

- The **aaa authentication login radius-login group radius local** command configures the router to use RADIUS for authentication at the login prompt. If RADIUS returns an error, the user is authenticated using the local database.
- The **aaa authentication ppp radius-ppp if-needed group radius** command configures the Cisco IOS software to use PPP authentication using CHAP or PAP if the user has not already logged in. If the EXEC facility has authenticated the user, PPP authentication is not performed.
- The **aaa authorization exec default group radius if-authenticated** command queries the RADIUS database for information that is used during EXEC authorization, such as autocommands and privilege levels, but only provides authorization if the user has successfully authenticated.
- The **aaa authorization network default group radius** command queries RADIUS for network authorization, address assignment, and other access lists.
- The **login authentication radius-login** command enables the radius-login method list for line 3.
- The **ppp authentication radius-ppp** command enables the radius-ppp method list for serial interface 0.

The following example shows how to configure the router to prompt for and verify a username and password, authorize the user's EXEC level, and specify it as the method of authorization for privilege level 2. In this example, if a local username is entered at the username prompt, that username is used for authentication.

If the user is authenticated using the local database, EXEC authorization using RADIUS will fail because no data is saved from the RADIUS authentication. The method list also uses the local database to find an autocommand. If there is no autocommand, the user becomes the EXEC user. If the user then attempts to issue commands that are set at privilege level 2, TACACS+ is used to attempt to authorize the command.

```
aaa authentication login default group radius local
aaa authorization exec default group radius local
aaa authorization command 2 default group tacacs+ if-authenticated
radius-server host 172.16.71.146 auth-port 1645 acct-port 1646
radius-server attribute 44 include-in-access-req
radius-server attribute 8 include-in-access-req
```

The lines in this sample RADIUS authentication and authorization configuration are defined as follows:

- The **aaa authentication login default group radius local** command specifies that the username and password are verified by RADIUS or, if RADIUS is not responding, by the router's local user database.
- The **aaa authorization exec default group radius local** command specifies that RADIUS authentication information be used to set the user's EXEC level if the user authenticates with RADIUS. If no RADIUS information is used, this command specifies that the local user database be used for EXEC authorization.
- The **aaa authorization command 2 default group tacacs+ if-authenticated** command specifies TACACS+ authorization for commands set at privilege level 2, if the user has already successfully authenticated.
- The **radius-server host 172.16.71.146 auth-port 1645 acct-port 1646** command specifies the IP address of the RADIUS server host, the UDP destination port for authentication requests, and the UDP destination port for accounting requests.
- The **radius-server attribute 44 include-in-access-req** command sends RADIUS attribute 44 (Acct-Seccion-ID) in access-request packets.
- The **radius-server attribute 8 include-in-access-req** command sends RADIUS attribute 8 (Framed-IP-Address) in access-request packets.

TACACS+ Authentication Examples

The following example shows how to configure TACACS+ as the security protocol to be used for PPP authentication:

```
aaa new-model
aaa authentication ppp test group tacacs+ local
interface serial 0
ppp authentication chap pap test
tacacs-server host 10.1.2.3
tacacs-server key goaway
```

The lines in this sample TACACS+ authentication configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, "test," to be used on serial interfaces running PPP. The keywords **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.
- The **interface** command selects the line.
- The **ppp authentication** command applies the test method list to this line.
- The **tacacs-server host** command identifies the TACACS+ daemon as having an IP address of 10.1.2.3.
- The **tacacs-server key** command defines the shared encryption key to be "goaway."

The following example shows how to configure AAA authentication for PPP:

```
aaa authentication ppp default if-needed group tacacs+ local
```

In this example, the keyword **default** means that PPP authentication is applied by default to all interfaces. The **if-needed** keyword means that if the user has already authenticated by going through the ASCII login procedure, then PPP is not necessary and can be skipped. If authentication is needed, the

keywords **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.

The following example shows how to create the same authentication algorithm for PAP, but it calls the method list "MIS-access" instead of "default":

```
aaa authentication ppp MIS-access if-needed group tacacs+ local
interface serial 0
ppp authentication pap MIS-access
```

In this example, because the list does not apply to any interfaces (unlike the default list, which applies automatically to all interfaces), the administrator must select interfaces to which this authentication scheme should apply by using the **interface** command. The administrator must then apply this method list to those interfaces by using the **ppp authentication** command.

Kerberos Authentication Examples

To specify Kerberos as the login authentication method, use the following command:

```
aaa authentication login default krb5
```

To specify Kerberos authentication for PPP, use the following command:

```
aaa authentication ppp default krb5
```

AAA Scalability Example

The following example shows a general security configuration using AAA with RADIUS as the security protocol. In this example, the network access server is configured to allocate 16 background processes to handle AAA requests for PPP.

```
aaa new-model
radius-server host alcatraz
radius-server key myRaDiUSpassWoRd
radius-server configure-nas
username root password ALongPassword
aaa authentication ppp dialins group radius local
aaa authentication login admins local
aaa authorization network default group radius local
aaa accounting network default start-stop group radius
aaa processes 16
line 1 16
 autoselect ppp
 autoselect during-login
 login authentication admins
 modem dialin
interface group-async 1
 group-range 1 16
 encapsulation ppp
 ppp authentication pap dialins
```

The lines in this sample RADIUS AAA configuration are defined as follows:

- The **aaa new-model** command enables AAA network security services.
- The **radius-server host** command defines the name of the RADIUS server host.
- The **radius-server key** command defines the shared secret text string between the network access server and the RADIUS server host.

- The **radius-server configure-nas** command defines that the Cisco router or access server will query the RADIUS server for static routes and IP pool definitions when the device first starts up.
- The **username** command defines the username and password to be used for the PPP Password Authentication Protocol (PAP) caller identification.
- The **aaa authentication ppp dialins group radius local** command defines the authentication method list "dialins," which specifies that RADIUS authentication, then (if the RADIUS server does not respond) local authentication will be used on serial lines using PPP.
- The **aaa authentication login admins local** command defines another method list, "admins," for login authentication.
- The **aaa authorization network default group radius local** command is used to assign an address and other network parameters to the RADIUS user.
- The **aaa accounting network default start-stop group radius** command tracks PPP usage.
- The **aaa processes** command allocates 16 background processes to handle AAA requests for PPP.
- The **line** command switches the configuration mode from global configuration to line configuration and identifies the specific lines being configured.
- The **autoselect ppp** command configures the Cisco IOS software to allow a PPP session to start up automatically on these selected lines.
- The **autoselect during-login** command is used to display the username and password prompt without pressing the Return key. After the user logs in, the autoselect function (in this case, PPP) begins.
- The **login authentication admins** command applies the "admins" method list for login authentication.
- The **modem dialin** command configures modems attached to the selected lines to only accept incoming calls.
- The **interface group-async** command selects and defines an asynchronous interface group.
- The **group-range** command defines the member asynchronous interfaces in the interface group.
- The **encapsulation ppp** command sets PPP as the encapsulation method used on the specified interfaces.
- The **ppp authentication pap dialins** command applies the "dialins" method list to the specified interfaces.

Login and Failed Banner Examples

The following example shows how to configure a login banner (in this case, the phrase "Unauthorized Access Prohibited") that will be displayed when a user logs in to the system. The asterisk (*) is used as the delimiting character. (RADIUS is specified as the default login authentication method.)

```
aaa new-model
aaa authentication banner *Unauthorized Access Prohibited*
aaa authentication login default group radius
```

This configuration produces the following login banner:

```
Unauthorized Access Prohibited
Username:
```

The following example shows how to additionally configure a failed login banner (in this case, the phrase "Failed login. Try again.") that will be displayed when a user tries to log in to the system and fails. The asterisk (*) is used as the delimiting character. (RADIUS is specified as the default login authentication method.)

```
aaa new-model
aaa authentication banner *Unauthorized Access Prohibited*
aaa authentication fail-message *Failed login. Try again.*
aaa authentication login default group radius
```

This configuration produces the following login and failed login banner:

```
Unauthorized Access Prohibited
Username:
Password:
Failed login. Try again.
```

AAA Packet of Disconnect Server Key Example

The following example shows how to configure POD (packet of disconnect), which terminates connections on the network access server (NAS) when particular session attributes are identified.

```
aaa new-model
aaa authentication ppp default radius
aaa accounting network default start-stop radius
aaa accounting delay-start
aaa pod server server-key xyz123
radius-server host 172.16.0.0 non-standard
radius-server key rad123
```

Double Authentication Examples

The examples in this section illustrate possible configurations to be used with double authentication. Your configurations could differ significantly, depending on your network and security requirements.

This section includes the following examples:

- Configuration of the Local Host for AAA with Double Authentication Examples
- Configuration of the AAA Server for First-Stage (PPP) Authentication and Authorization Example
- Configuration of the AAA Server for Second-Stage (Per-User) Authentication and Authorization Examples
- Complete Configuration with TACACS+ Example

Note These configuration examples include specific IP addresses and other specific information. This information is for illustration purposes only: your configuration will use different IP addresses, different usernames and passwords, and different authorization statements.

Configuration of the Local Host for AAA with Double Authentication Examples

These two examples show how to configure a local host to use AAA for PPP and login authentication, and for network and EXEC authorization. One example is shown for RADIUS and one example for TACACS+.

In both examples, the first three lines configure AAA, with a specific server as the AAA server. The next two lines configure AAA for PPP and login authentication, and the last two lines configure network and EXEC authorization. The last line is necessary only if the **access-profile** command will be executed as an autocommand.

The following example shows router configuration with a RADIUS AAA server:

```
aaa new-model
radius-server host secureserver
radius-server key myradiuskey
aaa authentication ppp default group radius
aaa authentication login default group radius
aaa authorization network default group radius
aaa authorization exec default group radius
```

The following example shows router configuration with a TACACS+ server:

```
aaa new-model
tacacs-server host security
tacacs-server key mytacacskey
aaa authentication ppp default group tacacs+
aaa authentication login default group tacacs+
aaa authorization network default group tacacs+
aaa authorization exec default group tacacs+
```

Configuration of the AAA Server for First-Stage (PPP) Authentication and Authorization Example

This example shows a configuration on the AAA server. A partial sample AAA configuration is shown for RADIUS.

TACACS+ servers can be configured similarly. (See the section "Complete Configuration with TACACS+ Example" later in this chapter.)

This example defines authentication/authorization for a remote host named "hostx" that will be authenticated by CHAP in the first stage of double authentication. Note that the ACL AV pair limits the remote host to Telnet connections to the local host. The local host has the IP address 10.0.0.2.

The following example shows a partial AAA server configuration for RADIUS:

```
hostx   Password = "welcome"
        User-Service-Type = Framed-User,
        Framed-Protocol = PPP,
        cisco-avpair = "lcp:interface-config=ip unnumbered ethernet 0",
        cisco-avpair = "ip:inacl#3=permit tcp any 172.21.114.0 0.0.0.255 eq telnet",
        cisco-avpair = "ip:inacl#4=deny icmp any any",
        cisco-avpair = "ip:route#5=55.0.0.0 255.0.0.0",
        cisco-avpair = "ip:route#6=66.0.0.0 255.0.0.0",
        cisco-avpair = "ipx:inacl#3=deny any"
```

Configuration of the AAA Server for Second-Stage (Per-User) Authentication and Authorization Examples

This section contains partial sample AAA configurations on a RADIUS server. These configurations define authentication and authorization for a user (Pat) with the username "patuser," who will be user-authenticated in the second stage of double authentication.

TACACS+ servers can be configured similarly. (See the section "Complete Configuration with TACACS+ Example" later in this chapter.)

Three examples show sample RADIUS AAA configurations that could be used with each of the three forms of the **access-profile** command.

The first example shows a partial sample AAA configuration that works with the default form (no keywords) of the **access-profile** command. Note that only ACL AV pairs are defined. This example also sets up the **access-profile** command as an autocommand.

```
patuser    Password = "welcome"
           User-Service-Type = Shell-User,
           cisco-avpair = "shell:autocmd=access-profile"
           User-Service-Type = Framed-User,
           Framed-Protocol = PPP,
           cisco-avpair = "ip:inacl#3=permit tcp any host 10.0.0.2 eq telnet",
           cisco-avpair = "ip:inacl#4=deny icmp any any"
```

The second example shows a partial sample AAA configuration that works with the **access-profile merge** form of the **access-profile** command. This example also sets up the **access-profile merge** command as an autocommand.

```
patuser    Password = "welcome"
           User-Service-Type = Shell-User,
           cisco-avpair = "shell:autocmd=access-profile merge"
           User-Service-Type = Framed-User,
           Framed-Protocol = PPP,
           cisco-avpair = "ip:inacl#3=permit tcp any any"
           cisco-avpair = "ip:route=10.0.0.0 255.255.0.0",
           cisco-avpair = "ip:route=10.1.0.0 255.255.0.0",
           cisco-avpair = "ip:route=10.2.0.0 255.255.0.0"
```

The third example shows a partial sample AAA configuration that works with the **access-profile replace** form of the **access-profile** command. This example also sets up the **access-profile replace** command as an autocommand.

```
patuser    Password = "welcome"
           User-Service-Type = Shell-User,
           cisco-avpair = "shell:autocmd=access-profile replace"
           User-Service-Type = Framed-User,
           Framed-Protocol = PPP,
           cisco-avpair = "ip:inacl#3=permit tcp any any",
           cisco-avpair = "ip:inacl#4=permit icmp any any",
           cisco-avpair = "ip:route=10.10.0.0 255.255.0.0",
           cisco-avpair = "ip:route=10.11.0.0 255.255.0.0",
           cisco-avpair = "ip:route=10.12.0.0 255.255.0.0"
```

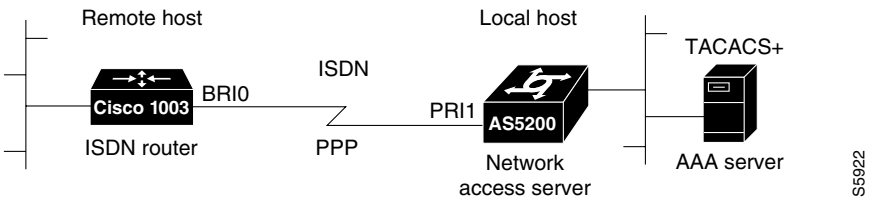
Complete Configuration with TACACS+ Example

This example shows TACACS+ authorization profile configurations both for the remote host (used in the first stage of double authentication) and for specific users (used in the second stage of double authentication). This TACACS+ example contains approximately the same configuration information as shown in the previous RADIUS examples.

This sample configuration shows authentication/authorization profiles on the TACACS+ server for the remote host “hostx” and for three users, with the usernames “pat_default,” “pat_merge,” and “pat_replace.” The configurations for these three usernames illustrate different configurations that correspond to the three different forms of the **access-profile** command. The three user configurations also illustrate setting up the autocommand for each form of the **access-profile** command.

Figure 4 shows the topology. The example that follows the figure shows a TACACS+ configuration file.

Figure 4 Example Topology for Double Authentication



This sample configuration shows authentication/authorization profiles on the TACACS+ server for the remote host “hostx” and for three users, with the usernames “pat_default,” “pat_merge,” and “pat_replace.”

```
key = “mytacacskey”

default authorization = permit


#-----------------------------Remote Host (BRI)-------------------------
#
# This allows the remote host to be authenticated by the local host
# during fist-stage authentication, and provides the remote host
# authorization profile.
#
#-----------------------------------------------------------------------

user = hostx
{
    login = cleartext “welcome”
    chap = cleartext “welcome”


    service = ppp protocol = lcp {
                interface-config=”ip unnumbered ethernet 0"
    }

    service = ppp protocol = ip {
            # It is important to have the hash sign and some string after
            # it. This indicates to the NAS that you have a per-user
            # config.

            inacl#3=”permit tcp any 172.21.114.0 0.0.0.255 eq telnet”
            inacl#4=”deny icmp any any”
```

```
            route#5="55.0.0.0 255.0.0.0"
            route#6="66.0.0.0 255.0.0.0"
    }

    service = ppp protocol = ipx {
            # see previous comment about the hash sign and string, in protocol = ip
            inacl#3="deny any"
    }


}


#------------------- "access-profile" default user "only acls" ------------------
#
# Without arguments, access-profile removes any access-lists it can find
# in the old configuration (both per-user and per-interface), and makes sure
# that the new profile contains ONLY access-list definitions.
#
#-------------------------------------------------------------------------------

user = pat_default
{
        login = cleartext "welcome"
        chap = cleartext "welcome"

        service = exec

        {
                # This is the autocommand that executes when pat_default logs in.
                autocmd = "access-profile"
        }

        service = ppp protocol = ip {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IP
                # access-lists (not even the ones installed prior to
                # this)!

                inacl#3="permit tcp any host 10.0.0.2 eq telnet"
                inacl#4="deny icmp any any"
        }


        service = ppp protocol = ipx {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IPX
                # access-lists (not even the ones installed prior to
                # this)!
        }


}


#---------------------- "access-profile merge" user ---------------------------
#
# With the 'merge' option, first all old access-lists are removed (as before),
# but then (almost) all AV pairs are uploaded and installed. This will allow
# for uploading any custom static routes, sap-filters, and so on, that the user
# may need in his or her profile. This needs to be used with care, as it leaves
# open the possibility of conflicting configurations.
```

```
#
#-------------------------------------------------------------------------

user = pat_merge
{
        login = cleartext "welcome"
        chap = cleartext "welcome"


        service = exec
        {
                # This is the autocommand that executes when pat_merge logs in.
                autocmd = "access-profile merge"
        }

        service = ppp protocol = ip
        {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IP
                # access-lists (not even the ones installed prior to
                # this)!

                inacl#3="permit tcp any any"
                route#2="10.0.0.0 255.255.0.0"
                route#3="10.1.0.0 255.255.0.0"
                route#4="10.2.0.0 255.255.0.0"

        }

        service = ppp protocol = ipx
        {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IPX
                # access-lists (not even the ones installed prior to
                # this)!

        }

}


#---------------------- "access-profile replace" user ----------------------------
#
# With the 'replace' option, ALL old configuration is removed and ALL new
# configuration is installed.
#
# One caveat: access-profile checks the new configuration for address-pool and
# address AV pairs. As addresses cannot be renegotiated at this point, the
# command will fail (and complain) when it encounters such an AV pair.
# Such AV pairs are considered to be "invalid" for this context.
#-------------------------------------------------------------------------


user = pat_replace
{
        login = cleartext "welcome"
        chap = cleartext "welcome"


        service = exec
        {
```

```
                # This is the autocommand that executes when pat_replace logs in.
                autocmd = "access-profile replace"
        }

        service = ppp protocol = ip
        {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IP
                # access-lists (not even the ones installed prior to
                # this)!

                inacl#3="permit tcp any any"
                inacl#4="permit icmp any any"


                route#2="10.10.0.0 255.255.0.0"
                route#3="10.11.0.0 255.255.0.0"
                route#4="10.12.0.0 255.255.0.0"
        }

        service = ppp protocol = ipx
        {
                # put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IPX
                # access-lists (not even the ones installed prior to
                # this)!
        }

}
```

Automated Double Authentication Example

This example shows a complete configuration file for a Cisco 2509 router with automated double authentication configured. The configuration commands that apply to automated double authentication are preceded by descriptions with a double asterisk (**).

```
Current configuration:
!
version 11.3
no service password-encryption
!
hostname myrouter
!
!
! **The following AAA commands are used to configure double authentication:
!
! **The following command enables AAA:
aaa new-model
! **The following command enables user authentication via the TACACS+ AAA server:
aaa authentication login default group tacacs+
aaa authentication login console none
! **The following command enables device authentication via the TACACS+ AAA server:
aaa authentication ppp default group tacacs+
! **The following command causes the remote user's authorization profile to be
! downloaded from the AAA server to the Cisco 2509 router when required:
aaa authorization exec default group tacacs+
! **The following command causes the remote device's authorization profile to be
! downloaded from the AAA server to the Cisco 2509 router when required:
aaa authorization network default group tacacs+
```

```
enable password mypassword
!
ip host blue 172.21.127.226
ip host green 172.21.127.218
ip host red 172.21.127.114
ip domain-name example.com
ip name-server 171.69.2.75
! **The following command globally enables automated double authentication:
ip trigger-authentication timeout 60 port 7500
isdn switch-type basic-5ess
!
!
interface Ethernet0
 ip address 172.21.127.186 255.255.255.248
 no ip route-cache
 no ip mroute-cache
 no keepalive
 ntp disable
 no cdp enable
!
interface Virtual-Template1
 ip unnumbered Ethernet0
 no ip route-cache
 no ip mroute-cache
!
interface Serial0
 ip address 172.21.127.105 255.255.255.248
 encapsulation ppp
 no ip mroute-cache
 no keepalive
 shutdown
 clockrate 2000000
 no cdp enable
!
interface Serial1
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 no cdp enable
!
! **Automated double authentication occurs via the ISDN BRI interface BRI0:
interface BRI0
 ip unnumbered Ethernet0
! **The following command turns on automated double authentication at this interface:
 ip trigger-authentication
! **PPP encapsulation is required:
 encapsulation ppp
 no ip route-cache
 no ip mroute-cache
 dialer idle-timeout 500
 dialer map ip 172.21.127.113 name myrouter 60074
 dialer-group 1
 no cdp enable
```

```
! **The following command specifies that device authentication occurs via PPP CHAP:
 ppp authentication chap
!
router eigrp 109
 network 172.21.0.0
 no auto-summary
!
ip default-gateway 172.21.127.185
no ip classless
ip route 172.21.127.114 255.255.255.255 172.21.127.113
! **Virtual profiles are required for double authentication to work:
virtual-profile virtual-template 1
dialer-list 1 protocol ip permit
no cdp run
! **The following command defines where the TACACS+ AAA server is:
tacacs-server host 171.69.57.35 port 1049
tacacs-server timeout 90
! **The following command defines the key to use with TACACS+ traffic (required):
tacacs-server key mytacacskey
snmp-server community public RO
!
line con 0
 exec-timeout 0 0
 login authentication console
line aux 0
 transport input all
line vty 0 4
 exec-timeout 0 0
 password lab
!
end
```

MS-CHAP Example

The following example shows how to configure a Cisco AS5200 Universal Access Server (enabled for AAA and communication with a RADIUS security server) for PPP authentication using MS-CHAP:

```
aaa new-model
aaa authentication login admins local
aaa authentication ppp dialins group radius local
aaa authorization network default group radius local
aaa accounting network default start-stop group radius

username root password ALongPassword

radius-server host alcatraz
radius-server key myRaDiUSpassWoRd

interface group-async 1
 group-range 1 16
 encapsulation ppp
 ppp authentication ms-chap dialins

line 1 16
 autoselect ppp
 autoselect during-login
 login authentication admins
 modem dialin
```

The lines in this sample RADIUS AAA configuration are defined as follows:

- The **aaa new-model** command enables AAA network security services.
- The **aaa authentication login admins local** command defines another method list, “admins”, for login authentication.
- The **aaa authentication ppp dialins group radius local** command defines the authentication method list “dialins,” which specifies that RADIUS authentication then (if the RADIUS server does not respond) local authentication will be used on serial lines using PPP.
- The **aaa authorization network default group radius local** command is used to assign an address and other network parameters to the RADIUS user.
- The **aaa accounting network default start-stop group radius** command tracks PPP usage.
- The **username** command defines the username and password to be used for the PPP Password Authentication Protocol (PAP) caller identification.
- The **radius-server host** command defines the name of the RADIUS server host.
- The **radius-server key** command defines the shared secret text string between the network access server and the RADIUS server host.
- The **interface group-async** command selects and defines an asynchronous interface group.
- The **group-range** command defines the member asynchronous interfaces in the interface group.
- The **encapsulation ppp** command sets PPP as the encapsulation method used on the specified interfaces.
- The **ppp authentication ms-chap dialins** command selects MS-CHAP as the method of PPP authentication and applies the “dialins” method list to the specified interfaces.
- The **line** command switches the configuration mode from global configuration to line configuration and identifies the specific lines being configured.
- The **autoselect ppp** command configures the Cisco IOS software to allow a PPP session to start up automatically on these selected lines.
- The **autoselect during-login** command is used to display the username and password prompt without pressing the Return key. After the user logs in, the autoselect function (in this case, PPP) begins.
- The **login authentication admins** command applies the “admins” method list for login authentication.
- The **modem dialin** command configures modems attached to the selected lines to only accept incoming calls.

AAA Double Authentication Secured by Absolute Timeout

The AAA Double Authentication Secured by Absolute Timeout feature allows you to secure the double authentication mechanism by protecting it with a per-user session timeout. This feature optimizes the connection to the network by service providers to only connections that are authorized, and it increases the security of the overall access to the network by ensuring that no unwanted sessions are connected.

Feature History for AAA Double Authentication Secured by Absolute Timeout

Release	Modification
12.3(7)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for AAA Double Authentication Secured by Absolute Timeout, page 74
- Restrictions for AAA Double Authentication Secured by Absolute Timeout, page 74
- Information About AAA Double Authentication Secured by Absolute Timeout, page 74
- How to Apply AAA Double Authentication Secured by Absolute Timeout, page 75
- AAA Double Authentication Secured by Absolute Timeout: Examples, page 77
- Additional References, page 81
- Command Reference, page 82

Prerequisites for AAA Double Authentication Secured by Absolute Timeout

- You need access to a Cisco RADIUS or TACACS+ server and should be familiar with configuring RADIUS or TACACS+.
- You should be familiar with configuring AAA.
- You should be familiar with enabling AAA automated double authentication.

Restrictions for AAA Double Authentication Secured by Absolute Timeout

- The AAA Double Authentication Secured by Absolute Timeout feature, like the existing double authentication feature, is for PPP connections only. Automated double authentication cannot be used with other protocols, such as X.25 or Serial Line Internet Protocol (SLIP).
- There may be a minimal impact on performance if a TACACS+ server is used. However, there is no performance impact if a RADIUS server is used.

Information About AAA Double Authentication Secured by Absolute Timeout

To configure the AAA Double Authentication Secured by Absolute Timeout feature, you should understand the following concept:

- Securing a AAA Double Authentication, page 74
- Verifying AAA Double Authentication Secured by Absolute Timeout, page 75

Securing a AAA Double Authentication

With the current AAA double authentication mechanism, a user must pass the first authentication using a host username and password. The second authentication, after Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP), uses a login username and password. In the first authentication, a PPP session timeout will be applied to the virtual access interface if it is configured locally or remotely. The AAA Double Authentication Secured by Absolute Timeout feature allows you to secure the double authentication mechanism by protecting it with a per-user session timeout. The per-user timeout, which can be customized, supersedes the generic absolute timeout value. This method works on the same principle as per-user access control lists (ACLs) in double authentication.

How to Apply AAA Double Authentication Secured by Absolute Timeout

To apply the absolute timeout, you need to configure "Session-Timeout" in the login user profile as a link control protocol (LCP) per-user attribute. There is no new or modified command-line interface (CLI) needed for this feature, but before you use the **access-profile** command when enabling AAA double authentication, you must first reauthorize LCP per-user attributes (for example, Session-Timeout) and then reauthorize Network Control Protocols (NCPs) to apply other necessary criteria, such as ACLs and routes. See the section "AAA Double Authentication Secured by Absolute Timeout: Examples."

Note Timeout configuration in a TACACS+ user profile is a little different from the configuration in a RADIUS user profile. In a RADIUS profile, only one "Session-Timeout" is configured, along with the autocmd "access-profile." The timeout will be applied to the exec session and to the PPP session. In TACACS+, however, the timeout must be configured under the service types "exec" and "ppp" (LCP) to apply a timeout to the exec session and to the PPP session. If the timeout is configured only under the service type "ppp," the timeout value is not available while doing an exec authorization—and the timeout will not be applied to the exec session.

Verifying AAA Double Authentication Secured by Absolute Timeout

To verify that AAA double authentication has been secured by absolute timeout and to see information about various attributes associated with the authentication, perform the following steps. These **show** and **debug** commands can be used in any order.

SUMMARY STEPS

1. **enable**
2. **show users**
3. **show interface**
4. **debug aaa authentication**
5. **debug aaa authorization**
6. **debug aaa per-user**
7. **debug ppp authentication**
8. **debug radius**

   or

   debug tacacs

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 1	**show users** **enable** **Example:** `Router# show users`	Displays information about the active lines on the router.
Step 1	**show interfaces virtual-access** *number* [**configuration**] **Example:** `Router# show interfaces virtual-access 2 configuration`	Displays status, traffic data, and configuration information about a specified virtual access interface.
Step 2	**debug aaa authentication** **Example:** `Router# debug authentication`	Displays information about AAA TACACS+ authentication.
Step 3	**debug aaa authorization** **Example:** `Router# debug aaa authorization`	Displays information about AAA TACACS+ authorization.
Step 4	**debug aaa per-user** **Example:** `Router# debug aaa per-user`	Displays the attributes that are applied to each user as the user authenticates.
Step 5	**debug ppp authentication** **Example:** `Router# debug ppp authentication`	Displays whether a user is passing authentication.
Step 6	**debug radius** or **debug tacacs** **Example:** `Router# debug radius` or `Router# debug tacacs`	Displays information associated with the RADIUS server. or Displays information associated with the TACACS+ server.

Examples

The following sample output is from the **show users** command:

```
Router# show users

    Line       User        Host(s      Idle       Location
  *  0 con 0    aaapbx2     idle        00:00:00   aaacon2 10
     8 vty 0    broker_def  idle        00:00:08   190.0.1.8

    Interface  User            Mode        Idle       Peer Address
    Vi2        broker_default  VDP         00:00:01   190.0.1.8 <=========
    Se0:22     aaapbx2         Sync PPP    00:00:23
```

The following sample output is from the **show interfaces virtual-access** command:

```
Router# show interfaces vi2 configuration

Virtual-Access2 is a Virtual Profile (sub)interface

Derived configuration: 150 bytes
!
interface Virtual-Access2
  ip unnumbered Serial0:23
  no ip route-cache
  timeout absolute 3 0
! The above line shows that the per-user session timeout has been applied.
  ppp authentication chap
  ppp timeout idle 180000
! The above line shows that the absolute timeout has been applied.
```

AAA Double Authentication Secured by Absolute Timeout: Examples

This section includes the following examples:

- RADIUS User Profile: Example, page 77
- TACACS+ User Profile: Example, page 78

RADIUS User Profile: Example

The following sample output shows that a RADIUS user profile has been applied and that AAA double authentication has been secured by an absolute timeout:

```
aaapbx2 Password = "cisco",
 Service-Type = Framed,
 Framed-Protocol = PPP,
 Session-Timeout = 180,
 Idle-Timeout = 180000,
 cisco-avpair = "ip:inacl#1=permit tcp any any eq telnet"
 cisco-avpair = "ip:inacl#2=permit icmp any any"


broker_default Password = "cisco",
 Service-Type = Administrative,
 cisco-avpair = "shell:autocmd=access-profile",
 Session-Timeout = 360,
```

```
 cisco-avpair = "ip:inacl#1=permit tcp any any"
 cisco-avpair = "ip:inacl#2=permit icmp any any"


broker_merge Password = "cisco",
 Service-Type = Administrative,
 cisco-avpair = "shell:autocmd=access-profile merge",
 Session-Timeout = 360,
 cisco-avpair = "ip:inacl#1=permit tcp any any"
 cisco-avpair = "ip:inacl#2=permit icmp any any"
 cisco-avpair = "ip:route#3=10.4.0.0 255.0.0.0"
 cisco-avpair = "ip:route#4=10.5.0.0 255.0.0.0"
 cisco-avpair = "ip:route#5=10.6.0.0 255.0.0.0"


broker_replace Password = "cisco",
 Service-Type = Administrative,
 cisco-avpair = "shell:autocmd=access-profile replace",
 Session-Timeout = 360,
 cisco-avpair = "ip:inacl#1=permit tcp any any"
 cisco-avpair = "ip:inacl#2=permit icmp any any"
 cisco-avpair = "ip:route#3=10.4.0.0 255.0.0.0"
 cisco-avpair = "ip:route#4=10.5.0.0 255.0.0.0"
 cisco-avpair = "ip:route#5=10.6.0.0 255.0.0.0"
```

TACACS+ User Profile: Example

The following sample output shows that a TACACS+ user profile has been applied and that AAA double authentication has been secured by an absolute timeout.

Remote Host

The following allows the remote host to be authenticated by the local host during first-stage authentication and provides the remote host authorization profile.

```
user = aaapbx2
 chap = cleartext Cisco
 pap = cleartext cisco
 login = cleartext cisco

 service = ppp protocol = lcp
  idletime = 3000
  timeout = 3

 service = ppp protocol = ip
  inacl#1="permit tcp any any eq telnet"

 service = ppp protocol = ipx
```

“access_profile” Default Test User “only acls”

Using the **access-profile** command without any arguments causes the removal of any access lists that are found in the old configuration (both per-user and per-interface) and makes sure that the new profile contains only access-list definitions.

```
user = broker_default
 login = cleartext Cisco
 chap = cleartext "cisco"

 service = exec

  autocmd = "access-profile"
```

```
! This is the autocommand that executes when broker_default logs in.
  timeout = 6

 service = ppp protocol = lcp
  timeout = 6

 service = ppp protocol = ip
! Put access lists, static routes, and other requirements that are
! needed here. Read the software specifications for details. If you leave
! this blank, the user will have no access lists (not even the ones that were
! installed prior to the creation of this user profile)!

  inacl#1="permit tcp any any"
  inacl#2="permit icmp host 10.0.0.0 any"

 service = ppp protocol = ipx
! Put access lists, static routes, and other requirements that are
! needed here. Read the software specifications for details. If you leave
! this blank, the user will have no access lists (not even the ones that were
! installed prior to the creation of this user profile)!
```

"access-profile merge" Test User

With the "merge" option, all old access lists are removed (as before), but then almost any AV pair is allowed to be uploaded and installed. This merge will allow for the uploading of any custom static routes, Service Advertisement Protocol (SAP) filters, and other requirements that the user may need in his or her profile. This merge needs to be used with care because it leaves everything open in terms of conflicting configurations.

```
user = broker_merge
 login = cleartext Cisco
 chap = cleartext "cisco"

 service = exec

  autocmd = "access-profile merge"
! This is the autocommand that executes when broker_merge logs in.
  timeout = 6

 service = ppp protocol = lcp
 timeout = 6

 service = ppp protocol = ip
! Put access lists, static routes, and other requirements that are
! needed here. Read the software specifications for details. If you leave
! this blank, the user will have no access lists (not even the ones that were
! installed prior to the creation of this user profile)!

  route#1="10.4.0.0 255.0.0.0"
  route#2="10.5.0.0 255.0.0.0"
  route#3="10.6.0.0 255.0.0.0"
  inacl#5="permit tcp any any"
  inacl#6="permit icmp host 10.60.0.0 any"

 service = ppp protocol = ipx
! Put access lists, static routes, and other requirements that are
! needed here. Read the software specifications for details. If you leave
! this blank, the user will have no access lists (not even the ones that were
! installed prior to the creation of this user profile)!
```

"access_profile replace" Test User

If you use the **access-profile** command with the **replace** keyword, the command works as it does currently; that is, all old configuration is removed and all new configuration is installed.

Note When the **access-profile** command is configured, the new configuration is checked for address pools and address AV pairs. Because addresses cannot be renegotiated at this point, the command will fail to work when it encounters such an address AV pair.

```
user = broker_replace

 login = cleartext Cisco
 chap = cleartext "cisco"

 service = exec

  autocmd = "access-profile replace"
! This is the autocommand that executes when broker_replace logs in.
  timeout = 6

 service = ppp protocol = lcp
  timeout = 6

 service = ppp protocol = ip
! Put access lists, static routes, and other requirements that are
! needed here. Read the software specifications for details. If you leave
! this blank, the user will have no access lists (not even the ones that were
! installed prior to the creation of this user profile)!

  route#1="10.7.0.0 255.0.0.0"
  route#2="10.8.0.0 255.0.0.0"
  route#3="10.9.0.0 255.0.0.0"
  inacl#4="permit tcp any any"

 service = ppp protocol = ipx
! Put access lists, static routes, and other requirements that are
! needed here. Read the software specifications for details. If you leave
! this blank, the user will have no access lists (not even the ones that were
! installed prior to the creation of this user profile)!
```

Note Timeout configuration in a TACACS+ user profile is a little different from the configuration in a RADIUS user profile. In a RADIUS profile, only one "Session-Timeout" is configured, along with the autocmd "access-profile." The timeout will be applied to the exec session and to the PPP session. In TACACS+, however, the timeout must be configured under the service types "exec" and "ppp" (LCP) to apply a timeout to the exec session and to the PPP session. If the timeout is configured only under the service type "ppp," the timeout value is not available while doing an exec authorization—and the timeout will not be applied to the exec session.

Additional References

The following sections provide references related to AAA Double Authentication Secured by Absolute Timeout.

Related Documents

Related Topic	Document Title
Configuring AAA	"Authentication, Authorization, and Accounting" section of the *Cisco IOS Security Configuration Guide*, Release 12.3
Enabling AAA Double Authentication	"Configuring Authentication" chapter of the "Authentication, Authorization, and Accounting" section of the *Cisco IOS Security Configuration Guide*, Release 12.3
Configuring RADIUS	"Configuring RADIUS" chapter of the "Security Server Protocols" section of the *Cisco IOS Security Configuration Guide*, Release 12.3
Configuring TACACS+	"Configuring TACACS+" chapter of the "Security Server Protocols" section of the *Cisco IOS Security Configuration Guide*, Release 12.3
Security Commands	*Cisco IOS Security Command Reference*, Release 12.3 T

Standards

Standards	Title
This feature has no new or modified standards.	—

MIBs

MIBs	MIBs Link
This feature has no new or modified MIBs.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
This feature has no new or modified RFCs.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

This feature uses no new or modified commands. To see the command pages for the commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

Login Password Retry Lockout

The Login Password Retry Lockout feature allows system administrators to lock out a local authentication, authorization, and accounting (AAA) user account after a configured number of unsuccessful attempts by the user to log in.

Feature History for Login Password Retry Lockout

Release	Modification
12.3(14)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for Login Password Retry Lockout, page 83
- Restrictions for Login Password Retry Lockout, page 84
- Information About Login Password Retry Lockout, page 84
- How to Configure Login Password Retry Lockout, page 84
- Configuration Examples for Login Password Retry Lockout, page 87
- Additional References, page 88
- Command Reference, page 89
- Glossary, page 90

Prerequisites for Login Password Retry Lockout

- You must be running a Cisco IOS image that contains the AAA component.

Restrictions for Login Password Retry Lockout

- Authorized users can lock themselves out because there is no distinction between an attacker who is guessing passwords and an authorized user who is entering the password incorrectly multiple times.
- A denial of service (DoS) attack is possible, that is, an authorized user could be locked out by an attacker if the username of the authorized user is known to the attacker.

Information About Login Password Retry Lockout

To configure the Login Password Retry Lockout feature, you should understand the following concept:

- Locking Out a Local AAA User Account, page 84

Locking Out a Local AAA User Account

The Login Password Retry Lockout feature allows system administrators to lock out a local AAA user account after a configured number of unsuccessful attempts by the user to log in using the username that corresponds to the AAA user account. A locked-out user cannot successfully log in again until the user account is unlocked by the administrator.

A system message is generated when a user is either locked by the system or unlocked by the system administrator. The following is an example of such a system message:

```
%AAA-5-USER_LOCKED: User user1 locked out on authentication failure.
```

The system administrator cannot be locked out.

Note The system administrator is a special user who has been configured using the maximum privilege level (root privilege—level 15). A user who has been configured using a lesser privilege level can change the privilege level using the **enable** command. If the user can change to the root privilege (level 15), that user is able to act as a system administrator.

This feature is applicable to any login authentication method, such as ASCII, Challenge Handshake Authentication Protocol (CHAP), and Password Authentication Protocol (PAP).

Note No messages are displayed to users after authentication failures that are due to the locked status (that is, there is no distinction between a normal authentication failure and an authentication failure due to the locked status of the user.

How to Configure Login Password Retry Lockout

This section contains the following procedures:

- Configuring Login Password Retry Lockout, page 85
- Unlocking a Locked-Out User, page 86

- Clearing the Unsuccessful Attempts of a User, page 86
- Monitoring and Maintaining Login Password Retry Lockout, page 87

Configuring Login Password Retry Lockout

To configure Login Password Retry Lockout, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **username** *name* [**privilege** *level*] **password** *encryption-type password*
4. **aaa new-model**
5. **aaa local authentication attempts max-fail** *number-of-unsuccessful-attempts*
6. **aaa authentication login default** *method*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**username** *name* [**privilege** *level*] **password** *encryption-type password* **Example:** `Router (config)# username user1 privilege 15 password 0 cisco`	Establishes a username-based authentication system.
Step 4	**aaa new-model** **Example:** `Router (config)# aaa new-model`	Enables the AAA access control model.

	Command or Action	Purpose
Step 5	**aaa local authentication attempts max-fail** *number-of-unsuccessful-attempts* **Example:** `Router (config)# aaa local authentication attempts max-fail 3`	Specifies the maximum number of unsuccessful attempts before a user is locked out.
Step 6	**aaa authentication login default** method **Example:** `Router (config)# aaa authentication login default local`	Method list for login, specifying to authenticate using the local AAA user database.

Unlocking a Locked-Out User

To unlock the locked-out user, perform the following steps.

Note This task can be performed only by users having root privilege (level 15).

SUMMARY STEPS

1. **enable**
2. **clear aaa local user lockout** {**username** *username* | **all**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear aaa local user lockout** {**username** *username*	**all**} **Example:** `Router# clear aaa local user lockout username user1`	Unlocks a locked-out user.

Clearing the Unsuccessful Attempts of a User

To clear the unsuccessful attempts of a user that have already been logged, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **clear aaa local user fail-attempts** {**username** *username* | **all**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear aaa local user fail-attempts** {**username** *username* | **all**} **Example:** `Router# clear aaa local user fail-attempts username user1`	Clears the unsuccessful attempts of the user. • This command is useful for cases in which the user configuration was changed and the unsuccessful attempts that are already logged must be cleared.

Monitoring and Maintaining Login Password Retry Lockout

To monitor and maintain the Login Password Retry Lockout configuration, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show aaa local user locked**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show aaa local user locked** **Example:** `Router# show aaa local user locked`	Displays a list of the locked-out users.

Configuration Examples for Login Password Retry Lockout

This section provides the following configuration examples:

- Login Password Retry Lockout: Example, page 87
- show aaa local user lockout Command: Example, page 88

Login Password Retry Lockout: Example

The following **show running-config** command output illustrates that the maximum number of failed user attempts has been set for 2:

```
Router # show running-config

Building configuration...

Current configuration : 1214 bytes
!
version 12.3
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname LAC-2
!
boot-start-marker
boot-end-marker
!
!
username sysadmin
username sysad privilege 15 password 0 cisco
username user1 password 0 cisco
aaa new-model
aaa local authentication attempts max-fail 2
!
!
aaa authentication login default local
aaa dnis map enable
aaa session-id common
```

show aaa local user lockout Command: Example

The following output shows that user1 is locked out:

```
Router# show aaa local user lockout

          Local-user          Lock time
          user1               04:28:49 UTC Sat Jun 19 2004
```

Additional References

The following sections provide references related to Login Password Retry Lockout.

Related Documents

Related Topic	Document Title
Cisco IOS security commands	*Cisco IOS Security Command Reference*, Release 12.3T

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **aaa local authentication attempts max-fail**
- **clear aaa local user fail-attempts**
- **clear aaa local user lockout**
- **show aaa local user locked**

Glossary

- **Local AAA method**—Method by which it is possible to configure a local user database on a router and to have AAA provision authentication or authorization of users from this database.
- **Local AAA user**—User who is authenticated using the Local AAA method.

Note Refer to *Internetworking Terms and Acronyms* for terms not included in this glossary.

MSCHAP Version 2

Feature History

Release	Modification
12.2(2)XB5	This feature was introduced.
12.2(13)T	This feature was integrated into Cisco IOS Release 12.2(13)T.

This document describes the MSCHAP Version 2 feature in Cisco IOS Release 12.2(13)T and includes the following sections:

- Feature Overview, page 91
- Supported Platforms, page 92
- Supported Standards, MIBs, and RFCs, page 93
- Prerequisites, page 93
- Configuration Tasks, page 94
- Configuration Examples, page 95
- Command Reference, page 96

Feature Overview

The MSCHAP Version 2 feature in Cisco IOS Release 12.2(13)T introduces the ability of Cisco routers to utilize Microsoft Challenge Handshake Authentication Protocol Version 2 (MSCHAP V2) authentication for PPP connections between a computer using a Microsoft Windows operating system and a network access server (NAS). MSCHAP V2 authentication is an updated version of MSCHAP that is similar to but incompatible with MSCHAP Version 1 (V1). MSCHAP V2 introduces mutual authentication between peers and a change password feature.

Benefits

MSCHAP V2 authentication is the default authentication method used by the Microsoft Windows 2000 operating system. Support of this authentication method on Cisco routers will enable users of the Microsoft Windows 2000 operating system to establish remote PPP sessions without needing to first configure an authentication method on the client.

MSCHAP V2 authentication introduces an additional feature not available with MSCHAP V1 or standard CHAP authentication, the change password feature. This feature allows the client to change the account password if the RADIUS server reports that the password has expired.

Restrictions

The client operating system must support all MSCHAP V2 capabilities.

MSCHAP V2 authentication is not compatible with MSCHAP V1 authentication.

The change password feature is supported only for RADIUS authentication. This feature is not available for local authentication.

In order for the MSCHAP Version 2 feature to correctly interpret the authentication failure attribute sent by the RADIUS server, the **ppp max-bad-auth** command must be configured and the number of authentication retries must be set at two or more.

In order for the MSCHAP Version 2 feature to support the ability to change a password, the authentication failure attribute sent by the RADIUS server must be correctly interpreted as described in this section. In addition, the **radius server vsa send authentication** command must be configured, allowing the RADIUS client to send a vendor-specific attribute to the RADIUS server. The change password feature is supported only for RADIUS authentication.

The Microsoft Windows 2000, Microsoft Windows XP, and Microsoft Windows NT operating systems have a known caveat that prevents the change password function from working. This caveat can be fixed by downloading a patch from Microsoft at the following URL:

http://support.microsoft.com/default.aspx?scid=kb;en-us;Q326770

Related Documents

- The part “PPP Configuration” in the *Cisco IOS Dial Technologies Configuration Guide,* Release 12.2.
- *Cisco IOS Dial Technologies Command Reference*, Release 12.2
- The section “Configuring PPP Authentication Using AAA” in the chapter “Configuring Authentication” in the *Cisco IOS Security Configuration Guide*, Release 12.2
- The chapter “Configuring RADIUS” in the *Cisco IOS Security Configuration Guide*, Release 12.2
- *Cisco IOS Security Command Reference*, Release 12.2
- RFC 1661, *The Point-to-Point Protocol (PPP)*
- RFC 2548, *Microsoft Vendor-specific RADIUS Attributes*

Supported Platforms

- Cisco 800 series
- Cisco 1710
- Cisco 2600 series
- Cisco 3600 series
- Cisco 7200 series

- Cisco 7500 series
- Cisco AS5300
- Cisco AS5400
- Cisco AS5800
- Cisco AS5850

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

No new or modified MIBs are supported by this feature.

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

- RFC 2759, *Microsoft PPP CHAP Extensions, Version 2*

Prerequisites

Before enabling MSCHAP V2 authentication on the NAS, you must perform the following tasks:

- Configure an interface type and enter interface configuration mode by using the **interface** command.
- Configure the interface for PPP encapsulation by using the **encapsulation** command.

For more information on completing these tasks, refer to the section “PPP Configuration” in the *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2.

The RADIUS server must be configured for authentication. Refer to vendor-specific documentation for information on configuring RADIUS authentication on the RADIUS server.

Configuration Tasks

See the following sections for configuration tasks for the MSCHAP Version 2 feature. Each task in the list is identified as either required or optional.

- Configuring MSCHAP V2 Authentication (required)
- Verifying MSCHAP V2 Configuration (optional)

Configuring MSCHAP V2 Authentication

MSCHAP V2 authentication requires prior configuration of an interface type and PPP encapsulation. For more information on configuring PPP, refer to the part "PPP Configuration" in the *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2.

To configure the NAS to accept MSCHAP V2 authentication for local or RADIUS authentication, and to allow proper interpretation of authentication failure attributes and vendor-specific RADIUS attributes for RADIUS authentication, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router# radius-server vsa send authentication`	Configures the NAS to recognize and use vendor-specific attributes.
Step 2	`Router# interface type number`	Configures an interface type and enters interface configuration mode.
Step 3	`Router(config-if)# ppp max-bad-auth number`	Configures a point-to-point interface not to reset itself immediately after an authentication failure but instead to allow a specified number of authentication retries. The *number* argument must be set to a value of at least 2 for authentication failure attributes to be interpreted by the NAS.
Step 4	`Router(config-if)# ppp authentication ms-chap-v2`	Enables MSCHAP V2 authentication on a NAS.

Verifying MSCHAP V2 Configuration

To verify that the MSCHAP Version 2 feature is configured properly, perform the following steps:

Step 1 Enter the **show running-config** command with the **interface** *type number* keyword and argument combination to verify the configuration of MSCHAP V2 as the authentication method for that interface:

```
Router# show running-config interface async 65

interface Asynch65
ip address 10.0.0.2 255.0.0.0
encapsulation ppp
async mode dedicated
no peer default ip address
ppp max-bad-auth 3
ppp authentication ms-chap-v2
```

Step 2 Enter the **debug ppp** command with the **negotiation** keyword to verify successful MSCHAP V2 negotiation:

```
Router# debug ppp negotiation

*Jan 15 13:24:43.999:Se0/0 PPP:Using configured call direction
*Jan 15 13:24:43.999:Se0/0 PPP:Treating connection as a callin
*Jan 15 13:24:43.999:Se0/0 PPP:Phase is ESTABLISHING, Passive Open
*Jan 15 13:24:43.999:Se0/0 LCP:State is Listen
*Jan 15 13:24:44.023:Se0/0 LCP:I CONFREQ [Listen] id 1 len 14
*Jan 15 13:24:44.023:Se0/0 LCP:   MRU 1492 (0x010405D4)
*Jan 15 13:24:44.023:Se0/0 LCP:   MagicNumber 0x308783B9 (0x0506308783B9)
*Jan 15 13:24:44.023:Se0/0 LCP:O CONFREQ [Listen] id 1 len 19
*Jan 15 13:24:44.023:Se0/0 LCP:   MRU 1492 (0x010405D4)
*Jan 15 13:24:44.023:Se0/0 LCP:   AuthProto MS-CHAP-V2 (0x0305C22381)
*Jan 15 13:24:44.023:Se0/0 LCP:   MagicNumber 0x308A180D (0x0506308A180D)
*Jan 15 13:24:44.027:Se0/0 LCP:O CONFACK [Listen] id 1 len 14
*Jan 15 13:24:44.027:Se0/0 LCP:   MRU 1492 (0x010405D4)
*Jan 15 13:24:44.027:Se0/0 LCP:   MagicNumber 0x308783B9 (0x0506308783B9)
*Jan 15 13:24:44.027:Se0/0 LCP:I CONFACK [ACKsent] id 1 len 19
*Jan 15 13:24:44.027:Se0/0 LCP:   MRU 1492 (0x010405D4)
*Jan 15 13:24:44.027:Se0/0 LCP:   AuthProto MS-CHAP-V2 (0x0305C22381)
*Jan 15 13:24:44.027:Se0/0 LCP:   MagicNumber 0x308A180D (0x0506308A180D)
*Jan 15 13:24:44.027:Se0/0 LCP:State is Open
*Jan 15 13:24:44.027:Se0/0 PPP:Phase is AUTHENTICATING, by this end
*Jan 15 13:24:44.027:Se0/0 MS-CHAP-V2:O CHALLENGE id 1 len 24 from "lac"
*Jan 15 13:24:44.031:Se0/0 MS-CHAP-V2:I RESPONSE id 1 len 58 from "haag"
*Jan 15 13:24:44.031:Se0/0 PPP:Phase is FORWARDING, Attempting Forward
*Jan 15 13:24:44.031:Se0/0 PPP:Phase is AUTHENTICATING, Unauthenticated User
*Jan 15 13:24:44.039:Se0/0 PPP:Phase is FORWARDING, Attempting Forward
*Jan 15 13:24:44.043:Se0/0 PPP:Phase is AUTHENTICATING, Authenticated User
*Jan 15 13:24:44.043:Se0/0 MS-CHAP-V2:O SUCCESS id 1 len 46 msg is
"S=4EE927A06B0D624448F27B4BDDA51B5620396EC3"
*Jan 15 13:24:44.043:Se0/0 PPP:Phase is UP
```

Step 3 Enter the **debug ppp** command with the **authentication** keyword to verify successful MSCHAP V2 authentication:

```
Router# debug ppp authentication

*Jan 15 13:26:28.659:Se0/0 PPP:Authorization required
*Jan 15 13:26:28.659:Se0/0 PPP:Using configured call direction
*Jan 15 13:26:28.659:Se0/0 PPP:Treating connection as a callin
*Jan 15 13:26:28.687:Se0/0 MS-CHAP-V2:O CHALLENGE id 1 len 24 from "lac"
*Jan 15 13:26:28.691:Se0/0 MS-CHAP-V2:I RESPONSE id 1 len 58 from "haag"
*Jan 15 13:26:28.691:Se0/0 PPP:Sent MSCHAP-V2 LOGIN Request to AAA
*Jan 15 13:26:28.695:Se0/0 PPP:Received LOGIN Response from AAA = PASS
*Jan 15 13:26:28.703:Se0/0 MS-CHAP-V2:O SUCCESS id 1 len 46 msg is "S=87F5A4BE
```

Configuration Examples

This section provides the following configuration examples:

- Local Authentication Example
- RADIUS Authentication Example

Local Authentication Example

The following example configures PPP on an asynchronous interface and enables MSCHAP V2 authentication locally:

```
interface Async65
  ip address 10.0.0.2 255.0.0.0
  encapsulation ppp
  async mode dedicated
  no peer default ip address
  ppp max-bad-auth 3
  ppp authentication ms-chap-v2
  username client password secret
```

RADIUS Authentication Example

The following example configures PPP on an asynchronous interface and enables MSCHAP V2 authentication via RADIUS:

```
interface Async65
  ip address 10.0.0.2 255.0.0.0
  encapsulation ppp
  async mode dedicated
  no peer default ip address
  ppp max-bad-auth 3
  ppp authentication ms-chap-v2
  exit
aaa authentication ppp default group radius
 radius-server host 10.0.0.2 255.0.0.0
 radius-server key secret
 radius-server vsa send authentication
```

Command Reference

The following new command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **ppp authentication ms-chap-v2**

RADIUS EAP Support

Feature History

Release	Modification
12.2(2)XB5	This feature was introduced on the Cisco 2650, Cisco 3640, Cisco 3660, Cisco AS5300, and Cisco AS400 platforms.
12.2(13)T	This feature was integrated into Cisco IOS Release 12.2(13)T.

This feature module describes the RADIUS EAP Support feature in Cisco IOS Release 12.2(13)T. It includes the following sections:

- Feature Overview, page 97
- Supported Platforms, page 99
- Supported Standards, MIBs, and RFCs, page 100
- Prerequisites, page 100
- Configuration Tasks, page 100
- Configuration Examples, page 101
- Command Reference, page 103
- Glossary, page 104

Feature Overview

The RADIUS EAP Support feature allows users to apply to the client authentication methods that may not be supported by the network access server; this is done via the Extensible Authentication Protocol (EAP). Before this feature was introduced, support for various authentication methods for PPP connections required custom vendor-specific work and changes to the client and NAS.

EAP is an authentication protocol for PPP that supports multiple authentication mechanisms that are negotiated during the authentication phase (instead of the link control protocol [LCP] phase). EAP allows a third-party authentication server to interact with a PPP implementation through a generic interface.

How EAP Works

By default, EAP runs in proxy mode. This means that EAP allows the entire authentication process to be negotiated by the NAS to a back-end server that may reside on or be accessed via a RADIUS server. After EAP is negotiated between the client and the NAS during LCP exchange, all further authentication messages are transparently transmitted between the client and the back-end server. The NAS is no longer directly involved in the authentication process; that is, the NAS works as a proxy, sending EAP messages between the remote peers.

Note EAP can also run in a local mode; the session is authenticated using the Message Digest 5 (MD5) algorithm and obeys the same authentication rules as Challenge Handshake Authentication Protocol (CHAP). To disable proxy mode and authenticate locally, you must use the **ppp eap local** command.

Newly Supported Attributes

The RADIUS EAP Support feature introduces support for the following RADIUS attributes:

Number	IETF Attribute	Description
79	EAP-Message	Encapsulates one fragment of an EAP message, which includes the PPP type, request-id, length, and EAP-type fields.
80	Message Authenticator	Ensures source integrity of the message; all messages that are received with invalid checksums are silently discarded by either end. This attribute contains an HMAC-MD5 checksum of the entire RADIUS request or response message and uses the RADIUS server secret as the key.

Benefits

The RADIUS EAP Support feature makes it possible to apply to the client various authentication methods within PPP (including proprietary authentication) that are not supported by the NAS. Thus, customers can use standard support mechanisms for authentication schemes, such as token cards and public key, to strengthen end-user and device authenticated access to their networks.

Restrictions

When EAP is running in proxy mode, there may be a significant increase in the authentication time because every packet from the peer must be sent to the RADIUS server and every EAP packet from the RADIUS server must be sent back to the client. Although this extra processing will cause delays, you can increase the default authentication timeout value by using the **ppp timeout authentication** command.

Related Documents

- The section "Configuring PPP Authentication Using AAA" in the chapter "Configuring Authentication" in *Cisco IOS Security Configuration Guide*, Release 12.2
- The chapter "Configuring RADIUS" in *Cisco IOS Security Configuration Guide*, Release 12.2
- The chapter "PPP Configuration" in *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2
- *Cisco IOS Dial Technologies Command Reference*, Release 12.2
- *Cisco IOS Security Command Reference*, Release 12.2

Supported Platforms

- Supported platforms in Cisco IOS Release 12.2(2)XB5 include Cisco AS5300, Cisco AS5400, Cisco 2650, Cisco 3640, and Cisco 3660.
- For platforms supported in Cisco IOS Release 12.2(13)T, consult Cisco Feature Navigator.

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that are supported on specific platforms. To obtain updated information about platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. In the release section, you can compare releases side by side to display both the features unique to each software release and the features that releases in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL: http://www.cisco.com/register

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

- RFC 2284, *PPP Extensible Authentication Protocol (EAP)*
- RFC 1938, *A One-Time Password System*
- RFC 2869, *RADIUS Extensions*

Prerequisites

Before enabling EAP RADIUS on the client, you must perform the following tasks:

- Configure an interface type and enter interface configuration mode by using the **interface** command.
- Configure the interface for PPP encapsulation by using the **encapsulation** command.

For more information on completing these tasks, refer to the chapter "Configuring Media-Independent PPP and Multilink PPP" in the *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2.

Configuration Tasks

See the following sections for configuration tasks for the RADIUS EAP Support feature. Each task in the list is identified as either required or optional.

- Configuring EAP (required)
- Verifying EAP (optional)

Configuring EAP

To configure EAP on an interface configured for PPP encapsulation, use the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp authentication eap`	Enables EAP as the authentication protocol.
`Router(config-if)# ppp eap identity string`	(Optional) Specifies the EAP identity when requested by the peer.
`Router(config-if)# ppp eap password [number] string`	(Optional) Sets the EAP password for peer authentication. **Note** This command should only be configured on the client.
`Router(config-if)# ppp eap local`	(Optional) Authenticates locally instead of using a RADIUS back-end server, which is the default. **Note** This command should only be configured on the NAS.
`Router(config-if)# ppp eap wait`	(Optional) Waits for the caller to authenticate itself first. By default, the client always authenticates itself before the caller does. **Note** This command should only be configured on the NAS.
`Router(config-if)# ppp eap refuse [callin]`	(Optional) Refuses to authenticate using EAP. If the **callin** keyword is enabled, only incoming calls will not be authenticated. **Note** This command should only be configured on the NAS.

Verifying EAP

To verify EAP configurations on your client or NAS, use at least one of the following commands in privileged EXEC configuration mode:

Command	Purpose
`Router# show users`	Displays information about the active lines on the router.
`Router# show interfaces`	Displays statistics for all interfaces configured on the router or access server.
`Router# show running-config`	Ensures that your configurations appear as part of the running configuration.

Configuration Examples

This section provides the following configuration examples:

- EAP Local Configuration on Client Example
- EAP Proxy Configuration for NAS Example

EAP Local Configuration on Client Example

The following example is a sample configuration for a client configured for EAP:

```
interface Ethernet0/0
 ip address 1.1.1.202 255.255.255.0
 no ip mroute-cache
 half-duplex
!
interface BRI0/0
 ip address 192.168.101.100 255.255.255.0
 encapsulation ppp
 no ip mroute-cache
 dialer map ip 192.168.101.101 56167
 dialer-group 1
 isdn switch-type basic-5ess
 ppp eap identity user
 ppp eap password 7 141B1309
!
!
ip default-gateway 1.1.1.1
ip classless
ip route 192.168.101.101 255.255.255.255 BRI0/0
no ip http server
!
dialer-list 1 protocol ip permit
```

EAP Proxy Configuration for NAS Example

The following example is a sample configuration for a NAS configured to use EAP proxy:

```
aaa authentication login default group radius
aaa authentication login NOAUTH none
aaa authentication ppp default if-needed group radius
aaa session-id common
enable secret 5 $1$x5D0$cfTL/D8Be.34PgTbdGdgl/
!
username dtw5 password 0 lab
username user password 0 lab

ip subnet-zero
no ip domain-lookup
ip host lab24-boot 172.19.192.254
ip host lb 172.19.192.254
!
isdn switch-type primary-5ess
!
controller T1 3
 framing esf
 linecode b8zs
 pri-group timeslots 1-24
!
interface Ethernet0
 ip address 1.1.1.108 255.255.255.0
 no ip route-cache
 no ip mroute-cache
!
interface Serial3:23
 ip address 192.168.101.101 255.255.255.0
 encapsulation ppp
 dialer map ip 192.168.101.100 60213
```

```
 dialer-group 1
 isdn switch-type primary-5ess
 isdn T321 0
 ppp authentication eap
 ppp eap password 7 011F0706
!
!
ip default-gateway 10.0.190.1
ip classless
ip route 192.168.101.0 255.255.255.0 Serial3:23
no ip http server
!
dialer-list 1 protocol ip permit
!
radius-server host 1.1.1.201 auth-port 1645 acct-port 1646 key lab
radius-server retransmit 3
call rsvp-sync
!
mgcp profile default
!
!
line con 0
 exec-timeout 0 0
 logging synchronous
 login authentication NOAUTH
line 1 48
line aux 0
line vty 0 4
 password lab
```

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

New Command

- **ppp authentication**

Modified Commands

- **ppp eap identity**
- **ppp eap local**
- **ppp eap password**
- **ppp eap refuse**
- **ppp eap wait**

Glossary

attribute—A RADIUS Internet Engineering Task Force (IETF) attribute is one of the original set of 255 standard attributes that are used to communicate authentication, authorization, and accounting (AAA) information between a client and a server. Because IETF attributes are standard, the attribute data is predefined and well known; thus all clients and servers that exchange AAA information via IETF attributes must agree on attribute data such as the exact meaning of the attributes and the general bounds of the values for each attribute.

CHAP—Challenge Handshake Authentication Protocol. Security feature that is supported on lines using PPP encapsulation and prevents unauthorized access. CHAP does not itself prevent unauthorized access; it merely identifies the remote end. The router or access server then determines whether that user is allowed access.

EAP—Extensible Authentication Protocol. A PPP authentication protocol that supports multiple authentication mechanisms that are negotiated during the authentication phase (instead of the Link Control Protocol [LCP] phase). EAP allows a third-party authentication server to interact with the PPP implementation through a generic interface.

LCP—link control protocol. Protocol that establishes, configures, and tests data-link connections for use by PPP.

MD5 (HMAC variant)—Message Digest 5. A hash algorithm used to authenticate packet data. HMAC is a key hashing for message authentication.

NAS—network access server. A device providing local network access to users across a remote access network such as the public switched telephone network (PSTN).

PAP—Password Authentication Protocol. Authentication protocol that allows PPP peers to authenticate one another. The remote router attempting to connect to the local router is required to send an authentication request. Unlike CHAP, PAP passes the password and host name or username in the clear (unencrypted). PAP does not itself prevent unauthorized access; it merely identifies the remote end. The router or access server then determines if that user is allowed access. PAP is supported only on PPP lines.

PPP—Point-to-Point Protocol. A protocol that encapsulates network layer protocol information over point-to-point links. PPP is defined in RFC 1661.

RADIUS—Remote Authentication Dial-In User Service. Database for authenticating modem and ISDN connections and for tracking connection time.

RADIUS Packet of Disconnect

Feature History

Release	Modification
12.1(2)XH	This command was introduced.
12.1(3)T	This command was integrated into Cisco IOS Release 12.1(3)T.
12.2(2)XB	Support for the voice applications as well as support for the Cisco AS5350, Cisco AS5400, and Cisco 3600 series routers was added.
12.2(2)XB1	Support for the Cisco AS5800 was added.
12.2(11)T	This feature was integrated into Cisco IOS Release 12.2(11)T and support for the Cisco AS5850 was added.

This document describes the RADIUS Packet of Disconnect feature in Cisco IOS Release 12.2(11)T. It includes the following sections.

Feature Overview, page 105

Supported Platforms, page 107

Supported Standards, MIBs, and RFCs, page 108

Prerequisites, page 108

Configuration Tasks, page 108

Configuration Examples, page 111

Command Reference, page 111

Glossary, page 112

Feature Overview

This feature consists of a method for terminating a call that has already been connected. This “Packet of Disconnect” (POD) is a RADIUS access_request packet and is intended to be used in situations where the authenticating agent server wants to disconnect the user after the session has been accepted by the RADIUS access_accept packet. This may be needed in at least two situations:

- Detection of fraudulent use, which cannot be performed before accepting the call. A price structure so complex that the maximum session duration cannot be estimated before accepting the call. This may be the case when certain types of discounts are applied or when multiple users use the same subscription simultaneously.

- To prevent unauthorized servers from disconnecting users, the authorizing agent that issues the POD packet must include three parameters in its packet of disconnect request. For a call to be disconnected, all parameters must match their expected values at the gateway. If the parameters do not match, the gateway discards the packet of disconnect packet and sends a NACK (negative acknowledgement message) to the agent.

The parameters are the following:

- An h323-conf-id vendor-specific attribute (VSA) with the same content as received from the gateway for this call.
- An h323-call-origin VSA with the same content as received from the gateway for the leg of interest.
- A 16-byte MD5 hash value that is carried in the *authentication* field of the POD request.

Benefits

- Ability to terminate an in-progress voice call

Restrictions

Proper matching identification information must be communicated by the:

- billing server and gateway configuration
- the gateway's original accounting start request
- the server's POD request

Related Features and Technologies

- AAA, documented in the *Cisco IOS Security Configuration Guide,* Release 12.2

Related Documents

- *Cisco IOS Security Configuration Guide*, Release 12.2
- *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.2
- *Cisco IOS Voice, Video, and Fax Configuration Guide*, Release 12.2
- *Cisco IOS Voice, Video, and Fax Command Reference*, Release 12.2

Supported Platforms

- Cisco 3600 series
- Cisco AS5300
- Cisco AS5350
- Cisco AS5400
- Cisco AS5800
- Cisco AS5850

Table 10 Release and Platform Support for this Feature

Platform	First Limited Cisco IOS Lifetime Release	First Cisco IOS T Release
Cisco 3600 Series	12.2(2)XB	12.2(11)T
Cisco 5300	12.1(2)XH	12.1(3)T
Cisco 5350	12.2(2)XB	12.2(11)T
Cisco 5400	12.2(2)XB	12.2(11)T
Cisco 5800	12.1(2)XH	12.1(3)T
Cisco 5850	X	12.2(11)T

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB web site on Cisco.com at http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

- RFC 2865, *Remote Authentication Dial-in User Service*

Prerequisites

- Configure AAA as described in *Cisco IOS Security Configuration Guide*, Cisco IOS Release 12.2.
- Use Cisco IOS Release 12.2(11)T or later.

Configuration Tasks

See the following sections for configuration tasks for this Packet of Disconnect feature. Each task in the list is identified as either required or optional.

- Configuring AAA POD Server (required)
- Verifying AAA POD Server (optional)

Configuring AAA POD Server

To configure POD, perform the following tasks in global configuration mode:

	Command	Purpose
Step 5	`Router(config)# aaa pod server [port port-number] [auth-type {any| all| session-key}] server-key [encryption-type] string`	Enables inbound user sessions to be disconnected when specific session attributes are presented. **port** *port-number*—(Optional) The network access server User Datagram Protocol (UDP) port to use for POD requests. Default value is 1700. **auth-type—**(Optional) The type of authorization required for disconnecting sessions. **any—**Session that matches all of the attributes sent in the POD packet is disconnected. The POD packet may contain one or more of four key attributes (user-name, framed-IP-address, session-ID, and session-key). **all—**Only a session that matches all four key attributes is disconnected. **All** is the default. **session-key—**Session with a matching session-key attribute is disconnected. All other attributes are ignored. **server-key—**Configures the shared-secret text string. *encryption-type*—(Optional) Single-digit number that defines whether the text immediately following is encrypted, and, if so, what type of encryption is used. Defined encryption types are 0, which means that the text immediately following is not encrypted, and 7, which means that the text is encrypted using an encryption algorithm defined by Cisco. *string*—The shared-secret text string that is shared between the network access server and the client workstation. This shared-secret string must be the same on both systems.

Verifying AAA POD Server

To verify that the gateway is configured correctly to performa s an AA POD server, enter the **show running-configuration** command in privileged EXEC mode to display the command settings for the router.

```
Router# show running-configuration
!
aaa authentication login h323 group radius
aaa authorization exec h323 group radius
aaa accounting update newinfo
aaa accounting connection h323 start-stop group radius
aaa pod server server-key cisco
aaa session-id common
!
```

Troubleshooting Tips

- Ensure that the POD port is configured correctly in both the gateway(using **aaa pod server** command) and the radius server. Both should be the same.
- Ensure that the shared-secret key configured in the gateway (using **aaa pod server** command) and in the AAA server are the same.
- Turn on **debug aaa pod** command to see what's going on. This will let you know if the gateway receives the POD packet from the server and if so, it will display any errors encountered.

The following example shows output from a successful POD request, when using the **show debug** command.

```
Router# debug aaa pod
AAA POD packet processing debugging is on
Router# show debug
General OS:
  AAA POD packet processing debugging is on
Router#
Apr 25 17:15:59.318:POD:172.19.139.206 request queued
Apr 25 17:15:59.318:voice_pod_request:
Apr 25 17:15:59.318:voip_populate_pod_attr_list:
Apr 25 17:15:59.318:voip_pod_get_guid:
Apr 25 17:15:59.318:voip_pod_get_vsa_attr_val:
Apr 25 17:15:59.318:voip_pod_get_vsa_attr_val:attr_len=50
Apr 25 17:15:59.318:voip_pod_get_vsa_attr_val:attr=h323-conf-id
Apr 25 17:15:59.318:voip_pod_get_vsa_attr_val:attr_len=50 value_len=35
Apr 25 17:15:59.318:voip_pod_get_guid:conf-id=FFA7785F F7F607BB
00000000 993FB1F4 n_bytes=35
Apr 25 17:15:59.318:voip_pod_get_guid:GUID = FFA7785F F7F607BB 00000000

993FB1F4
Apr 25 17:15:59.318:voip_populate_pod_attr_list:
Apr 25 17:15:59.318:voip_pod_get_vsa_attr_val:
Apr 25 17:15:59.318:voip_pod_get_vsa_attr_val:attr_len=23
Apr 25 17:15:59.318:voip_pod_get_vsa_attr_val:attr=h323-originate
Apr 25 17:15:59.318:voip_pod_get_vsa_attr_val:attr_len=23 value_len=6
Apr 25 17:15:59.318:voip_get_call_direction:
Apr 25 17:15:59.318:voip_get_call_direction:returning answer
Apr 25 17:15:59.318:voip_eval_pod_attr:
Apr 25 17:15:59.318:cc_api_trigger_disconnect:
Apr 25 17:15:59.322:POD:Sending ACK to 172.19.139.206/1700
Apr 25 17:15:59.322:voip_pod_clean:
```

Configuration Examples

This section provides a configuration example for a gateway performing as an AAA POD server:

- AAA POD Server Example

AAA POD Server Example

```
Router(config)# aaa pod server server-key xyz123
```

Command Reference

The following modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **aaa pod server**
- **debug aaa pod**

Glossary

AAA—authentication, authorization, and accounting.

NACK—negative acknowledgement message.

POD—packet of disconnect. An access_reject packet sent from a RADIUS server to the gateway in order to disconnect a call which has been connected already. After validation of the packet, the gateway disconnects the user. The packet contains the information to disconnect the call.

POD server—a Cisco gateway configured to accept and process POD requests from a RADIUS authentication/authorization agent.

RADIUS—Remote Authentication Dial-In User Service. An authentication and accounting system used by many Internet service providers.

UDP—User Datagram Protocol. Connectionless transport layer protocol in the TCP/IP protocol stack. UDP is a simple protocol that exchanges datagrams without acknowledgments or guaranteed delivery, requiring that error processing and retransmission be handled by other protocols. UDP is defined in RFC 768.

VoIP—voice over IP. The ability to carry normal telephony-style voice over an IP-based Internet with POTS-like functionality, reliability, and voice quality. VoIP is a blanket term that generally refers to the Cisco standards-based (for example, H.323) approach to IP voice traffic.

VSA—vendor-specific attribute.

Configuring Authorization

AAA authorization enables you to limit the services available to a user. When AAA authorization is enabled, the network access server uses information retrieved from the user's profile, which is located either in the local user database or on the security server, to configure the user's session. Once this is done, the user will be granted access to a requested service only if the information in the user profile allows it.

For a complete description of the authorization commands used in this chapter, refer to the chapter "Authorization Commands" in the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Finding Additional Feature Support Information" section on page cxvii in the chapter Using Cisco IOS Software for Release 12.4.

In This Chapter

This chapter contains the following sections:

- Named Method Lists for Authorization
- AAA Authorization Methods
- Method Lists and Server Groups
- AAA Authorization Types
- AAA Authorization Prerequisites
- AAA Authorization Configuration Task List
- Authorization Attribute-Value Pairs
- Authorization Configuration Examples

Named Method Lists for Authorization

Method lists for authorization define the ways that authorization will be performed and the sequence in which these methods will be performed. A method list is simply a named list describing the authorization methods to be queried (such as RADIUS or TACACS+), in sequence. Method lists enable you to designate one or more security protocols to be used for authorization, thus ensuring a backup system in case the initial method fails. Cisco IOS software uses the first method listed to authorize users for

specific network services; if that method fails to respond, the Cisco IOS software selects the next method listed in the method list. This process continues until there is successful communication with a listed authorization method, or all methods defined are exhausted.

Note The Cisco IOS software attempts authorization with the next listed method only when there is no response from the previous method. If authorization fails at any point in this cycle—meaning that the security server or local username database responds by denying the user services—the authorization process stops and no other authorization methods are attempted.

Method lists are specific to the authorization type requested:

- Auth-proxy—Applies specific security policies on a per-user basis. For detailed information on the authentication proxy feature, refer to the chapter "Configuring Authentication Proxy" in the "Traffic Filtering and Firewalls" part of this book.
- Commands—Applies to the EXEC mode commands a user issues. Command authorization attempts authorization for all EXEC mode commands, including global configuration commands, associated with a specific privilege level.
- EXEC—Applies to the attributes associated with a user EXEC terminal session.
- Network—Applies to network connections. This can include a PPP, SLIP, or ARAP connection.
- Reverse Access—Applies to reverse Telnet sessions.

When you create a named method list, you are defining a particular list of authorization methods for the indicated authorization type.

Once defined, method lists must be applied to specific lines or interfaces before any of the defined methods will be performed. The only exception is the default method list (which is named "default"). If the **aaa authorization** command for a particular authorization type is issued without a named method list specified, the default method list is automatically applied to all interfaces or lines except those that have a named method list explicitly defined. (A defined method list overrides the default method list.) If no default method list is defined, local authorization takes place by default.

AAA Authorization Methods

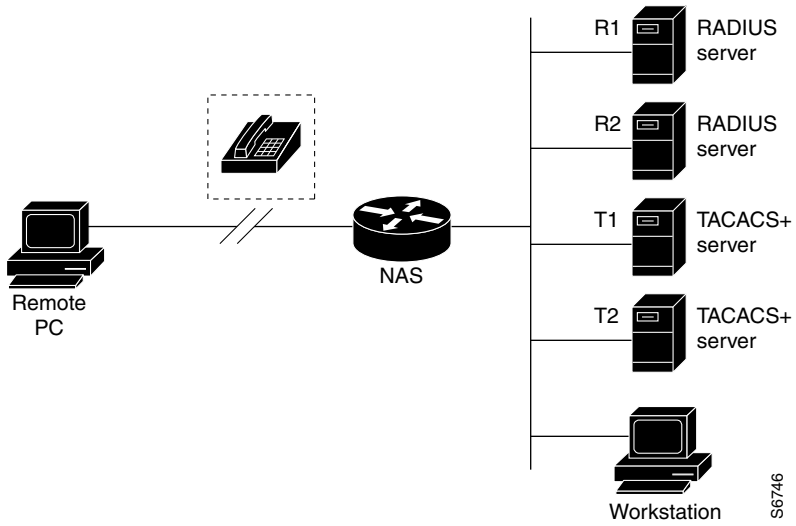
AAA supports five different methods of authorization:

- TACACS+—The network access server exchanges authorization information with the TACACS+ security daemon. TACACS+ authorization defines specific rights for users by associating attribute-value pairs, which are stored in a database on the TACACS+ security server, with the appropriate user.
- If-Authenticated—The user is allowed to access the requested function provided the user has been authenticated successfully.
- None—The network access server does not request authorization information; authorization is not performed over this line/interface.
- Local—The router or access server consults its local database, as defined by the **username** command, for example, to authorize specific rights for users. Only a limited set of functions can be controlled via the local database.
- RADIUS—The network access server requests authorization information from the RADIUS security server. RADIUS authorization defines specific rights for users by associating attributes, which are stored in a database on the RADIUS server, with the appropriate user.

Method Lists and Server Groups

A server group is a way to group existing RADIUS or TACACS+ server hosts for use in method lists. Figure 5 shows a typical AAA network configuration that includes four security servers: R1 and R2 are RADIUS servers, and T1 and T2 are TACACS+ servers. R1 and R2 make up the group of RADIUS servers. T1 and T2 make up the group of TACACS+ servers.

Figure 5 Typical AAA Network Configuration



Using server groups, you can specify a subset of the configured server hosts and use them for a particular service. For example, server groups allow you to define R1 and R2 as separate server groups, and T1 and T2 as separate server groups. This means you can specify either R1 and T1 in the method list or R2 and T2 in the method list, which provides more flexibility in the way that you assign RADIUS and TACACS+ resources.

Server groups also can include multiple host entries for the same server, as long as each entry has a unique identifier. The combination of an IP address and a UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. In other words, this unique identifier enables RADIUS requests to be sent to different UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service—for example, authorization—the second host entry configured acts as fail-over backup to the first one. Using this example, if the first host entry fails to provide accounting services, the network access server will try the second host entry configured on the same device for accounting services. (The RADIUS host entries will be tried in the order they are configured.)

For more information about configuring server groups and about configuring server groups based on DNIS numbers, refer to the chapter “Configuring RADIUS” or the chapter “Configuring TACACS+”

AAA Authorization Types

Cisco IOS software supports five different types of authorization:

- Auth-proxy—Applies specific security policies on a per-user basis. For detailed information on the authentication proxy feature, refer to the "Configuring Authentication Proxy" chapter in the "Traffic Filtering and Firewalls" section of this book.
- Commands—Applies to the EXEC mode commands a user issues. Command authorization attempts authorization for all EXEC mode commands, including global configuration commands, associated with a specific privilege level.
- EXEC—Applies to the attributes associated with a user EXEC terminal session.
- Network—Applies to network connections. This can include a PPP, SLIP, or ARAP connection.
- Reverse Access—Applies to reverse Telnet sessions.
- Configuration—Applies to downloading configurations from the AAA server.
- IP Mobile—Applies to authorization for IP mobile services.

AAA Authorization Prerequisites

Before configuring authorization using named method lists, you must first perform the following tasks:

- Enable AAA on your network access server. For more information about enabling AAA on your Cisco router or access server, refer to the "AAA Overview" chapter.
- Configure AAA authentication. Authorization generally takes place after authentication and relies on authentication to work properly. For more information about AAA authentication, refer to the "Configuring Authentication" chapter.
- Define the characteristics of your RADIUS or TACACS+ security server if you are issuing RADIUS or TACACS+ authorization. For more information about configuring your Cisco network access server to communicate with your RADIUS security server, refer to the chapter "Configuring RADIUS". For more information about configuring your Cisco network access server to communicate with your TACACS+ security server, refer to the chapter "Configuring TACACS+".
- Define the rights associated with specific users by using the **username** command if you are issuing local authorization. For more information about the **username** command, refer to the *Cisco IOS Security Command Reference*.

AAA Authorization Configuration Task List

This section describes the following configuration tasks:

- Configuring AAA Authorization Using Named Method Lists
- Disabling Authorization for Global Configuration Commands
- Configuring Authorization for Reverse Telnet

For authorization configuration examples using the commands in this chapter, refer to the section "Authorization Configuration Examples" at the end of the this chapter.

Configuring AAA Authorization Using Named Method Lists

To configure AAA authorization using named method lists, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa authorization {auth-proxy | network | exec | commands level | reverse-access | configuration | ipmobile} {default | list-name} [method1 [method2...]]`	Creates an authorization method list for a particular authorization type and enable authorization.
Step 2	`Router(config)# line [aux | console | tty | vty] line-number [ending-line-number]` or `Router(config)# interface interface-type interface-number`	Enters the line configuration mode for the lines to which you want to apply the authorization method list. Alternately, enters the interface configuration mode for the interfaces to which you want to apply the authorization method list.
Step 3	`Router(config-line)# authorization {arap | commands level | exec | reverse-access} {default | list-name}` or `Router(config-line)# ppp authorization {default | list-name}`	Applies the authorization list to a line or set of lines. Alternately, applies the authorization list to an interface or set of interfaces.

This section includes the following sections:

- Authorization Types
- Authorization Methods

Authorization Types

Named authorization method lists are specific to the indicated type of authorization.

To create a method list to enable authorization that applies specific security policies on a per-user basis, use the auth-proxy keyword. For detailed information on the authentication proxy feature, refer to the chapter "Configuring Authentication Proxy" in the "Traffic Filtering and Firewalls" part of this book.

To create a method list to enable authorization for all network-related service requests (including SLIP, PPP, PPP NCPs, and ARAP), use the **network** keyword.

To create a method list to enable authorization to determine if a user is allowed to run an EXEC shell, use the **exec** keyword.

To create a method list to enable authorization for specific, individual EXEC commands associated with a specific privilege level, use the **commands** keyword. (This allows you to authorize all commands associated with a specified command level from 0 to 15.)

To create a method list to enable authorization for reverse Telnet functions, use the **reverse-access** keyword.

For information about the types of authorization supported by the Cisco IOS software, refer to the "AAA Authorization Types" section of this chapter.

Authorization Methods

To have the network access server request authorization information via a TACACS+ security server, use the **aaa authorization** command with the **group tacacs+** *method* keyword. For more specific information about configuring authorization using a TACACS+ security server, refer to the chapter "Configuring TACACS+." For an example of how to enable a TACACS+ server to authorize the use of network services, including PPP and ARA, see the section "TACACS+ Authorization Examples" at the end of this chapter.

To allow users to have access to the functions they request as long as they have been authenticated, use the **aaa authorization** command with the **if-authenticated** *method* keyword. If you select this method, all requested functions are automatically granted to authenticated users.

There may be times when you do not want to run authorization from a particular interface or line. To stop authorization activities on designated lines or interfaces, use the **none** *method* keyword. If you select this method, authorization is disabled for all actions.

To select local authorization, which means that the router or access server consults its local user database to determine the functions a user is permitted to use, use the **aaa authorization** command with the **local** *method* keyword. The functions associated with local authorization are defined by using the **username** global configuration command. For a list of permitted functions, refer to the chapter "Configuring Authentication."

To have the network access server request authorization via a RADIUS security server, use the **radius** *method* keyword. For more specific information about configuring authorization using a RADIUS security server, refer to the chapter "Configuring RADIUS."

To have the network access server request authorization via a RADIUS security server, use the **aaa authorization** command with the **group radius** *method* keyword. For more specific information about configuring authorization using a RADIUS security server, refer to the chapter "Configuring RADIUS". For an example of how to enable a RADIUS server to authorize services, see the "RADIUS Authorization Example" section at the end of this chapter.

Note Authorization method lists for SLIP follow whatever is configured for PPP on the relevant interface. If no lists are defined and applied to a particular interface (or no PPP settings are configured), the default setting for authorization applies.

Disabling Authorization for Global Configuration Commands

The **aaa authorization** command with the keyword **commands** attempts authorization for all EXEC mode commands, including global configuration commands, associated with a specific privilege level. Because there are configuration commands that are identical to some EXEC-level commands, there can be some confusion in the authorization process. Using **no aaa authorization config-commands** stops the network access server from attempting configuration command authorization.

To disable AAA authorization for all global configuration commands, use the following command in global configuration mode:

Command	Purpose
`Router(config)# no aaa authorization config-commands`	Disables authorization for all global configuration commands.

Configuring Authorization for Reverse Telnet

Telnet is a standard terminal emulation protocol used for remote terminal connection. Normally, you log in to a network access server (typically through a dialup connection) and then use Telnet to access other network devices from that network access server. There are times, however, when it is necessary to establish a reverse Telnet session. In reverse Telnet sessions, the Telnet connection is established in the opposite direction—from inside a network to a network access server on the network periphery to gain access to modems or other devices connected to that network access server. Reverse Telnet is used to provide users with dialout capability by allowing them to Telnet to modem ports attached to a network access server.

It is important to control access to ports accessible through reverse Telnet. Failure to do so could, for example, allow unauthorized users free access to modems where they can trap and divert incoming calls or make outgoing calls to unauthorized destinations.

Authentication during reverse Telnet is performed through the standard AAA login procedure for Telnet. Typically the user has to provide a username and password to establish either a Telnet or reverse Telnet session. Reverse Telnet authorization provides an additional (optional) level of security by requiring authorization in addition to authentication. When enabled, reverse Telnet authorization can use RADIUS or TACACS+ to authorize whether or not this user is allowed reverse Telnet access to specific asynchronous ports, after the user successfully authenticates through the standard Telnet login procedure.

Reverse Telnet authorization offers the following benefits:

- An additional level of protection by ensuring that users engaged in reverse Telnet activities are indeed authorized to access a specific asynchronous port using reverse Telnet.
- An alternative method (other than access lists) to manage reverse Telnet authorization.

To configure a network access server to request authorization information from a TACACS+ or RADIUS server before allowing a user to establish a reverse Telnet session, use the following command in global configuration mode:

Command	Purpose
`Router(config)# aaa authorization reverse-access` *method1* [*method2 ...*]	Configures the network access server to request authorization information before allowing a user to establish a reverse Telnet session.

This feature enables the network access server to request reverse Telnet authorization information from the security server, whether RADIUS or TACACS+. You must configure the specific reverse Telnet privileges for the user on the security server itself.

Authorization Attribute-Value Pairs

RADIUS and TACACS+ authorization both define specific rights for users by processing attributes, which are stored in a database on the security server. For both RADIUS and TACACS+, attributes are defined on the security server, associated with the user, and sent to the network access server where they are applied to the user's connection.

For a list of supported RADIUS attributes, refer to the appendix "RADIUS Attributes". For a list of supported TACACS+ AV pairs, refer to the appendix "TACACS+ Attribute-Value Pairs."

Authorization Configuration Examples

The following sections provide authorization configuration examples:

- Named Method List Configuration Example
- TACACS+ Authorization Examples
- RADIUS Authorization Example
- Reverse Telnet Authorization Examples

Named Method List Configuration Example

The following example shows how to configure a Cisco AS5300 (enabled for AAA and communication with a RADIUS security server) for AAA services to be provided by the RADIUS server. If the RADIUS server fails to respond, then the local database will be queried for authentication and authorization information, and accounting services will be handled by a TACACS+ server.

```
aaa new-model
aaa authentication login admins local
aaa authentication ppp dialins group radius local
aaa authorization network scoobee group radius local
aaa accounting network charley start-stop group radius

username root password ALongPassword

radius-server host alcatraz
radius-server key myRaDiUSpassWoRd

interface group-async 1
 group-range 1 16
 encapsulation ppp
 ppp authentication chap dialins
 ppp authorization scoobee
 ppp accounting charley

line 1 16
 autoselect ppp
 autoselect during-login
 login authentication admins
 modem dialin
```

The lines in this sample RADIUS AAA configuration are defined as follows:

- The **aaa new-model** command enables AAA network security services.
- The **aaa authentication login admins local** command defines a method list, admins, for login authentication.
- The **aaa authentication ppp dialins group radius local** command defines the authentication method list “dialins,” which specifies that RADIUS authentication then (if the RADIUS server does not respond) local authentication will be used on serial lines using PPP.
- The **aaa authorization network scoobee group radius local** command defines the network authorization method list named scoobee, which specifies that RADIUS authorization will be used on serial lines using PPP. If the RADIUS server fails to respond, then local network authorization will be performed.

- The **aaa accounting network charley start-stop group radius** command defines the network accounting method list named charley, which specifies that RADIUS accounting services (in this case, start and stop records for specific events) will be used on serial lines using PPP.
- The **username** command defines the username and password to be used for the PPP Password Authentication Protocol (PAP) caller identification.
- The **radius-server host** command defines the name of the RADIUS server host.
- The **radius-server key** command defines the shared secret text string between the network access server and the RADIUS server host.
- The **interface group-async** command selects and defines an asynchronous interface group.
- The **group-range** command defines the member asynchronous interfaces in the interface group.
- The **encapsulation ppp** command sets PPP as the encapsulation method used on the specified interfaces.
- The **ppp authentication chap dialins** command selects Challenge Handshake Authentication Protocol (CHAP) as the method of PPP authentication and applies the "dialins" method list to the specified interfaces.
- The **ppp authorization scoobee** command applies the scoobee network authorization method list to the specified interfaces.
- The **ppp accounting charley** command applies the charley network accounting method list to the specified interfaces.
- The **line** command switches the configuration mode from global configuration to line configuration and identifies the specific lines being configured.
- The **autoselect ppp** command configures the Cisco IOS software to allow a PPP session to start up automatically on these selected lines.
- The **autoselect during-login** command is used to display the username and password prompt without pressing the Return key. After the user logs in, the autoselect function (in this case, PPP) begins.
- The **login authentication admins** command applies the admins method list for login authentication.
- The **modem dialin** command configures modems attached to the selected lines to only accept incoming calls.

TACACS+ Authorization Examples

The following examples show how to use a TACACS+ server to authorize the use of network services, including PPP and ARA. If the TACACS+ server is not available or an error occurs during the authorization process, the fallback method (none) is to grant all authorization requests:

```
aaa authorization network default group tacacs+ none
```

The following example shows how to allow network authorization using TACACS+:

```
aaa authorization network default group tacacs+
```

The following example shows how to provide the same authorization, but it also creates address pools called "mci" and "att":

```
aaa authorization network default group tacacs+
ip address-pool local
ip local-pool mci 172.16.0.1 172.16.0.255
ip local-pool att 172.17.0.1 172.17.0.255
```

These address pools can then be selected by the TACACS daemon. A sample configuration of the daemon follows:

```
user = mci_customer1 {
    login = cleartext "some password"
    service = ppp protocol = ip {
        addr-pool=mci
    }
}

user = att_customer1 {
    login = cleartext "some other password"
    service = ppp protocol = ip {
        addr-pool=att
     }
```

RADIUS Authorization Example

The following example shows how to configure the router to authorize using RADIUS:

```
aaa new-model
aaa authorization exec default group radius if-authenticated
aaa authorization network default group radius
radius-server host ip
radius-server key
```

The lines in this sample RADIUS authorization configuration are defined as follows:

- The **aaa authorization exec default group radius if-authenticated** command configures the network access server to contact the RADIUS server to determine if users are permitted to start an EXEC shell when they log in. If an error occurs when the network access server contacts the RADIUS server, the fallback method is to permit the CLI to start, provided the user has been properly authenticated.

  The RADIUS information returned may be used to specify an autocommand or a connection access list be applied to this connection.
- The **aaa authorization network default group radius** command configures network authorization via RADIUS. This can be used to govern address assignment, the application of access lists, and various other per-user quantities.

Note Because no fallback method is specified in this example, authorization will fail if, for any reason, there is no response from the RADIUS server.

Reverse Telnet Authorization Examples

The following examples show how to cause the network access server to request authorization information from a TACACS+ security server before allowing a user to establish a reverse Telnet session:

```
aaa new-model
aaa authentication login default group tacacs+
aaa authorization reverse-access default group tacacs+
!
tacacs-server host 172.31.255.0
tacacs-server timeout 90
tacacs-server key goaway
```

The lines in this sample TACACS+ reverse Telnet authorization configuration are defined as follows:

- The **aaa new-model** command enables AAA.
- The **aaa authentication login default group tacacs+** command specifies TACACS+ as the default method for user authentication during login.
- The **aaa authorization reverse-access default group tacacs+** command specifies TACACS+ as the method for user authorization when trying to establish a reverse Telnet session.
- The **tacacs-server host** command identifies the TACACS+ server.
- The **tacacs-server timeout** command sets the interval of time that the network access server waits for the TACACS+ server to reply.
- The **tacacs-server key** command defines the encryption key used for all TACACS+ communications between the network access server and the TACACS+ daemon.

The following example shows how to configure a generic TACACS+ server to grant a user, pat, reverse Telnet access to port tty2 on the network access server named “maple” and to port tty5 on the network access server named “oak”:

```
user = pat
  login = cleartext lab
  service = raccess {
    port#1 = maple/tty2
    port#2 = oak/tty5
```

Note In this example, “maple” and “oak” are the configured host names of network access servers, not DNS names or alias.

The following example shows how to configure the TACACS+ server (CiscoSecure) to grant a user named pat reverse Telnet access:

```
user = pat
profile_id = 90
profile_cycle = 1
member = Tacacs_Users
service=shell {
default cmd=permit
}
service=raccess {
allow "c2511e0" "tty1" ".*"
refuse ".*" ".*" ".*"
password = clear "goaway"
```

Note CiscoSecure only supports reverse Telnet using the command line interface in versions 2.1(*x*) through version 2.2(1).

An empty “service=raccess {}” clause permits a user to have unconditional access to network access server ports for reverse Telnet. If no “service=raccess” clause exists, the user is denied access to any port for reverse Telnet.

For more information about configuring TACACS+, refer to the chapter “Configuring TACACS+.” For more information about configuring CiscoSecure, refer to the *CiscoSecure Access Control Server User Guide*, version 2.1(2) or greater.

The following example shows how to cause the network access server to request authorization from a RADIUS security server before allowing a user to establish a reverse Telnet session:

```
aaa new-model
```

```
aaa authentication login default group radius
aaa authorization reverse-access default group radius
!
radius-server host 172.31.255.0
radius-server key go away
auth-port 1645 acct-port 1646
```

The lines in this sample RADIUS reverse Telnet authorization configuration are defined as follows:

- The **aaa new-model** command enables AAA.
- The **aaa authentication login default group radius** command specifies RADIUS as the default method for user authentication during login.
- The **aaa authorization reverse-access default group radius** command specifies RADIUS as the method for user authorization when trying to establish a reverse Telnet session.
- The **radius-server host** command identifies the RADIUS server.
- The **radius-server key** command defines the encryption key used for all RADIUS communications between the network access server and the RADIUS daemon.

The following example shows how to send a request to the RADIUS server to grant a user named “pat” reverse Telnet access at port tty2 on the network access server named “maple”:

```
Username = “pat”
Password = “goaway”
User-Service-Type = Shell-User
cisco-avpair = “raccess:port#1=maple/tty2”
```

The syntax "raccess:port=any/any" permits a user to have unconditional access to network access server ports for reverse Telnet. If no "raccess:port={*nasname*}/{*tty number*}" clause exists in the user profile, the user is denied access to reverse Telnet on all ports.

For more information about configuring RADIUS, refer to the chapter “Configuring RADIUS.”

Configuring Accounting

The AAA accounting feature enables you to track the services that users are accessing and the amount of network resources that they are consuming. When AAA accounting is enabled, the network access server reports user activity to the TACACS+ or RADIUS security server (depending on which security method you have implemented) in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the security server. This data can then be analyzed for network management, client billing, and auditing.

For a complete description of the accounting commands used in this chapter, refer to the chapter "Accounting Commands" in the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Finding Additional Feature Support Information" section on page cxvii in the chapter Using Cisco IOS Software for Release 12.4.

In This Chapter

This chapter contains the following sections:

- Named Method Lists for Accounting
- AAA Accounting Types
- AAA Accounting Enhancements
- AAA Accounting Prerequisites
- AAA Accounting Configuration Task List
- Accounting Attribute-Value Pairs
- Accounting Configuration Examples

Named Method Lists for Accounting

Like authentication and authorization method lists, method lists for accounting define the way accounting will be performed and the sequence in which these methods are performed.

Named accounting method lists enable you to designate a particular security protocol to be used on specific lines or interfaces for accounting services. The only exception is the default method list (which, by coincidence, is named "default"). The default method list is automatically applied to all interfaces except those that have a named method list explicitly defined. A defined method list overrides the default method list.

A method list is simply a named list describing the accounting methods to be queried (such as RADIUS or TACACS+), in sequence. Method lists enable you to designate one or more security protocols to be used for accounting, thus ensuring a backup system for accounting in case the initial method fails. Cisco IOS software uses the first method listed to support accounting; if that method fails to respond, the Cisco IOS software selects the next accounting method listed in the method list. This process continues until there is successful communication with a listed accounting method, or all methods defined are exhausted.

Note The Cisco IOS software attempts accounting with the next listed accounting method only when there is no response from the previous method. If accounting fails at any point in this cycle—meaning that the security server responds by denying the user access—the accounting process stops and no other accounting methods are attempted.

Accounting method lists are specific to the type of accounting being requested. AAA supports six different types of accounting:

- Network—Provides information for all PPP, SLIP, or ARAP sessions, including packet and byte counts.
- EXEC—Provides information about user EXEC terminal sessions of the network access server.
- Commands—Provides information about the EXEC mode commands that a user issues. Command accounting generates accounting records for all EXEC mode commands, including global configuration commands, associated with a specific privilege level.
- Connection—Provides information about all outbound connections made from the network access server, such as Telnet, local-area transport (LAT), TN3270, packet assembler/disassembler (PAD), and rlogin.
- System—Provides information about system-level events.
- Resource—Provides "start" and "stop" records for calls that have passed user authentication, and provides "stop" records for calls that fail to authenticate.

Note System accounting does not use named accounting lists; you can only define the default list for system accounting.

Once again, when you create a named method list, you are defining a particular list of accounting methods for the indicated accounting type.

Accounting method lists must be applied to specific lines or interfaces before any of the defined methods will be performed. The only exception is the default method list (which is named "default"). If the **aaa accounting** command for a particular accounting type is issued without a named method list specified, the default method list is automatically applied to all interfaces or lines except those that have a named method list explicitly defined. (A defined method list overrides the default method list.) If no default method list is defined, then no accounting takes place.

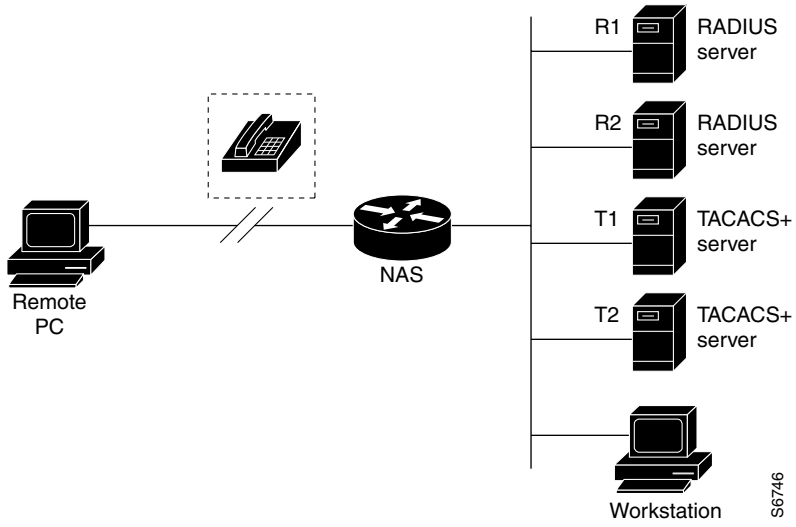
This section includes the following subsections:

- Method Lists and Server Groups
- AAA Accounting Methods

Method Lists and Server Groups

A server group is a way to group existing RADIUS or TACACS+ server hosts for use in method lists. Figure 6 shows a typical AAA network configuration that includes four security servers: R1 and R2 are RADIUS servers, and T1 and T2 are TACACS+ servers. R1 and R2 comprise the group of RADIUS servers. T1 and T2 comprise the group of TACACS+ servers.

Figure 6 Typical AAA Network Configuration



In Cisco IOS software, RADIUS and TACACS+ server configurations are global. Using server groups, you can specify a subset of the configured server hosts and use them for a particular service. For example, server groups allow you to define R1 and R2 as separate server groups (SG1 and SG2), and T1 and T2 as separate server groups (SG3 and SG4). This means you can specify either R1 and T1 (SG1 and SG3) in the method list or R2 and T2 (SG2 and SG4) in the method list, which provides more flexibility in the way that you assign RADIUS and TACACS+ resources.

Server groups also can include multiple host entries for the same server, as long as each entry has a unique identifier. The combination of an IP address and a UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. In other words, this unique identifier enables RADIUS requests to be sent to different UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service—for example, accounting—the second host entry configured acts as failover backup to the first one. Using this example, if the first host entry fails to provide accounting services, the network access server will try the second host entry configured on the same device for accounting services. (The RADIUS host entries will be tried in the order in which they are configured.)

For more information about configuring server groups and about configuring server groups based on DNIS numbers, refer to the chapter “Configuring RADIUS” or the chapter “Configuring TACACS+.”

AAA Accounting Methods

Cisco IOS supports the following two methods for accounting:

- TACACS+—The network access server reports user activity to the TACACS+ security server in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the security server.
- RADIUS—The network access server reports user activity to the RADIUS security server in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the security server.

AAA Accounting Types

AAA supports six different accounting types:

- Network Accounting
- Connection Accounting
- EXEC Accounting
- System Accounting
- Command Accounting
- Resource Accounting

Network Accounting

Network accounting provides information for all PPP, SLIP, or ARAP sessions, including packet and byte counts.

The following example shows the information contained in a RADIUS network accounting record for a PPP user who comes in through an EXEC session:

```
Wed Jun 27 04:44:45 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 5
        User-Name = "fgeorge"
        Client-Port-DNIS = "4327528"
        Caller-ID = "562"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = "0000000D"
        Acct-Delay-Time = 0
        User-Id = "fgeorge"
        NAS-Identifier = "172.16.25.15"

Wed Jun 27 04:45:00 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 5
        User-Name = "fgeorge"
        Client-Port-DNIS = "4327528"
        Caller-ID = "562"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Framed
```

```
        Acct-Session-Id = "0000000E"
        Framed-IP-Address = "10.1.1.2"
        Framed-Protocol = PPP
        Acct-Delay-Time = 0
        User-Id = "fgeorge"
        NAS-Identifier = "172.16.25.15"

Wed Jun 27 04:47:46 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 5
        User-Name = "fgeorge"
        Client-Port-DNIS = "4327528"
        Caller-ID = "562"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Framed
        Acct-Session-Id = "0000000E"
        Framed-IP-Address = "10.1.1.2"
        Framed-Protocol = PPP
        Acct-Input-Octets = 3075
        Acct-Output-Octets = 167
        Acct-Input-Packets = 39
        Acct-Output-Packets = 9
        Acct-Session-Time = 171
        Acct-Delay-Time = 0
        User-Id = "fgeorge"
        NAS-Identifier = "172.16.25.15"

Wed Jun 27 04:48:45 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 5
        User-Name = "fgeorge"
        Client-Port-DNIS = "4327528"
        Caller-ID = "408"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = "0000000D"
        Acct-Delay-Time = 0
        User-Id = "fgeorge"
        NAS-Identifier = "172.16.25.15"
```

The following example shows the information contained in a TACACS+ network accounting record for a PPP user who first started an EXEC session:

```
Wed Jun 27 04:00:35 2001 172.16.25.15   fgeorge   tty4   562/4327528    starttask_id=28
service=shell
Wed Jun 27 04:00:46 2001 172.16.25.15    fgeorge   tty4 562/4327528    starttask_id=30
addr=10.1.1.1   service=ppp
Wed Jun 27 04:00:49 2001 172.16.25.15    fgeorge   tty4    408/4327528     update
task_id=30      addr=10.1.1.1   service=ppp     protocol=ip     addr=10.1.1.1
Wed Jun 27 04:01:31 2001 172.16.25.15   fgeorge   tty4   562/4327528    stoptask_id=30
addr=10.1.1.1   service=ppp     protocol=ip     addr=10.1.1.1   bytes_in=2844
bytes_out=1682  paks_in=36      paks_out=24     elapsed_time=51
Wed Jun 27 04:01:32 2001 172.16.25.15   fgeorge   tty4   562/4327528    stoptask_id=28
service=shell   elapsed_time=57
```

Note The precise format of accounting packets records may vary depending on your particular security server daemon.

The following example shows the information contained in a RADIUS network accounting record for a PPP user who comes in through autoselect:

```
Wed Jun 27 04:30:52 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 3
        User-Name = "fgeorge"
        Client-Port-DNIS = "4327528"
        Caller-ID = "562"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Framed
        Acct-Session-Id = "0000000B"
        Framed-Protocol = PPP
        Acct-Delay-Time = 0
        User-Id = "fgeorge"
        NAS-Identifier = "172.16.25.15"

Wed Jun 27 04:36:49 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 3
        User-Name = "fgeorge"
        Client-Port-DNIS = "4327528"
        Caller-ID = "562"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Framed
        Acct-Session-Id = "0000000B"
        Framed-Protocol = PPP
        Framed-IP-Address = "10.1.1.1"
        Acct-Input-Octets = 8630
        Acct-Output-Octets = 5722
        Acct-Input-Packets = 94
        Acct-Output-Packets = 64
        Acct-Session-Time = 357
        Acct-Delay-Time = 0
        User-Id = "fgeorge"
        NAS-Identifier = "172.16.25.15"
```

The following example shows the information contained in a TACACS+ network accounting record for a PPP user who comes in through autoselect:

```
Wed Jun 27 04:02:19 2001 172.16.25.15   fgeorge   Async5  562/4327528    starttask_id=35
service=ppp
Wed Jun 27 04:02:25 2001 172.16.25.15    fgeorge   Async5  562/4327528     update
task_id=35      service=ppp     protocol=ip     addr=10.1.1.2
Wed Jun 27 04:05:03 2001 172.16.25.15   fgeorge   Async5  562/4327528    stoptask_id=35
service=ppp     protocol=ip     addr=10.1.1.2   bytes_in=3366   bytes_out=2149
paks_in=42      paks_out=28     elapsed_time=164
```

Connection Accounting

Connection accounting provides information about all outbound connections made from the network access server, such as Telnet, local-area transport (LAT), TN3270, packet assembler/disassembler (PAD), and rlogin.

The following example shows the information contained in a RADIUS connection accounting record for an outbound Telnet connection:

```
Wed Jun 27 04:28:00 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 2
        User-Name = "fgeorge"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329477"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Login
        Acct-Session-Id = "00000008"
        Login-Service = Telnet
        Login-IP-Host = "171.68.202.158"
        Acct-Delay-Time = 0
        User-Id = "fgeorge"
        NAS-Identifier = "172.16.25.15"

Wed Jun 27 04:28:39 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 2
        User-Name = "fgeorge"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329477"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Login
        Acct-Session-Id = "00000008"
        Login-Service = Telnet
        Login-IP-Host = "171.68.202.158"
        Acct-Input-Octets = 10774
        Acct-Output-Octets = 112
        Acct-Input-Packets = 91
        Acct-Output-Packets = 99
        Acct-Session-Time = 39
        Acct-Delay-Time = 0
        User-Id = "fgeorge"
        NAS-Identifier = "172.16.25.15"
```

The following example shows the information contained in a TACACS+ connection accounting record for an outbound Telnet connection:

```
Wed Jun 27 03:47:43 2001        172.16.25.15    fgeorge   tty3    5622329430/4327528
start    task_id=10      service=connection      protocol=telnet addr=171.68.202.158
cmd=telnet fgeorge-sun
Wed Jun 27 03:48:38 2001       172.16.25.15   fgeorge   tty3    5622329430/4327528  stop
task_id=10      service=connection      protocol=telnet addr=171.68.202.158 cmd=telnet
fgeorge-sun     bytes_in=4467   bytes_out=96    paks_in=61      paks_out=72
elapsed_time=55
```

The following example shows the information contained in a RADIUS connection accounting record for an outbound rlogin connection:

```
Wed Jun 27 04:29:48 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 2
```

```
        User-Name = "fgeorge"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329477"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Login
        Acct-Session-Id = "0000000A"
        Login-Service = Rlogin
        Login-IP-Host = "171.68.202.158"
        Acct-Delay-Time = 0
        User-Id = "fgeorge"
        NAS-Identifier = "172.16.25.15"

Wed Jun 27 04:30:09 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 2
        User-Name = "fgeorge"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329477"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Login
        Acct-Session-Id = "0000000A"
        Login-Service = Rlogin
        Login-IP-Host = "171.68.202.158"
        Acct-Input-Octets = 18686
        Acct-Output-Octets = 86
        Acct-Input-Packets = 90
        Acct-Output-Packets = 68
        Acct-Session-Time = 22
        Acct-Delay-Time = 0
        User-Id = "fgeorge"
        NAS-Identifier = "172.16.25.15"
```

The following example shows the information contained in a TACACS+ connection accounting record for an outbound rlogin connection:

```
Wed Jun 27 03:48:46 2001        172.16.25.15    fgeorge   tty3    5622329430/4327528
start    task_id=12      service=connection      protocol=rlogin addr=171.68.202.158
cmd=rlogin fgeorge-sun /user fgeorge
Wed Jun 27 03:51:37 2001       172.16.25.15   fgeorge   tty3    5622329430/4327528  stop
task_id=12      service=connection      protocol=rlogin addr=171.68.202.158 cmd=rlogin
fgeorge-sun /user fgeorge bytes_in=659926 bytes_out=138   paks_in=2378    paks_
out=1251        elapsed_time=171
```

The following example shows the information contained in a TACACS+ connection accounting record for an outbound LAT connection:

```
Wed Jun 27 03:53:06 2001        172.16.25.15    fgeorge   tty3    5622329430/4327528
start    task_id=18      service=connection      protocol=lat    addr=VAX        cmd=lat
VAX
Wed Jun 27 03:54:15 2001       172.16.25.15   fgeorge   tty3    5622329430/4327528  stop
task_id=18      service=connection      protocol=lat    addr=VAX        cmd=lat VAX
bytes_in=0      bytes_out=0     paks_in=0      paks_out=0      elapsed_time=6
```

EXEC Accounting

EXEC accounting provides information about user EXEC terminal sessions (user shells) on the network access server, including username, date, start and stop times, the access server IP address, and (for dial-in users) the telephone number the call originated from.

The following example shows the information contained in a RADIUS EXEC accounting record for a dial-in user:

```
Wed Jun 27 04:26:23 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 1
        User-Name = "fgeorge"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329483"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = "00000006"
        Acct-Delay-Time = 0
        User-Id = "fgeorge"
        NAS-Identifier = "172.16.25.15"

Wed Jun 27 04:27:25 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 1
        User-Name = "fgeorge"
        Client-Port-DNIS = "4327528"
        Caller-ID = "5622329483"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = "00000006"
        Acct-Session-Time = 62
        Acct-Delay-Time = 0
        User-Id = "fgeorge"
        NAS-Identifier = "172.16.25.15"
```

The following example shows the information contained in a TACACS+ EXEC accounting record for a dial-in user:

```
Wed Jun 27 03:46:21 2001        172.16.25.15    fgeorge   tty3    5622329430/4327528
start    task_id=2       service=shell
Wed Jun 27 04:08:55 2001       172.16.25.15   fgeorge   tty3    5622329430/4327528  stop
task_id=2       service=shell   elapsed_time=1354
```

The following example shows the information contained in a RADIUS EXEC accounting record for a Telnet user:

```
Wed Jun 27 04:48:32 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 26
        User-Name = "fgeorge"
        Caller-ID = "171.68.202.158"
        Acct-Status-Type = Start
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = "00000010"
        Acct-Delay-Time = 0
        User-Id = "fgeorge"
        NAS-Identifier = "172.16.25.15"
```

```
Wed Jun 27 04:48:46 2001
        NAS-IP-Address = "172.16.25.15"
        NAS-Port = 26
        User-Name = "fgeorge"
        Caller-ID = "171.68.202.158"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Exec-User
        Acct-Session-Id = "00000010"
        Acct-Session-Time = 14
        Acct-Delay-Time = 0
        User-Id = "fgeorge"
        NAS-Identifier = "172.16.25.15"
```

The following example shows the information contained in a TACACS+ EXEC accounting record for a Telnet user:

```
Wed Jun 27 04:06:53 2001        172.16.25.15    fgeorge   tty26   171.68.202.158
starttask_id=41      service=shell
Wed Jun 27 04:07:02 2001        172.16.25.15    fgeorge   tty26   171.68.202.158
stoptask_id=41       service=shell   elapsed_time=9
```

System Accounting

System accounting provides information about all system-level events (for example, when the system reboots or when accounting is turned on or off).

The following accounting record shows a typical TACACS+ system accounting record server indicating that AAA accounting has been turned off:

```
Wed Jun 27 03:55:32 2001       172.16.25.15    unknown unknown unknown start   task_id=25
service=system  event=sys_acct  reason=reconfigure
```

Note The precise format of accounting packets records may vary depending on your particular TACACS+ daemon.

The following accounting record shows a TACACS+ system accounting record indicating that AAA accounting has been turned on:

```
Wed Jun 27 03:55:22 2001       172.16.25.15    unknown unknown unknown stop    task_id=23
service=system  event=sys_acct  reason=reconfigure
```

Additional tasks for measuring system resources are covered in other chapters in the Cisco IOS software configuration guides. For example, IP accounting tasks are described in the "Configuring IP Services" chapter in the *Cisco IOS IP Configuration Guide*.

Command Accounting

Command accounting provides information about the EXEC shell commands for a specified privilege level that are being executed on a network access server. Each command accounting record includes a list of the commands executed for that privilege level, as well as the date and time each command was executed, and the user who executed it.

The following example shows the information contained in a TACACS+ command accounting record for privilege level 1:

```
Wed Jun 27 03:46:47 2001       172.16.25.15    fgeorge   tty3    5622329430/4327528  stop
task_id=3       service=shell   priv-lvl=1      cmd=show version <cr>
Wed Jun 27 03:46:58 2001       172.16.25.15    fgeorge   tty3    5622329430/4327528  stop
task_id=4       service=shell   priv-lvl=1      cmd=show interfaces Ethernet 0 <cr>
Wed Jun 27 03:47:03 2001       172.16.25.15    fgeorge   tty3    5622329430/4327528  stop
task_id=5       service=shell   priv-lvl=1      cmd=show ip route <cr>
```

The following example shows the information contained in a TACACS+ command accounting record for privilege level 15:

```
Wed Jun 27 03:47:17 2001       172.16.25.15    fgeorge   tty3    5622329430/4327528  stop
task_id=6       service=shell   priv-lvl=15     cmd=configure terminal <cr>
Wed Jun 27 03:47:21 2001       172.16.25.15    fgeorge   tty3    5622329430/4327528  stop
task_id=7       service=shell   priv-lvl=15     cmd=interface Serial 0 <cr>
Wed Jun 27 03:47:29 2001       172.16.25.15    fgeorge   tty3    5622329430/4327528  stop
task_id=8       service=shell   priv-lvl=15     cmd=ip address 1.1.1.1 255.255.255.0 <cr>
```

Note The Cisco Systems implementation of RADIUS does not support command accounting.

Resource Accounting

The Cisco implementation of AAA accounting provides “start” and “stop” record support for calls that have passed user authentication. The additional feature of generating “stop” records for calls that fail to authenticate as part of user authentication is also supported. Such records are necessary for users employing accounting records to manage and monitor their networks.

This section includes the following subsections:

- AAA Resource Failure Stop Accounting
- AAA Resource Accounting for Start-Stop Records

AAA Resource Failure Stop Accounting

Before AAA resource failure stop accounting, there was no method of providing accounting records for calls that failed to reach the user authentication stage of a call setup sequence. Such records are necessary for users employing accounting records to manage and monitor their networks and their wholesale customers.

This functionality will generate a “stop” accounting record for any calls that do not reach user authentication; “stop” records will be generated from the moment of call setup. All calls that pass user authentication will behave as before; that is, no additional accounting records will be seen.

Note For Cisco IOS Release 12.2, this function is supported only on the Cisco AS5300 and Cisco AS5800.

Figure 7 illustrates a call setup sequence with normal call flow (no disconnect) and without AAA resource failure stop accounting enabled.

Figure 7 Modem Dial-In Call Setup Sequence With Normal Flow and Without Resource Failure Stop Accounting Enabled

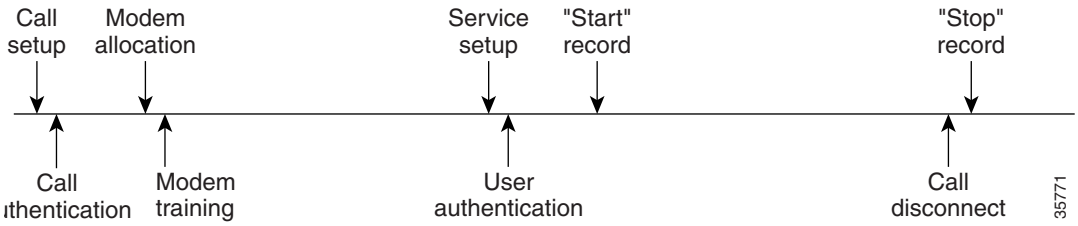


Figure 8 illustrates a call setup sequence with normal call flow (no disconnect) and with AAA resource failure stop accounting enabled.

Figure 8 Modem Dial-In Call Setup Sequence With Normal Flow and WIth Resource Failure Stop Accounting Enabled

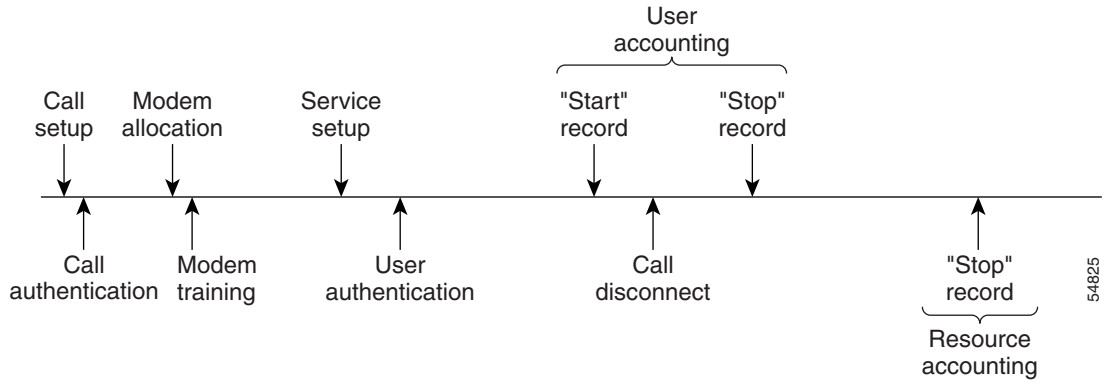


Figure 9 illustrates a call setup sequence with call disconnect occurring before user authentication and with AAA resource failure stop accounting enabled.

Figure 9 Modem Dial-In Call Setup Sequence With Call Disconnect Occurring Before User Authentication and With Resource Failure Stop Accounting Enabled

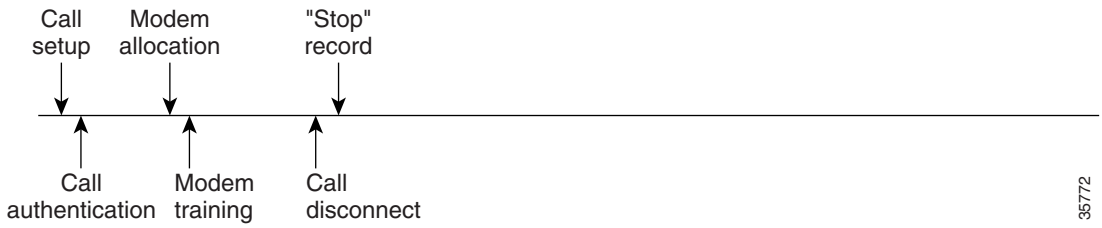
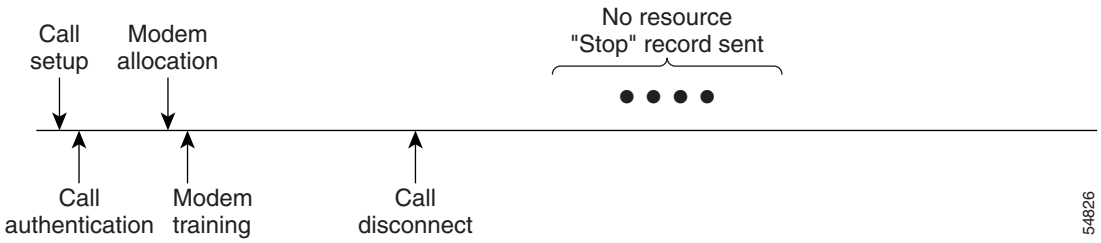


Figure 11 illustrates a call setup sequence with call disconnect occurring before user authentication and without AAA resource failure stop accounting enabled.

Figure 10 Modem Dial-In Call Setup Sequence With Call Disconnect Occurring Before User Authentication and Without Resource Failure Stop Accounting Enabled



AAA Resource Accounting for Start-Stop Records

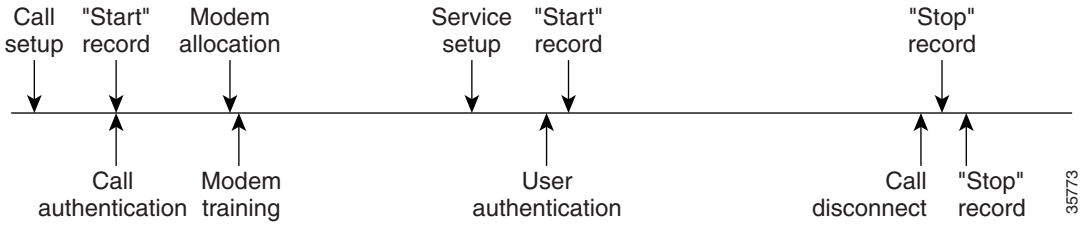
AAA resource accounting for start-stop records supports the ability to send a “start” record at each call setup, followed by a corresponding “stop” record at the call disconnect. This functionality can be used to manage and monitor wholesale customers from one source of data reporting, such as accounting records.

With this feature, a call setup and call disconnect “start-stop” accounting record tracks the progress of the resource connection to the device. A separate user authentication “start-stop” accounting record tracks the user management progress. These two sets of accounting records are interlinked by using a unique session ID for the call.

Note For Cisco IOS Release 12.2, this function is supported only on the Cisco AS5300 and Cisco AS5800.

Figure 11 illustrates a call setup sequence with AAA resource start-stop accounting enabled.

Figure 11 Modem Dial-In Call Setup Sequence With Resource Start-Stop Accounting Enabled



AAA Accounting Enhancements

The section includes the following enhancements:

- AAA Broadcast Accounting
- AAA Session MIB

AAA Broadcast Accounting

AAA broadcast accounting allows accounting information to be sent to multiple AAA servers at the same time; that is, accounting information can be broadcast to one or more AAA servers simultaneously. This functionality allows service providers to send accounting information to their own private AAA servers and to the AAA servers of their end customers. It also provides redundant billing information for voice applications.

Note Accounting information can be sent simultaneously to a maximum of four AAA servers.

Broadcasting is allowed among groups of RADIUS or TACACS+ servers, and each server group can define its backup servers for failover independently of other groups.

Thus, service providers and their end customers can use different protocols (RADIUS or TACACS+) for the accounting server. Service providers and their end customers can also specify their backup servers independently. As for voice applications, redundant accounting information can be managed independently through a separate group with its own failover sequence.

AAA Session MIB

The AAA session MIB feature allows customers to monitor and terminate their authenticated client connections using Simple Network Management Protocol (SNMP). The data of the client is presented so that it correlates directly to the AAA accounting information reported by either the RADIUS or the TACACS+ server. AAA session MIB provides the following information:

- Statistics for each AAA function (when used in conjunction with the **show radius statistics** command)
- Status of servers providing AAA functions
- Identities of external AAA servers
- Real-time information (such as idle times), providing additional criteria for use by SNMP networks for assessing whether or not to terminate an active call

Note This command is supported only on Cisco AS5300 and Cisco AS5800 universal access server platforms.

Table 11 shows the SNMP user-end data objects that can be used to monitor and terminate authenticated client connections with the AAA session MIB feature.

Table 11 SNMP End-User Data Objects

SessionId	The session identification used by the AAA accounting protocol (same value as reported by RADIUS attribute 44 (Acct-Session-ID)).
UserId	The user login ID or zero-length string if a login is unavailable.
IpAddr	The IP address of the session or 0.0.0.0 if an IP address is not applicable or unavailable.
IdleTime	The elapsed time in seconds that the session has been idle.
Disconnect	The session termination object used to disconnect the given client.
CallId	The entry index corresponding to this accounting session that the Call Tracker record stored.

Table 12 describes the AAA summary information provided by the AAA session MIB feature using SNMP on a per-system basis.

Table 12 SNMP AAA Session Summary

ActiveTableEntries	Number of sessions currently active.
ActiveTableHighWaterMark	Maximum number of sessions present at once since last system reinstallation.
TotalSessions	Total number of sessions since last system reinstallation.
DisconnectedSessions	Total number of sessions that have been disconnected using since last system reinstallation.

AAA Accounting Prerequisites

Before configuring accounting using named method lists, you must first perform the following tasks:

- Enable AAA on your network access server. For more information about enabling AAA on your Cisco router or access server, refer to the chapter "AAA Overview".
- Define the characteristics of your RADIUS or TACACS+ security server if you are issuing RADIUS or TACACS+ authorization. For more information about configuring your Cisco network access server to communicate with your RADIUS security server, refer to the chapter "Configuring RADIUS". For more information about configuring your Cisco network access server to communicate with your TACACS+ security server, refer to the chapter "Configuring TACACS+".

AAA Accounting Configuration Task List

This section describes the following configuration tasks:

- Configuring AAA Accounting Using Named Method Lists
- Suppressing Generation of Accounting Records for Null Username Sessions
- Generating Interim Accounting Records
- Generating Accounting Records for Failed Login or Session
- Specifying Accounting NETWORK-Stop Records Before EXEC-Stop Records
- Configuring AAA Resource Failure Stop Accounting
- Configuring AAA Resource Accounting for Start-Stop Records
- Configuring AAA Broadcast Accounting
- Configuring AAA Resource Failure Stop Accounting
- Configuring AAA Session MIB
- Monitoring Accounting
- Troubleshooting Accounting

For accounting configuration examples using the commands in this chapter, refer to the section "Accounting Configuration Examples" at the end of the this chapter.

Configuring AAA Accounting Using Named Method Lists

To configure AAA accounting using named method lists, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa accounting {system | network | exec | connection | commands level} {default | list-name} {start-stop | stop-only | none} [method1 [method2...]]`	Creates an accounting method list and enables accounting. The argument *list-name* is a character string used to name the list you are creating.
Step 2	`Router(config)# line [aux | console | tty | vty] line-number [ending-line-number]` or `Router(config)# interface interface-type interface-number`	Enters the line configuration mode for the lines to which you want to apply the accounting method list. or Enters the interface configuration mode for the interfaces to which you want to apply the accounting method list.
Step 3	`Router(config-line)# accounting {arap | commands level | connection | exec} {default | list-name}` or `Router(config-if)# ppp accounting {default | list-name}`	Applies the accounting method list to a line or set of lines. or Applies the accounting method list to an interface or set of interfaces.

Note System accounting does not use named method lists. For system accounting, you can define only the default method list.

This section includes the following sections:

- Accounting Types
- Accounting Record Types
- Accounting Methods

Accounting Types

Named accounting method lists are specific to the indicated type of accounting.

- **network**—To create a method list to enable authorization for all network-related service requests (including SLIP, PPP, PPP NCPs, and ARA protocols), use the **network** keyword. For example, to create a method list that provides accounting information for ARAP (network) sessions, use the **arap** keyword.
- **exec**—To create a method list that provides accounting records about user EXEC terminal sessions on the network access server, including username, date, start and stop times, use the **exec** keyword.
- **commands**—To create a method list that provides accounting information about specific, individual EXEC commands associated with a specific privilege level, use the **commands** keyword.

- **connection**—To create a method list that provides accounting information about all outbound connections made from the network access server, use the **connection** keyword.
- **resource**—Creates a method list to provide accounting records for calls that have passed user authentication or calls that failed to be authenticated.

Note System accounting does not support named method lists.

Accounting Record Types

For minimal accounting, use the **stop-only** keyword, which instructs the specified method (RADIUS or TACACS+) to send a stop record accounting notice at the end of the requested user process. For more accounting information, use the **start-stop** keyword to send a start accounting notice at the beginning of the requested event and a stop accounting notice at the end of the event. To stop all accounting activities on this line or interface, use the **none** keyword.

Accounting Methods

Table 13 lists the supported accounting methods.

Table 13 AAA Accounting Methods

Keyword	Description
group radius	Uses the list of all RADIUS servers for accounting.
group tacacs+	Uses the list of all TACACS+ servers for accounting.
group *group-name*	Uses a subset of RADIUS or TACACS+ servers for accounting as defined by the server group *group-name*.

The method argument refers to the actual method the authentication algorithm tries. Additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all other methods return an error, specify additional methods in the command. For example, to create a method list named acct_tac1 that specifies RADIUS as the backup method of authentication in the event that TACACS+ authentication returns an error, enter the following command:

```
aaa accounting network acct_tac1 stop-only group tacacs+ group radius
```

To create a default list that is used when a named list is *not* specified in the **aaa accounting** command, use the **default** keyword followed by the methods you want used in default situations. The default method list is automatically applied to all interfaces.

For example, to specify RADIUS as the default method for user authentication during login, enter the following command:

```
aaa accounting network default stop-only group radius
```

AAA accounting supports the following methods:

- **group tacacs**—To have the network access server send accounting information to a TACACS+ security server, use the **group tacacs+** *method* keyword. For more specific information about configuring TACACS+ for accounting services, refer to the chapter "Configuring TACACS+".
- **group radius**—To have the network access server send accounting information to a RADIUS security server, use the **group radius** *method* keyword. For more specific information about configuring RADIUS for accounting services, refer to the chapter "Configuring RADIUS".

Note Accounting method lists for SLIP follow whatever is configured for PPP on the relevant interface. If no lists are defined and applied to a particular interface (or no PPP settings are configured), the default setting for accounting applies.

- **group** *group-name*—To specify a subset of RADIUS or TACACS+ servers to use as the accounting method, use the **aaa accounting** command with the **group** *group-name* method. To specify and define the group name and the members of the group, use the **aaa group server** command. For example, use the **aaa group server** command to first define the members of **group loginrad**:

```
aaa group server radius loginrad
 server 172.16.2.3
 server 172.16.2 17
 server 172.16.2.32
```

This command specifies RADIUS servers 172.16.2.3, 172.16.2.17, and 172.16.2.32 as members of the group *loginrad*.

To specify **group loginrad** as the method of network accounting when no other method list has been defined, enter the following command:

```
aaa accounting network default start-stop group loginrad
```

Before you can use a group name as the accounting method, you need to enable communication with the RADIUS or TACACS+ security server. For more information about establishing communication with a RADIUS server, refer to the chapter "Configuring RADIUS". For more information about establishing communication with a TACACS+ server, refer to the chapter "Configuring TACACS+".

Suppressing Generation of Accounting Records for Null Username Sessions

When AAA accounting is activated, the Cisco IOS software issues accounting records for all users on the system, including users whose username string, because of protocol translation, is NULL. An example of this is users who come in on lines where the **aaa authentication login** *method-list* **none** command is applied. To prevent accounting records from being generated for sessions that do not have usernames associated with them, use the following command in global configuration mode:

Command	Purpose
`Router(config)# aaa accounting suppress null-username`	Prevents accounting records from being generated for users whose username string is NULL.

Generating Interim Accounting Records

To enable periodic interim accounting records to be sent to the accounting server, use the following command in global configuration mode:

Command	Purpose
`Router(config)# aaa accounting update {[newinfo] [periodic] number}`	Enables periodic interim accounting records to be sent to the accounting server.

When the **aaa accounting update** command is activated, the Cisco IOS software issues interim accounting records for all users on the system. If the keyword **newinfo** is used, interim accounting records will be sent to the accounting server every time there is new accounting information to report. An example of this would be when IPCP completes IP address negotiation with the remote peer. The interim accounting record will include the negotiated IP address used by the remote peer.

When used with the keyword **periodic**, interim accounting records are sent periodically as defined by the argument number. The interim accounting record contains all of the accounting information recorded for that user up to the time the interim accounting record is sent.

Caution Using the **aaa accounting update periodic** command can cause heavy congestion when many users are logged in to the network.

Generating Accounting Records for Failed Login or Session

When AAA accounting is activated, the Cisco IOS software does not generate accounting records for system users who fail login authentication, or who succeed in login authentication but fail PPP negotiation for some reason.

To specify that accounting stop records be generated for users who fail to authenticate at login or during session negotiation, use the following command in global configuration mode:

Command	Purpose
`Router(config)# aaa accounting send stop-record authentication failure`	Generates “stop” records for users who fail to authenticate at login or during session negotiation using PPP.

Specifying Accounting NETWORK-Stop Records Before EXEC-Stop Records

For PPP users who start EXEC terminal sessions, you can specify that NETWORK records be generated before EXEC-stop records. In some cases, such as billing customers for specific services, is can be desirable to keep network start and stop records together, essentially “nesting” them within the framework of the EXEC start and stop messages. For example, a user dialing in using PPP can create the following records: EXEC-start, NETWORK-start, EXEC-stop, NETWORK-stop. By nesting the accounting records, NETWORK-stop records follow NETWORK-start messages: EXEC-start, NETWORK-start, NETWORK-stop, EXEC-stop.

To nest accounting records for user sessions, use the following command in global configuration mode:

Command	Purpose
`Router(config)# aaa accounting nested`	Nests network accounting records.

Configuring AAA Resource Failure Stop Accounting

To enable resource failure stop accounting, use the following command in global configuration:

Command	Purpose
`Router(config)# aaa accounting resource` *method-list* **stop-failure group** *server-group*	Generates a "stop" record for any calls that do not reach user authentication. **Note** Before configuring this feature, you must first perform the tasks described in the section "AAA Accounting Prerequisites" and enable Simple Network Management Protocol on your network access server. For more information about enabling SNMP on your Cisco router or access server, refer to the chapter "Configuring SNMP" of the *Cisco IOS Configuration Fundamentals Configuration Guide*.

Configuring AAA Resource Accounting for Start-Stop Records

To enable full resource accounting for start-stop records, use the following command in global configuration mode:

Command	Purpose
`Router(config)# aaa accounting resource` *method-list* **start-stop group** *server-group*	Supports the ability to send a "start" record at each call setup. followed with a corresponding "stop" record at the call disconnect. **Note** Before configuring this feature, you must first perform the tasks described in "AAA Accounting Prerequisites" and enable Simple Network Management Protocol on your network access server. For more information about enabling SNMP on your Cisco router or access server, refer to the chapter "Configuring SNMP" chapter of the *Cisco IOS Configuration Fundamentals Configuration Guide*.

Configuring AAA Broadcast Accounting

To configure AAA broadcast accounting, use the **aaa accounting** command in global configuration mode. This command has been modified to allow the **broadcast** keyword.

Command	Purpose
`Router(config)# aaa accounting {system | network | exec | connection | commands level} {default | list-name} {start-stop | stop-only | none} [broadcast] method1 [method2...]`	Enables sending accounting records to multiple AAA servers. Simultaneously sends accounting records to the first server in each group. If the first server is unavailable, failover occurs using the backup servers defined within that group.

Configuring Per-DNIS AAA Broadcast Accounting

To configure AAA broadcast accounting per Dialed Number Identification Service (DNIS), use the **aaa dnis map accounting network** command in global configuration mode. This command has been modified to allow the **broadcast** keyword and multiple server groups.

Command	Purpose
`Router(config)# aaa dnis map dnis-number accounting network [start-stop | stop-only | none] [broadcast] method1 [method2...]`	Allows per-DNIS accounting configuration. This command has precedence over the global **aaa accounting** command. Enables sending accounting records to multiple AAA servers. Simultaneously sends accounting records to the first server in each group. If the first server is unavailable, failover occurs using the backup servers defined within that group.

Configuring AAA Session MIB

Before configuring the AAA session MIB feature, you must perform the following tasks:

- Configure SNMP. For information on SNMP, see the chapter "Configuring SNMP Support" of the *Cisco IOS Configuration Fundamentals Configuration Guide*.
- Configure AAA.
- Define the characteristics of your RADIUS or TACACS+ server.

Note Overusing SNMP can affect the overall performance of your system; therefore, normal network management performance must be considered when this feature is used.

To configure AAA session MIB, use the following command in global configuration mode

:

	Command	Purpose
Step 1	`Router(config)# aaa session-mib disconnect`	Monitors and terminates authenticated client connections using SNMP. To terminate the call, the **disconnect** keyword must be used.

Monitoring Accounting

No specific **show** command exists for either RADIUS or TACACS+ accounting. To obtain accounting records displaying information about users currently logged in, use the following command in privileged EXEC mode:

Command	Purpose
`Router# show accounting`	Allows display of the active accountable events on the network and helps collect information in the event of a data loss on the accounting server.

Troubleshooting Accounting

To troubleshoot accounting information, use the following command in privileged EXEC mode:

Command	Purpose
`Router# debug aaa accounting`	Displays information on accountable events as they occur.

Accounting Attribute-Value Pairs

The network access server monitors the accounting functions defined in either TACACS+ attribute-value (AV) pairs or RADIUS attributes, depending on which security method you have implemented. For a list of supported RADIUS accounting attributes, refer to the appendix "RADIUS Attributes." For a list of supported TACACS+ accounting AV pairs, refer to the appendix "TACACS+ Attribute-Value Pairs."

Accounting Configuration Examples

This section contains the following examples:

- Configuring Named Method List Example
- Configuring AAA Resource Accounting
- Configuring AAA Broadcast Accounting Example
- Configuring Per-DNIS AAA Broadcast Accounting Example
- AAA Session MIB Example

Configuring Named Method List Example

The following example shows how to configure a Cisco AS5200 (enabled for AAA and communication with a RADIUS security server) in order for AAA services to be provided by the RADIUS server. If the RADIUS server fails to respond, then the local database will be queried for authentication and authorization information, and accounting services will be handled by a TACACS+ server.

```
aaa new-model
aaa authentication login admins local
aaa authentication ppp dialins goup radius local
aaa authorization network scoobee group radius local
aaa accounting network charley start-stop group radius group tacacs+

username root password ALongPassword

tacacs-server host 172.31.255.0
tacacs-server key goaway

radius-server host 172.16.2.7
radius-server key myRaDiUSpassWoRd

interface group-async 1
 group-range 1 16
 encapsulation ppp
 ppp authentication chap dialins
 ppp authorization scoobee
 ppp accounting charley

line 1 16
 autoselect ppp
 autoselect during-login
 login authentication admins
 modem dialin
```

The lines in this sample RADIUS AAA configuration are defined as follows:

- The **aaa new-model** command enables AAA network security services.
- The **aaa authentication login admins local** command defines a method list, “admins”, for login authentication.
- The **aaa authentication ppp dialins group radius local** command defines the authentication method list “dialins”, which specifies that first RADIUS authentication and then (if the RADIUS server does not respond) local authentication will be used on serial lines using PPP.
- The **aaa authorization network scoobee group radius local** command defines the network authorization method list named “scoobee”, which specifies that RADIUS authorization will be used on serial lines using PPP. If the RADIUS server fails to respond, then local network authorization will be performed.
- The **aaa accounting network charley start-stop group radius group tacacs+** command defines the network accounting method list named charley, which specifies that RADIUS accounting services (in this case, start and stop records for specific events) will be used on serial lines using PPP. If the RADIUS server fails to respond, accounting services will be handled by a TACACS+ server.
- The **username** command defines the username and password to be used for the PPP Password Authentication Protocol (PAP) caller identification.
- The **tacacs-server host** command defines the name of the TACACS+ server host.

- The **tacacs-server key** command defines the shared secret text string between the network access server and the TACACS+ server host.
- The **radius-server host** command defines the name of the RADIUS server host.
- The **radius-server key** command defines the shared secret text string between the network access server and the RADIUS server host.
- The **interface group-async** command selects and defines an asynchronous interface group.
- The **group-range** command defines the member asynchronous interfaces in the interface group.
- The **encapsulation ppp** command sets PPP as the encapsulation method used on the specified interfaces.
- The **ppp authentication chap dialins** command selects Challenge Handshake Authentication Protocol (CHAP) as the method of PPP authentication and applies the "dialins" method list to the specified interfaces.
- The **ppp authorization scoobee** command applies the scoobee network authorization method list to the specified interfaces.
- The **ppp accounting charley** command applies the charley network accounting method list to the specified interfaces.
- The **line** command switches the configuration mode from global configuration to line configuration and identifies the specific lines being configured.
- The **autoselect ppp** command configures the Cisco IOS software to allow a PPP session to start up automatically on these selected lines.
- The **autoselect during-login** command is used to display the username and password prompt without pressing the Return key. After the user logs in, the autoselect function (in this case, PPP) begins.
- The **login authentication admins** command applies the admins method list for login authentication.
- The **modem dialin** command configures modems attached to the selected lines to only accept incoming calls.

The **show accounting** command yields the following output for the preceding configuration:

```
Active Accounted actions on tty1, User rubble Priv 1
 Task ID 5, Network Accounting record, 00:00:52 Elapsed
 task_id=5 service=ppp protocol=ip address=10.0.0.98
```

Table 14 describes the fields contained in the preceding output.

Table 14 show accounting Field Descriptions

Field	Description
Active Accounted actions on	Terminal line or interface name user with which the user logged in.
User	User's ID.
Priv	User's privilege level.
Task ID	Unique identifier for each accounting session.
Accounting Record	Type of accounting session.
Elapsed	Length of time (hh:mm:ss) for this session type.
attribute=value	AV pairs associated with this accounting session.

Configuring AAA Resource Accounting

The following example shows how to configure the resource failure stop accounting and resource accounting for start-stop records functions:

```
!Enable AAA on your network access server.
aaa new-model
!Enable authentication at login and list the AOL string name to use for login
authentication.
aaa authentication login AOL group radius local
!Enable authentication for ppp and list the default method to use for PPP authentication.
aaa authentication ppp default group radius local
!Enable authorization for all exec sessions and list the AOL string name to use for
authorization.
aaa authorization exec AOL group radius if-authenticated
!Enable authorization for all network-related service requests and list the default method
to use for all network-related authorizations.
aaa authorization network default group radius if-authenticated
!Enable accounting for all exec sessions and list the default method to use for all
start-stop accounting services.
aaa accounting exec default start-stop group radius
!Enable accounting for all network-related service requests and list the default method to
use for all start-stop accounting services.
aaa accounting network default start-stop group radius
!Enable failure stop accounting.
aaa accounting resource default stop-failure group radius
!Enable resource accounting for start-stop records.
aaa accounting resource default start-stop group radius
```

Configuring AAA Broadcast Accounting Example

The following example shows how to turn on broadcast accounting using the global **aaa accounting** command:

```
aaa group server radius isp
 server 1.0.0.1
 server 1.0.0.2

aaa group server tacacs+ isp_customer
 server 3.0.0.1

aaa accounting network default start-stop broadcast group isp group isp_customer

radius-server host 1.0.0.1
radius-server host 1.0.0.2
radius-server key key1
tacacs-server host 3.0.0.1 key key2
```

The **broadcast** keyword causes “start” and “stop” accounting records for network connections to be sent simultaneously to server 1.0.0.1 in the group isp and to server 3.0.0.1 in the group isp_customer. If server 1.0.0.1 is unavailable, failover to server 1.0.0.2 occurs. If server 3.0.0.1 is unavailable, no failover occurs because backup servers are not configured for the group isp_customer.

Configuring Per-DNIS AAA Broadcast Accounting Example

The following example shows how to turn on per DNIS broadcast accounting using the global **aaa dnis map accounting network** command:

```
aaa group server radius isp
 server 1.0.0.1
 server 1.0.0.2

aaa group server tacacs+ isp_customer
 server 3.0.0.1

aaa dnis map enable
aaa dnis map 7777 accounting network start-stop broadcast group isp group isp_customer

radius-server host 1.0.0.1
radius-server host 1.0.0.2
radius-server key key_1
tacacs-server host 3.0.0.1 key key_2
```

The **broadcast** keyword causes “start” and “stop” accounting records for network connection calls having DNIS number 7777 to be sent simultaneously to server 1.0.0.1 in the group isp and to server 3.0.0.1 in the group isp_customer. If server 1.0.0.1 is unavailable, failover to server 1.0.0.2 occurs. If server 3.0.0.1 is unavailable, no failover occurs because backup servers are not configured for the group isp_customer.

AAA Session MIB Example

The following example shows how to set up the AAA session MIB feature to disconnect authenticated client connections for PPP users:

```
aaa new-model
aaa authentication ppp default group radius
aaa authorization network default group radius
aaa accounting network default start-stop group radius
aaa session-mib disconnect
```

Cisco Systems

Part 2: Security Server Protocols

RADIUS

This part consists of the following:

- Configuring RADIUS
- AAA Dead-Server Detection
- ACL Default Direction
- Attribute Screening for Access Requests
- Enable Multilink PPP via RADIUS for Preauthentication User
- Enhanced Test Command
- Framed-Route in RADIUS Accounting
- Offload Server Accounting Enhancement
- Per VRF AAA
- RFC-2867 RADIUS Tunnel Accounting
- RADIUS Attribute Screening
- RADIUS Centralized Filter Management
- RADIUS Debug Enhancements
- RADIUS Logical Line ID
- RADIUS NAS-IP-Address Attribute Configurability
- RADIUS Route Download
- RADIUS Support of 56-Bit Acct Session-Id
- RADIUS Tunnel Preference for Load Balancing and Fail-Over
- RADIUS Server Reorder on Failure
- Subscriber Service Switch
- Tunnel Authentication via RADIUS on Tunnel Terminator

Configuring RADIUS

This chapter describes the Remote Authentication Dial-In User Service (RADIUS) security system, defines its operation, and identifies appropriate and inappropriate network environments for using RADIUS technology. The "RADIUS Configuration Task List" section describes how to configure RADIUS with the authentication, authorization, and accounting (AAA) command set.

For a complete description of the RADIUS commands used in this chapter, refer to the chapter "RADIUS Commands" in the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the"Finding Additional Feature Support Information" section on page cxvii in the chapter Using Cisco IOS Software for Release 12.4.

In This Chapter

This chapter includes the following sections:

- About RADIUS
- RADIUS Operation
- RADIUS Configuration Task List
- Monitoring and Maintaining RADIUS
- RADIUS Attributes
- RADIUS Configuration Examples

About RADIUS

RADIUS is a distributed client/server system that secures networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco routers and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.

RADIUS is a fully open protocol, distributed in source code format, that can be modified to work with any security system currently available on the market.

Cisco supports RADIUS under its AAA security paradigm. RADIUS can be used with other AAA security protocols, such as TACACS+, Kerberos, and local username lookup. RADIUS is supported on all Cisco platforms, but some RADIUS-supported features run only on specified platforms.

RADIUS has been implemented in a variety of network environments that require high levels of security while maintaining network access for remote users.

Use RADIUS in the following network environments that require access security:

- Networks with multiple-vendor access servers, each supporting RADIUS. For example, access servers from several vendors use a single RADIUS server-based security database. In an IP-based network with multiple vendors' access servers, dial-in users are authenticated through a RADIUS server that has been customized to work with the Kerberos security system.
- Turnkey network security environments in which applications support the RADIUS protocol, such as in an access environment that uses a "smart card" access control system. In one case, RADIUS has been used with Enigma's security cards to validate users and grant access to network resources.
- Networks already using RADIUS. You can add a Cisco router with RADIUS to the network. This might be the first step when you make a transition to a Terminal Access Controller Access Control System Plus (TACACS+) server.
- Networks in which a user must only access a single service. Using RADIUS, you can control user access to a single host, to a single utility such as Telnet, or to a single protocol such as Point-to-Point Protocol (PPP). For example, when a user logs in, RADIUS identifies this user as having authorization to run PPP using IP address 10.2.3.4 and the defined access list is started.
- Networks that require resource accounting. You can use RADIUS accounting independent of RADIUS authentication or authorization. The RADIUS accounting functions allow data to be sent at the start and end of services, indicating the amount of resources (such as time, packets, bytes, and so on) used during the session. An Internet service provider (ISP) might use a freeware-based version of RADIUS access control and accounting software to meet special security and billing needs.
- Networks that wish to support preauthentication. Using the RADIUS server in your network, you can configure AAA preauthentication and set up the preauthentication profiles. Preauthentication enables service providers to better manage ports using their existing RADIUS solutions, and to efficiently manage the use of shared resources to offer differing service-level agreements.

RADIUS is not suitable in the following network security situations:

- Multiprotocol access environments. RADIUS does not support the following protocols:
  - AppleTalk Remote Access (ARA)
  - NetBIOS Frame Control Protocol (NBFCP)
  - NetWare Asynchronous Services Interface (NASI)
  - X.25 PAD connections
- Router-to-router situations. RADIUS does not provide two-way authentication. RADIUS can be used to authenticate from one router to a non-Cisco router if the non-Cisco router requires RADIUS authentication.
- Networks using a variety of services. RADIUS generally binds a user to one service model.

RADIUS Operation

When a user attempts to log in and authenticate to an access server using RADIUS, the following steps occur:

1. The user is prompted for and enters a username and password.
2. The username and encrypted password are sent over the network to the RADIUS server.

3. The user receives one of the following responses from the RADIUS server:
   a. ACCEPT—The user is authenticated.
   b. REJECT—The user is not authenticated and is prompted to reenter the username and password, or access is denied.
   c. CHALLENGE—A challenge is issued by the RADIUS server. The challenge collects additional data from the user.
   d. CHANGE PASSWORD—A request is issued by the RADIUS server, asking the user to select a new password.

The ACCEPT or REJECT response is bundled with additional data that is used for EXEC or network authorization. You must first complete RADIUS authentication before using RADIUS authorization. The additional data included with the ACCEPT or REJECT packets consists of the following:

- Services that the user can access, including Telnet, rlogin, or local-area transport (LAT) connections, and PPP, Serial Line Internet Protocol (SLIP), or EXEC services.
- Connection parameters, including the host or client IP address, access list, and user timeouts.

RADIUS Configuration Task List

To configure RADIUS on your Cisco router or access server, you must perform the following tasks:

- Use the **aaa new-model** global configuration command to enable AAA. AAA must be configured if you plan to use RADIUS. For more information about using the **aaa new-model** command, refer to the "AAA Overview" chapter.
- Use the **aaa authentication** global configuration command to define method lists for RADIUS authentication. For more information about using the **aaa authentication** command, refer to the "Configuring Authentication" chapter.
- Use **line** and **interface** commands to enable the defined method lists to be used. For more information, refer to the "Configuring Authentication" chapter.

The following configuration tasks are optional:

- You may use the **aaa group server** command to group selected RADIUS hosts for specific services. For more information about using the **aaa group server** command, refer to the "Configuring AAA Server Groups" section in this chapter.
- You may use the **aaa dnis map** command to select RADIUS server groups based on DNIS number. To use this command, you must define RADIUS server groups using the **aaa group server** command. For more information about using the **aaa dnis map** command, refer to the section "Configuring AAA Server Group Selection Based on DNIS" in this chapter.
- You may use the **aaa authorization** global command to authorize specific user functions. For more information about using the **aaa authorization** command, refer to the chapter "Configuring Authorization."
- You may use the **aaa accounting** command to enable accounting for RADIUS connections. For more information about using the **aaa accounting** command, refer to the chapter "Configuring Accounting."
- You may use the **dialer aaa** interface configuration command to create remote site profiles that contain outgoing call attributes on the AAA server. For more information about using the **dialer aaa** command, refer to the section "Configuring Suffix and Password in RADIUS Access Requests" in this chapter.

This section describes how to set up RADIUS for authentication, authorization, and accounting on your network, and includes the following sections:

- Configuring Router to RADIUS Server Communication (Required)
- Configuring Router to Use Vendor-Specific RADIUS Attributes (Optional)
- Configuring Router for Vendor-Proprietary RADIUS Server Communication (Optional)
- Configuring Router to Query RADIUS Server for Static Routes and IP Addresses (Optional)
- Configuring Router to Expand Network Access Server Port Information (Optional)
- Configuring AAA Server Groups (Optional)
- Configuring AAA Server Groups with Deadtime (Optional)
- Configuring AAA DNIS Authentication
- Configuring AAA Server Group Selection Based on DNIS (Optional)
- Configuring AAA Preauthentication
- Configuring a Guard Timer
- Specifying RADIUS Authentication
- Specifying RADIUS Authorization (Optional)
- Specifying RADIUS Accounting (Optional)
- Configuring RADIUS Login-IP-Host (Optional)
- Configuring RADIUS Prompt (Optional)
- Configuring Suffix and Password in RADIUS Access Requests (Optional)

For RADIUS configuration examples using the commands in this chapter, refer to the section "RADIUS Configuration Examples" at the end of this chapter.

Configuring Router to RADIUS Server Communication

The RADIUS host is normally a multiuser system running RADIUS server software from Cisco (CiscoSecure ACS), Livingston, Merit, Microsoft, or another software provider. Configuring router to RADIUS server communication can have several components:

- Host name or IP address
- Authentication destination port
- Accounting destination port
- Timeout period
- Retransmission value
- Key string

RADIUS security servers are identified on the basis of their host name or IP address, host name and specific UDP port numbers, or IP address and specific UDP port numbers. The combination of the IP address and UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. In other words, this unique identifier enables RADIUS requests to be sent to multiple UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service—for example, accounting—the second host entry configured acts as fail-over backup to the first one. Using this

example, if the first host entry fails to provide accounting services, the network access server will try the second host entry configured on the same device for accounting services. (The RADIUS host entries will be tried in the order they are configured.)

A RADIUS server and a Cisco router use a shared secret text string to encrypt passwords and exchange responses.To configure RADIUS to use the AAA security commands, you must specify the host running the RADIUS server daemon and a secret text (key) string that it shares with the router.

The timeout, retransmission, and encryption key values are configurable globally for all RADIUS servers, on a per-server basis, or in some combination of global and per-server settings. To apply these settings globally to all RADIUS servers communicating with the router, use the three unique global commands: **radius-server timeout**, **radius-server retransmit**, and **radius-server key**. To apply these values on a specific RADIUS server, use the **radius-server host** command.

Note You can configure both global and per-server timeout, retransmission, and key value commands simultaneously on the same Cisco network access server. If both global and per-server functions are configured on a router, the per-server timer, retransmission, and key value commands override global timer, retransmission, and key value commands.

To configure per-server RADIUS server communication, use the following command in global configuration mode:

Command	Purpose
`Router(config)# radius-server host {hostname | ip-address} [auth-port port-number] [acct-port port-number] [timeout seconds] [retransmit retries] [key string] [alias {hostname | ip address}]`	Specifies the IP address or host name of the remote RADIUS server host and assign authentication and accounting destination port numbers. Use the **auth-port** *port-number* option to configure a specific UDP port on this RADIUS server to be used solely for authentication. Use the **acct-port** *port-number* option to configure a specific UDP port on this RADIUS server to be used solely for accounting. Use the **alias** keyword to configure up to eight multiple IP addresses for use when referring to RADIUS servers. To configure the network access server to recognize more than one host entry associated with a single IP address, simply repeat this command as many times as necessary, making sure that each UDP port number is different. Set the timeout, retransmit, and encryption key values to use with the specific RADIUS host. If no timeout is set, the global value is used; otherwise, enter a value in the range 1 to 1000. If no retransmit value is set, the global value is used; otherwise enter a value in the range 1 to 1000. If no key string is specified, the global value is used. **Note** The key is a text string that must match the encryption key used on the RADIUS server. Always configure the key as the last item in the **radius-server host** command syntax because the leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in your key, do not enclose the key in quotation marks unless the quotation marks themselves are part of the key.

To configure global communication settings between the router and a RADIUS server, use the following **radius-server** commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# radius-server key {0 string | 7 string | string}`	Specifies the shared secret text string used between the router and a RADIUS server. Use the **0** *line* option to configure an unencrypted shared secret. Use the **7** *line* option to configure an encrypted shared secret.
Step 2	`Router(config)# radius-server retransmit retries`	Specifies how many times the router transmits each RADIUS request to the server before giving up (the default is 3).
Step 3	`Router(config)# radius-server timeout seconds`	Specifies for how many seconds a router waits for a reply to a RADIUS request before retransmitting the request.
Step 4	`Router(config)# radius-server deadtime minutes`	Specifies for how many minutes a RADIUS server that is not responding to authentication requests is passed over by requests for RADIUS authentication.

Configuring Router to Use Vendor-Specific RADIUS Attributes

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating vendor-specific information between the network access server and the RADIUS server by using the vendor-specific attribute (attribute 26). Vendor-specific attributes (VSAs) allow vendors to support their own extended attributes not suitable for general use. The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. Cisco's vendor-ID is 9, and the supported option has vendor-type 1, which is named "cisco-avpair." The value is a string of the following format:

```
protocol : attribute sep value *
```

"Protocol" is a value of the Cisco "protocol" attribute for a particular type of authorization; protocols that can be used include IP, IPX, VPDN, VOIP, SHELL, RSVP, SIP, AIRNET, OUTBOUND. "Attribute" and "value" are an appropriate attribute-value (AV) pair defined in the Cisco TACACS+ specification, and "sep" is "=" for mandatory attributes and "*" for optional attributes. This allows the full set of features available for TACACS+ authorization to also be used for RADIUS.

For example, the following AV pair causes Cisco's "multiple named ip address pools" feature to be activated during IP authorization (during PPP's IPCP address assignment):

```
cisco-avpair= ″ip:addr-pool=first″
```

If you insert an "*", the AV pair "ip:addr-pool=first" becomes optional. Note that any AV pair can be made optional.

```
cisco-avpair= ″ip:addr-pool*first″
```

The following example shows how to cause a user logging in from a network access server to have immediate access to EXEC commands:

```
cisco-avpair= ″shell:priv-lvl=15″
```

Other vendors have their own unique vendor-IDs, options, and associated VSAs. For more information about vendor-IDs and VSAs, refer to RFC 2138, Remote Authentication Dial-In User Service (RADIUS).

To configure the network access server to recognize and use VSAs, use the following command in global configuration mode:

Command	Purpose
`Router(config)# radius-server vsa send [accounting | authentication]`	Enables the network access server to recognize and use VSAs as defined by RADIUS IETF attribute 26.

For a complete list of RADIUS attributes or more information about vendor-specific attribute 26, refer to the appendix "RADIUS Attributes."

Configuring Router for Vendor-Proprietary RADIUS Server Communication

Although an Internet Engineering Task Force (IETF) draft standard for RADIUS specifies a method for communicating vendor-proprietary information between the network access server and the RADIUS server, some vendors have extended the RADIUS attribute set in a unique way. Cisco IOS software supports a subset of vendor-proprietary RADIUS attributes.

As mentioned earlier, to configure RADIUS (whether vendor-proprietary or IETF draft-compliant), you must specify the host running the RADIUS server daemon and the secret text string it shares with the Cisco device. You specify the RADIUS host and secret text string by using the **radius-server** commands. To identify that the RADIUS server is using a vendor-proprietary implementation of RADIUS, use the **radius-server host non-standard** command. Vendor-proprietary attributes will not be supported unless you use the **radius-server host non-standard** command.

To specify a vendor-proprietary RADIUS server host and a shared secret text string, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# radius-server host {hostname | ip-address} non-standard`	Specifies the IP address or host name of the remote RADIUS server host and identifies that it is using a vendor-proprietary implementation of RADIUS.
Step 2	`Router(config)# radius-server key {0 string | 7 string | string}`	Specifies the shared secret text string used between the router and the vendor-proprietary RADIUS server. The router and the RADIUS server use this text string to encrypt passwords and exchange responses.

Configuring Router to Query RADIUS Server for Static Routes and IP Addresses

Some vendor-proprietary implementations of RADIUS let the user define static routes and IP pool definitions on the RADIUS server instead of on each individual network access server in the network. Each network access server then queries the RADIUS server for static route and IP pool information.

To have the Cisco router or access server query the RADIUS server for static routes and IP pool definitions when the device first starts up, use the following command in global configuration mode:

Command	Purpose
`Router(config)# radius-server configure-nas`	Tells the Cisco router or access server to query the RADIUS server for the static routes and IP pool definitions used throughout its domain.

Note Because the **radius-server configure-nas** command is performed when the Cisco router starts up, it will not take effect until you issue a **copy system:running config nvram:startup-config** command.

Configuring Router to Expand Network Access Server Port Information

There are some situations when PPP or login authentication occurs on an interface different from the interface on which the call itself comes in. For example, in a V.120 ISDN call, login or PPP authentication occurs on a virtual asynchronous interface “ttt” but the call itself occurs on one of the channels of the ISDN interface.

The **radius-server attribute nas-port extended** command configures RADIUS to expand the size of the NAS-Port attribute (RADIUS IETF attribute 5) field to 32 bits. The upper 16 bits of the NAS-Port attribute display the type and number of the controlling interface; the lower 16 bits indicate the interface undergoing authentication.

To display expanded interface information in the NAS-Port attribute field, use the following command in global configuration mode:

Command	Purpose
`Router(config)# radius-server attribute nas-port format`	Expands the size of the NAS-Port attribute from 16 to 32 bits to display extended interface information.

Note This command replaces the **radius-server extended-portnames** command and the **radius-server attribute nas-port extended** command.

On platforms with multiple interfaces (ports) per slot, the Cisco RADIUS implementation will not provide a unique NAS-Port attribute that permits distinguishing between the interfaces. For example, if a dual PRI interface is in slot 1, calls on both Serial1/0:1 and Serial1/1:1 will appear as NAS-Port = 20101.

Once again, this is because of the 16-bit field size limitation associated with RADIUS IETF NAS-Port attribute. In this case, the solution is to replace the NAS-Port attribute with a vendor-specific attribute (RADIUS IETF attribute 26). Cisco's vendor-ID is 9, and the Cisco-NAS-Port attribute is subtype 2.

Vendor-specific attributes (VSAs) can be turned on by entering the **radius-server vsa send** command. The port information in this attribute is provided and configured using the **aaa nas port extended** command.

To replace the NAS-Port attribute with RADIUS IETF attribute 26 and to display extended field information, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# radius-server vsa send [accounting | authentication]`	Enables the network access server to recognize and use vendor-specific attributes as defined by RADIUS IETF attribute 26.
Step 2	`Router(config)# aaa nas port extended`	Expands the size of the VSA NAS-Port field from 16 to 32 bits to display extended interface information.

The standard NAS-Port attribute (RADIUS IETF attribute 5) will continue to be sent. If you do not want this information to be sent, you can suppress it by using the **no radius-server attribute nas-port** command. When this command is configured, the standard NAS-Port attribute will no longer be sent.

For a complete list of RADIUS attributes, refer to the appendix "RADIUS Attributes."

For information about configuring RADIUS port identification for PPP, see the *Cisco IOS Wide-Area Networking Configuration Guide*.

Configuring AAA Server Groups

Configuring the router to use AAA server groups provides a way to group existing server hosts. This allows you to select a subset of the configured server hosts and use them for a particular service. A server group is used in conjunction with a global server-host list. The server group lists the IP addresses of the selected server hosts.

Server groups also can include multiple host entries for the same server, as long as each entry has a unique identifier. The combination of an IP address and a UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. In other words, this unique identifier enables RADIUS requests to be sent to different UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service—for example, accounting—the second host entry configured acts as failover backup to the first one. Using this example, if the first host entry fails to provide accounting services, the network access server will try the second host entry configured on the same device for accounting services. (The RADIUS host entries will be tried in the order in which they are configured.)

To define a server host with a server group name, enter the following commands in global configuration mode. The listed server must exist in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# radius-server host {hostname | ip-address} [auth-port port-number] [acct-port port-number] [timeout seconds] [retransmit retries] [key string] [alias {hostname | ip address}]`	Specifies and defines the IP address of the server host before configuring the AAA server-group. Refer to the section "Configuring Router to RADIUS Server Communication" of this chapter for more information on the **radius-server host** command.
Step 2	`Router(config-if)# aaa group server {radius | tacacs+} group-name`	Defines the AAA server group with a group name. All members of a group must be the same type; that is, RADIUS or TACACS+. This command puts the router in server group subconfiguration mode.
Step 3	`Router(config-sg)# server ip-address [auth-port port-number] [acct-port port-number]`	Associates a particular RADIUS server with the defined server group. Each security server is identified by its IP address and UDP port number. Repeat this step for each RADIUS server in the AAA server group. **Note** Each server in the group must be defined previously using the **radius-server host** command.

Configuring AAA Server Groups with Deadtime

After you have configured a server host with a server name, you can use the **deadtime** command to configure each server per server group. Configuring deadtime within a server group allows you to direct AAA traffic to separate groups of servers that have different operational characteristics.

Configuring deadtime is no longer limited to a global configuration. A separate timer has been attached to each server host in every server group. Therefore, when a server is found to be unresponsive after numerous retransmissions and timeouts, the server is assumed to be dead. The timers attached to each server host in all server groups are triggered. In essence, the timers are checked and subsequent requests to a server (once it is assumed to be dead) are directed to alternate timers, if configured. When the network access server receives a reply from the server, it checks and stops all configured timers (if running) for that server in all server groups.

If the timer has expired, only the server to which the timer is attached is assumed to be alive. This becomes the only server that can be tried for later AAA requests using the server groups to which the timer belongs.

Note Since one server has different timers and may have different deadtime values configured in the server groups, the same server may in the future have different states (dead and alive) at the same time.

Note To change the state of a server, you must start and stop all configured timers in all server groups.

The size of the server group will be slightly increased because of the addition of new timers and the deadtime attribute. The overall impact of the structure depends on the number and size of the server groups and how the servers are shared among server groups in a specific configuration.

To configure deadtime within a server group, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa group server radius group1`	Defines a RADIUS type server group.
Step 2	`Router(config-sg)# deadtime 1`	Configures and defines deadtime value in minutes. **Note** Local server group deadtime will override the global configuration. If omitted from the local server group configuration, the value will be inherited from the master list.
Step 3	`Router(config-sg)# exit`	Exits server group configuration mode.

Configuring AAA DNIS Authentication

DNIS preauthentication enables preauthentication at call setup based on the number dialed. The DNIS number is sent directly to the security server when a call is received. If authenticated by AAA, the call is accepted.

To configure DNIS authentication, perform the following tasks in global configuration mode:

	Command	Purpose
Step 1	`Router# config term`	Enters global configuration mode.
Step 2	`Router(config)# aaa preauth`	Enters AAA preauthentication mode.
Step 3	`Router(config-preauth)# group {radius | tacacs+ | server-group}`	(Optional) Selects the security server to use for AAA preauthentication requests. The default is RADIUS.
Step 4	`Router(config-preauth)# dnis [password string]`	Enables preauthentication using DNIS and optionally specifies a password to use in Access-Request packets.

Configuring AAA Server Group Selection Based on DNIS

Cisco IOS software allows you to assign a Dialed Number Identification Service (DNIS) number to a particular AAA server group so that the server group can process authentication, authorization, and accounting requests for users dialing in to the network using that particular DNIS. Any phone line (a regular home phone or a commercial T1/PRI line) can be associated with several phone numbers. The DNIS number identifies the number that was called to reach you.

For example, suppose you want to share the same phone number with several customers, but you want to know which customer is calling before you pick up the phone. You can customize how you answer the phone because DNIS allows you to know which customer is calling when you answer.

Cisco routers with either ISDN or internal modems can receive the DNIS number. This functionality allows users to assign different RADIUS server groups for different customers (that is, different RADIUS servers for different DNIS numbers). Additionally, using server groups you can specify the same server group for AAA services or a separate server group for each AAA service.

Cisco IOS software provides the flexibility to implement authentication and accounting services in several ways:

- Globally—AAA services are defined using global configuration access list commands and applied in general to all interfaces on a specific network access server.
- Per Interface—AAA services are defined using interface configuration commands and applied specifically to the interface being configured on a specific network access server.
- DNIS mapping—You can use DNIS to specify an AAA server to supply AAA services.

Because each of these AAA configuration methods can be configured simultaneously, Cisco has established an order of precedence to determine which server or groups of servers provide AAA services. The order of precedence is as follows:

- Per DNIS—If you configure the network access server to use DNIS to identify/determine which server group provides AAA services, then this method takes precedence over any additional AAA selection method.
- Per interface—If you configure the network access server per interface to use access lists to determine how a server provides AAA services, this method takes precedence over any global configuration AAA access lists.
- Globally—If you configure the network access server by using global AAA access lists to determine how the security server provides AAA services, this method has the least precedence.

Note Prior to configuring AAA Server Group Selection Based on DNIS, you must configure the list of RADIUS server hosts and configure the AAA server groups. See the sections "Configuring Router to RADIUS Server Communication" and "Configuring AAA Server Groups" of this chapter.

To configure the router to select a particular AAA server group based on the DNIS of the server group, configure DNIS mapping. To map a server group with a group name with DNIS number, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa dnis map enable`	Enables DNIS mapping.
Step 2	`Router(config)# aaa dnis map` *dnis-number* `authentication ppp group server-`*group-name*	Maps a DNIS number to a defined AAA server group; the servers in this server group are being used for authentication.
Step 3	`Router(config)# aaa dnis map dnis-number authorization network group server-group-name`	Maps a DNIS number to a defined AAA server group; the servers in this server group are being used for authorization.
Step 4	`Router(config)# aaa dnis map` *dnis-number* `accounting network [none | start-stop | stop-only] group` *server-group-name*	Maps a DNIS number to a defined AAA server group; the servers in this server group are being used for accounting.

Configuring AAA Preauthentication

Configuring AAA preauthentication with ISDN PRI or channel-associated signalling (CAS) allows service providers to better manage ports using their existing RADIUS solutions and efficiently manage the use of shared resources to offer differing service-level agreements. With ISDN PRI or CAS, information about an incoming call is available to the network access server (NAS) before the call is connected. The available call information includes the following:

- The Dialed Number Identification Service (DNIS) number, also referred to as the called number
- The Calling Line Identification (CLID) number, also referred to as the calling number
- The call type, also referred to as the bearer capability

This feature allows a Cisco NAS to decide—on the basis of the DNIS number, the CLID number, or the call type—whether to connect an incoming call. (With ISDN PRI, it enables user authentication and authorization before a call is answered. With CAS, the call must be answered; however, the call can be dropped if preauthentication fails.)

When an incoming call arrives from the public network switch, but before it is connected, AAA preauthentication enables the NAS to send the DNIS number, CLID number, and call type to a RADIUS server for authorization. If the server authorizes the call, then the NAS accepts the call. If the server does not authorize the call, then the NAS sends a disconnect message to the public network switch to reject the call.

In the event that the RADIUS server application becomes unavailable or is slow to respond, a guard timer can be set in the NAS. When the timer expires, the NAS uses a configurable parameter to accept or reject the incoming call that has no authorization.

This feature supports the use of attribute 44 by the RADIUS server application and the use of RADIUS attributes that are configured in the RADIUS preauthentication profiles to specify preauthentication behavior. They may also be used, for instance, to specify whether subsequent authentication should occur and, if so, what authentication method should be used.

The following restrictions apply to AAA preauthentication with ISDN PRI and CAS:

- Attribute 44 is available for CAS calls only when preauthentication or resource pooling is enabled.
- MMP is not available with ISDN PRI.
- AAA preauthentication is available only on the Cisco AS5300, Cisco AS5400, and Cisco AS5800 platforms.

Note Prior to configuring AAA preauthentication, you must enable the **aaa new-model** command and make sure the supporting preauthentication application is running on a RADIUS server in your network.

To configure AAA preauthentication, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa preauth`	Enters AAA preauthentication configuration mode.
Step 2	`Router(config-preauth)# group` *server-group*	Specifies the AAA RADIUS server group to use for preauthentication.
Step 3	`Router(config-preauth)# clid [if-avail | required] [accept-stop] [password` *string*`]`	Preauthenticates calls on the basis of the CLID number.

	Command	Purpose
Step 4	`Router(config-preauth)# ctype [if-avail | required] [accept-stop] [password string]`	Preauthenticates calls on the basis of the call type.
Step 5	`Router(config-preauth)# dnis [if-avail | required] [accept-stop] [password string]`	Preauthenticates calls on the basis of the DNIS number.
Step 6	`Router(config-preauth)# dnis bypass {dnis-group-name}`	Specifies a group of DNIS numbers that will be bypassed for preauthentication.

To configure DNIS preauthentication, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa preauth`	Enters AAA preauthentication mode.
Step 2	`Router(config-preauth)# group {radius | tacacs+ | server-group}`	(Optional) Selects the security server to use for AAA preauthentication requests. The default is RADIUS.
Step 3	`Router(config-preauth)# dnis [password string]`	Enables preauthentication using DNIS and optionally specifies a password to use in Access-Request packets.

In addition to configuring preauthentication on your Cisco router, you must set up the preauthentication profiles on the RADIUS server. For information on setting up the preauthentication profiles, see the following sections:

- Setting Up the RADIUS Profile for DNIS or CLID Preauthentication
- Setting Up the RADIUS Profile for Call Type Preauthentication
- Setting Up the RADIUS Profile for Preauthentication Enhancements for Callback
- Setting Up the RADIUS Profile for a Remote Host Name Used for Large-Scale Dial-Out
- Setting Up the RADIUS Profile for Modem Management
- Setting Up the RADIUS Profile for Subsequent Authentication
- Setting Up the RADIUS Profile for Subsequent Authentication Type
- Setting Up the RADIUS Profile to Include the Username
- Setting Up the RADIUS Profile for Two-Way Authentication
- Setting Up the RADIUS Profile to Support Authorization

Setting Up the RADIUS Profile for DNIS or CLID Preauthentication

To set up the RADIUS preauthentication profile, use the DNIS or CLID number as the username, and use the password defined in the **dnis** or **clid** command as the password.

Note The preauthentication profile must have “outbound” as the service type because the password is predefined on the NAS. Setting up the preauthentication profile in this manner prevents users from trying to log in to the NAS with the username of the DNIS number, CLID number, or call type and an obvious password. The “outbound” service type is also included in the access-request packet sent to the RADIUS server.

Setting Up the RADIUS Profile for Call Type Preauthentication

To set up the RADIUS preauthentication profile, use the call type string as the username, and use the password defined in the **ctype** command as the password. The following table shows the call type strings that may be used in the preauthentication profile:

Call Type String	ISDN Bearer Capabilities
digital	Unrestricted digital, restricted digital.
speech	Speech, 3.1 kHz audio, 7 kHz audio. **Note** This is the only call type available for CAS.
v.110	Anything with V.110 user information layer.
v.120	Anything with V.120 user information layer.

Note The preauthentication profile must have “outbound” as the service type because the password is predefined on the NAS. Setting up the preauthentication profile in this manner prevents users from trying to log in to the NAS with the username of the DNIS number, CLID number, or call type and an obvious password. The “outbound” service type is also included in the access-request packet sent to the RADIUS server and should be a check-in item if the RADIUS server supports check-in items.

Setting Up the RADIUS Profile for Preauthentication Enhancements for Callback

Callback allows remote network users such as telecommuters to dial in to the NAS without being charged. When callback is required, the NAS hangs up the current call and dials the caller back. When the NAS performs the callback, only information for the outgoing connection is applied. The rest of the attributes from the preauthentication access-accept message are discarded.

Note The destination IP address is not required to be returned from the RADIUS server.

The following example shows a RADIUS profile configuration with a callback number of 555-1111 and the service type set to outbound. The cisco-avpair = “preauth:send-name=<string>” uses the string “andy” and the cisco-avpair = “preauth:send-secret=<string>” uses the password “cisco.”

```
5551111 password = "cisco", Service-Type = Outbound
    Service-Type = Callback-Framed
    Framed-Protocol = PPP,
    Dialback-No = "5551212"
    Class = "ISP12"
    cisco-avpair = "preauth:send-name=andy"
    cisco-avpair = "preauth:send-secret=cisco"
```

Setting Up the RADIUS Profile for a Remote Host Name Used for Large-Scale Dial-Out

The following example adds to the previous example by protecting against accidentally calling a valid telephone number but accessing the wrong router by providing the name of the remote, for use in large-scale dial-out:

```
5551111 password = "cisco", Service-Type = Outbound
       Service-Type = Callback-Framed
       Framed-Protocol = PPP,
       Dialback-No = "5551212"
       Class = "ISP12"
       cisco-avpair = "preauth:send-name=andy"
       cisco-avpair = "preauth:send-secret=cisco"
       cisco-avpair = "preauth:remote-name=Router2"
```

Setting Up the RADIUS Profile for Modem Management

When DNIS, CLID, or call type preauthentication is used, the affirmative response from the RADIUS server may include a modem string for modem management in the NAS through vendor-specific attribute (VSA) 26. The modem management VSA has the following syntax:

```
cisco-avpair = "preauth:modem-service=modem min-speed <x> max-speed <y>
modulation <z> error-correction <a> compression <b>"
```

The modem management string within the VSA may contain the following:

Command	Argument
min-speed	<300 to 56000>, any
max-speed	<300 to 56000>, any
modulation	K56Flex, v22bis, v32bis, v34, v90, any
error-correction	lapm, mnp4
compression	mnp5, v42bis

When the modem management string is received from the RADIUS server in the form of a VSA, the information is passed to the Cisco IOS software and applied on a per-call basis. Modem ISDN channel aggregation (MICA) modems provide a control channel through which messages can be sent during the call setup time. Hence, this modem management feature is supported only with MICA modems and newer technologies. This feature is not supported with Microcom modems.

For more information on modem management, refer to the "Modem Configuration and Management" chapter of the *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2.

Setting Up the RADIUS Profile for Subsequent Authentication

If preauthentication passes, you may use vendor-proprietary RADIUS attribute 201 (Require-Auth) in the preauthentication profile to determine whether subsequent authentication is to be performed. If attribute 201, returned in the access-accept message, has a value of 0, then subsequent authentication will not be performed. If attribute 201 has a value of 1, then subsequent authentication will be performed as usual.

Attribute 201 has the following syntax:

```
cisco-avpair = "preauth:auth-required=<n>"
```

where <*n*> has the same value range as attribute 201 (that is, 0 or 1).

If attribute 201 is missing in the preauthentication profile, then a value of 1 is assumed, and subsequent authentication is performed.

Note To perform subsequent authentication, you must set up a regular user profile in addition to a preauthentication profile.

Setting Up the RADIUS Profile for Subsequent Authentication Type

If you have specified subsequent authentication in the preauthentication profile, you must also specify the authentication types to be used for subsequent authentication. To specify the authentication types allowed in subsequent authentication, use the following VSA:

```
cisco-avpair = "preauth:auth-type=<string>"
```

where <*string*> can be one of the following:

String	Description
chap	Requires username and password of CHAP for PPP authentication.
ms-chap	Requires username and password of MS-CHAP for PPP authentication.
pap	Requires username and password of PAP for PPP authentication.

To specify that multiple authentication types are allowed, you can configure more than one instance of this VSA in the preauthentication profile. The sequence of the authentication type VSAs in the preauthentication profile is significant because it specifies the order of authentication types to be used in the PPP negotiation.

This VSA is a per-user attribute and replaces the authentication type list in the **ppp authentication** interface command.

Note You should use this VSA only if subsequent authentication is required because it specifies the authentication type for subsequent authentication.

Setting Up the RADIUS Profile to Include the Username

If only preauthentication is used to authenticate a call, the NAS could be missing a username when it brings up the call. RADIUS may provide a username for the NAS to use through RADIUS attribute 1 (User-Name) or through a VSA returned in the access-accept packet. The VSA for specifying the username has the following syntax:

```
cisco-avpair = "preauth:username=<string>"
```

If no username is specified, the DNIS number, CLID number, or call type is used, depending on the last preauthentication command that has been configured (for example, if **clid** was the last preauthentication command configured, the CLID number will be used as the username).

If subsequent authentication is used to authenticate a call, there might be two usernames: one provided by RADIUS and one provided by the user. In this case, the username provided by the user overrides the one contained in the RADIUS preauthentication profile; the username provided by the user is used for both authentication and accounting.

Setting Up the RADIUS Profile for Two-Way Authentication

In the case of two-way authentication, the calling networking device will need to authenticate the NAS. The Password Authentication Protocol (PAP) username and password or Challenge Handshake Authentication Protocol (CHAP) username and password need not be configured locally on the NAS. Instead, username and password can be included in the access-accept messages for preauthentication.

Note The **ppp authentication** command must be configured with the **radius** method.

To apply for PAP, do not configure the **ppp pap sent-name password** command on the interface. The vendor-specific attributes (VSAs) “preauth:send-name” and “preauth:send-secret” will be used as the PAP username and PAP password for outbound authentication.

For CHAP, “preauth:send-name” will be used not only for outbound authentication, but also for inbound authentication. For a CHAP inbound case, the NAS will use the name defined in “preauth:send-name” in the challenge packet to the caller networking device. For a CHAP outbound case, both “preauth:send-name” and “preauth:send-secret” will be used in the response packet.

The following example shows a configuration that specifies two-way authentication:

```
5551111 password = "cisco", Service-Type = Outbound
    Service-Type = Framed-User
    cisco-avpair = "preauth:auth-required=1"
    cisco-avpair = "preauth:auth-type=pap"
    cisco-avpair = "preauth:send-name=andy"
    cisco-avpair = "preauth:send-secret=cisco"
    class = "<some class>"
```

Note Two-way authentication does not work when resource pooling is enabled.

Setting Up the RADIUS Profile to Support Authorization

If only preauthentication is configured, then subsequent authentication will be bypassed. Note that because the username and password are not available, authorization will also be bypassed. However, you may include authorization attributes in the preauthentication profile to apply per-user attributes and avoid having to return subsequently to RADIUS for authorization. To initiate the authorization process, you must also configure the **aaa authorization network** command on the NAS.

You may configure authorization attributes in the preauthentication profile with one exception: the service-type attribute (attribute 6). The service-type attribute must be converted to a VSA in the preauthentication profile. This VSA has the following syntax:

```
cisco-avpair = “preauth:service-type=<n>”
```

where <*n*> is one of the standard RFC 2138 values for attribute 6. For a list of possible Service-Type values, refer to the appendix RADIUS Attributes.

Note If subsequent authentication is required, the authorization attributes in the preauthentication profile will not be applied.

Configuring a Guard Timer

Because response times for preauthentication and authentication requests can vary, the guard timer allows you to control the handling of calls. The guard timer starts when the DNIS is sent to the RADIUS server. If the NAS does not receive a response from AAA before the guard timer expires, it accepts or rejects the calls on the basis of the configuration of the timer.

To set a guard timer to accept or reject a call in the event that the RADIUS server fails to respond to an authentication or preauthentication request, use one of the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# isdn guard-timer` *milliseconds* `[on-expiry {accept | reject}]`	Sets an ISDN guard timer to accept or reject a call in the event that the RADIUS server fails to respond to a preauthentication request.
`Router(control-config)# call guard-timer` *milliseconds* `[on-expiry {accept | reject}]`	Sets a CAS guard timer to accept or reject a call in the event that the RADIUS server fails to respond to a preauthentication request.

Specifying RADIUS Authentication

After you have identified the RADIUS server and defined the RADIUS authentication key, you must define method lists for RADIUS authentication. Because RADIUS authentication is facilitated through AAA, you must enter the **aaa authentication** command, specifying RADIUS as the authentication method. For more information, refer to the chapter "Configuring Authentication."

Specifying RADIUS Authorization

AAA authorization lets you set parameters that restrict a user's access to the network. Authorization using RADIUS provides one method for remote access control, including one-time authorization or authorization for each service, per-user account list and profile, user group support, and support of IP, IPX, ARA, and Telnet. Because RADIUS authorization is facilitated through AAA, you must issue the **aaa authorization** command, specifying RADIUS as the authorization method. For more information, refer to the chapter "Configuring Authorization."

Specifying RADIUS Accounting

The AAA accounting feature enables you to track the services users are accessing as well as the amount of network resources they are consuming. Because RADIUS accounting is facilitated through AAA, you must issue the **aaa accounting** command, specifying RADIUS as the accounting method. For more information, refer to the chapter "Configuring Accounting."

Configuring RADIUS Login-IP-Host

To enable the network access server to attempt more than one login host when trying to connect a dial in user, you can enter as many as three Login-IP-Host entries in the user's profile on the RADIUS server. The following example shows that three Login-IP-Host instances have been configured for the user *joeuser,* and that TCP-Clear will be used for the connection:

```
joeuser       Password = xyz
        Service-Type = Login,
        Login-Service = TCP-Clear,
        Login-IP-Host = 10.0.0.0,
        Login-IP-Host = 10.2.2.2,
        Login-IP-Host = 10.255.255.255,
        Login-TCP-Port = 23
```

The order in which the hosts are entered is the order in which they are attempted. Use the **ip tcp synwait-time** command to set the number of seconds that the network access server waits before trying to connect to the next host on the list; the default is 30 seconds.

Your RADIUS server might permit more than three Login-IP-Host entries; however, the network access server supports only three hosts in access-accept packets.

Configuring RADIUS Prompt

To control whether user responses to access-challenge packets are echoed to the screen, you can configure the Prompt attribute in the user profile on the RADIUS server. This attribute is included only in access-challenge packets. The following example shows the Prompt attribute set to No-Echo, which prevents the user's responses from echoing:

```
joeuser Password = xyz
Service-Type = Login,
Login-Service = Telnet,
Prompt = No-Echo,
Login-IP-Host = 172.31.255.255
```

To allow user responses to echo, set the attribute to Echo. If the Prompt attribute is not included in the user profile, responses are echoed by default.

This attribute overrides the behavior of the **radius-server challenge-noecho** command configured on the access server. For example, if the access server is configured to suppress echoing, but the individual user profile allows echoing, then the user responses are echoed.

Note To use the Prompt attribute, your RADIUS server must be configured to support access-challenge packets.

Configuring Suffix and Password in RADIUS Access Requests

Large-scale dial-out eliminates the need to configure dialer maps on every NAS for every destination. Instead, you can create remote site profiles that contain outgoing call attributes on the AAA server. The profile is downloaded by the NAS when packet traffic requires a call to be placed to a remote site.

You can configure the username in the access-request message to RADIUS. The default suffix of the username, "-out," is appended to the username. The format for composing the username attribute is IP address plus configured suffix.

To provide username configuration capability for large-scale dial-out, the **dialer aaa** command is implemented with the new **suffix** and **password** keywords.

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables the AAA access control model.
Step 2	`Router(config)# aaa route download` *min*	Enables the download static route feature and sets the amount of time between downloads.
Step 3	`Router(config)# aaa authorization configuration default`	Downloads static route configuration information from the AAA server using TACACS+ or RADIUS.
Step 4	`Router(config)# interface dialer` *1*	Defines a dialer rotary group.
Step 5	`Router(config-if)# dialer aaa`	Allows a dialer to access the AAA server for dialing information.
Step 6	`Router(config-if)# dialer aaa suffix` *suffix* `password` *password*	Allows a dialer to access the AAA server for dialing information and specifies a suffix and nondefault password for authentication.

Monitoring and Maintaining RADIUS

To monitor and maintain RADIUS, use the following commands in privileged EXEC mode:

Command	Purpose
`Router# debug radius`	Displays information associated with RADIUS.
`Router# show radius statistics`	Displays the RADIUS statistics for accounting and authentication packets.

RADIUS Attributes

The network access server monitors the RADIUS authorization and accounting functions defined by RADIUS attributes in each user-profile. For a list of supported RADIUS attributes, refer to the appendix "RADIUS Attributes."

This section includes the following sections:

- Vendor-Proprietary RADIUS Attributes
- RADIUS Tunnel Attributes

Vendor-Proprietary RADIUS Attributes

An Internet Engineering Task Force (IETF) draft standard for RADIUS specifies a method for communicating vendor-proprietary information between the network access server and the RADIUS server. Some vendors, nevertheless, have extended the RADIUS attribute set in a unique way. Cisco IOS software supports a subset of vendor-proprietary RADIUS attributes. For a list of supported vendor-proprietary RADIUS attributes, refer to the appendix "RADIUS Attributes."

RADIUS Tunnel Attributes

RADIUS is a security server authentication, authorization, and accounting (AAA) protocol originally developed by Livingston, Inc. RADIUS uses attribute value (AV) pairs to communicate information between the security server and the network access server. RFC 2138 and RFC 2139 describe the basic functionality of RADIUS and the original set of Internet Engineering Task Force (IETF)-standard AV pairs used to send AAA information. Two draft IETF standards, "RADIUS Attributes for Tunnel Protocol Support" and "RADIUS Accounting Modifications for Tunnel Protocol Support," extend the IETF-defined set of AV pairs to include attributes specific to virtual private networks (VPNs); these attributes are used to carry the tunneling information between the RADIUS server and the tunnel initiator. RFC 2865 and RFC 2868 extend the IETF-defined set of AV pairs to include attributes specific to compulsory tunneling in VPNs by allowing the user to specify authentication names for the network access server and the RADIUS server.

Cisco routers and access servers now support new RADIUS IETF-standard VPDN tunnel attributes. These new RADIUS IETF-standard attributes are listed in the "RADIUS Attributes" appendix. Refer to the following three configuration examples later in this chapter:

- RADIUS User Profile with RADIUS Tunneling Attributes Example
- L2TP Access Concentrator Examples
- L2TP Network Server Examples

For more information about L2F, L2TP, VPN, or VPDN, refer to the *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2.

RADIUS Configuration Examples

The following sections provide RADIUS configuration examples:

- RADIUS Authentication and Authorization Example
- RADIUS Authentication, Authorization, and Accounting Example
- Vendor-Proprietary RADIUS Configuration Example
- RADIUS Server with Server-Specific Values Example
- Multiple RADIUS Servers with Global and Server-Specific Values Example
- Multiple RADIUS Server Entries for the Same Server IP Address Example
- RADIUS Server Group Examples
- Multiple RADIUS Server Entries Using AAA Server Groups Example
- AAA Server Group Selection Based on DNIS Example
- AAA Preauthentication Examples

- RADIUS User Profile with RADIUS Tunneling Attributes Example
- Guard Timer Examples
- L2TP Access Concentrator Examples
- L2TP Network Server Examples

RADIUS Authentication and Authorization Example

The following example shows how to configure the router to authenticate and authorize using RADIUS:

```
aaa authentication login use-radius group radius local
aaa authentication ppp user-radius if-needed group radius
aaa authorization exec default group radius
aaa authorization network default group radius
```

The lines in this sample RADIUS authentication and authorization configuration are defined as follows:

- The **aaa authentication login use-radius group radius local** command configures the router to use RADIUS for authentication at the login prompt. If RADIUS returns an error, the user is authenticated using the local database. In this example, **use-radius** is the name of the method list, which specifies RADIUS and then local authentication.
- The **aaa authentication ppp user-radius if-needed group radius** command configures the Cisco IOS software to use RADIUS authentication for lines using PPP with CHAP or PAP if the user has not already been authorized. If the EXEC facility has authenticated the user, RADIUS authentication is not performed. In this example, **user-radius** is the name of the method list defining RADIUS as the if-needed authentication method.
- The **aaa authorization exec default group radius** command sets the RADIUS information that is used for EXEC authorization, autocommands, and access lists.
- The **aaa authorization network default group radius** command sets RADIUS for network authorization, address assignment, and access lists.

RADIUS Authentication, Authorization, and Accounting Example

The following example shows a general configuration using RADIUS with the AAA command set:

```
radius-server host 123.45.1.2
radius-server key myRaDiUSpassWoRd
username root password ALongPassword
aaa authentication ppp dialins group radius local
aaa authorization network default group radius local
aaa accounting network default start-stop group radius
aaa authentication login admins local
aaa authorization exec default local
line 1 16
 autoselect ppp
 autoselect during-login
 login authentication admins
 modem ri-is-cd
interface group-async 1
 encaps ppp
 ppp authentication pap dialins
```

The lines in this example RADIUS authentication, authorization, and accounting configuration are defined as follows:

- The **radius-server host** command defines the IP address of the RADIUS server host.
- The **radius-server key** command defines the shared secret text string between the network access server and the RADIUS server host.
- The **aaa authentication ppp dialins group radius local** command defines the authentication method list "dialins," which specifies that RADIUS authentication and then (if the RADIUS server does not respond) local authentication will be used on serial lines using PPP.
- The **ppp authentication pap dialins** command applies the "dialins" method list to the lines specified.
- The **aaa authorization network default group radius local** command is used to assign an address and other network parameters to the RADIUS user.
- The **aaa accounting network default start-stop group radius** command tracks PPP usage.
- The **aaa authentication login admins local** command defines another method list, "admins," for login authentication.
- The **login authentication admins** command applies the "admins" method list for login authentication.

Vendor-Proprietary RADIUS Configuration Example

The following example shows a general configuration using vendor-proprietary RADIUS with the AAA command set:

```
radius-server host alcatraz non-standard
radius-server key myRaDiUSpassWoRd
radius-server configure-nas
username root password ALongPassword
aaa authentication ppp dialins group radius local
aaa authorization network default group radius local
aaa accounting network default start-stop group radius
aaa authentication login admins local
aaa authorization exec default local
line 1 16
autoselect ppp
autoselect during-login
login authentication admins
modem ri-is-cd
interface group-async 1
encaps ppp
ppp authentication pap dialins
```

The lines in this example RADIUS authentication, authorization, and accounting configuration are defined as follows:

- The **radius-server host non-standard** command defines the name of the RADIUS server host and identifies that this RADIUS host uses a vendor-proprietary version of RADIUS.
- The **radius-server key** command defines the shared secret text string between the network access server and the RADIUS server host.
- The **radius-server configure-nas** command defines that the Cisco router or access server will query the RADIUS server for static routes and IP pool definitions when the device first starts up.

- The **aaa authentication ppp dialins group radius local** command defines the authentication method list "dialins," which specifies that RADIUS authentication, and then (if the RADIUS server does not respond) local authentication will be used on serial lines using PPP.
- The **ppp authentication pap dialins** command applies the "dialins" method list to the lines specified.
- The **aaa authorization network default group radius local** command is used to assign an address and other network parameters to the RADIUS user.
- The **aaa accounting network default start-stop group radius** command tracks PPP usage.
- The **aaa authentication login admins local** command defines another method list, "admins," for login authentication.
- The **login authentication admins** command applies the "admins" method list for login authentication.

RADIUS Server with Server-Specific Values Example

The following example shows how to configure server-specific timeout, retransmit, and key values for the RADIUS server with IP address 172.31.39.46:

```
radius-server host 172.31.39.46 timeout 6 retransmit 5 key rad123
```

Multiple RADIUS Servers with Global and Server-Specific Values Example

The following example shows how to configure two RADIUS servers with specific timeout, retransmit, and key values. In this example, the **aaa new-model** command enables AAA services on the router, while specific AAA commands define the AAA services. The **radius-server retransmit** command changes the global retransmission value to 4 for all RADIUS servers. The **radius-server host** command configures specific timeout, retransmission, and key values for the RADIUS server hosts with IP addresses 172.16.1.1 and 172.29.39.46.

```
! Enable AAA services on the router and define those services.
aaa new-model
aaa authentication login default group radius
aaa authentication login console-login none
aaa authentication ppp default group radius
aaa authorization network default group radius
aaa accounting exec default start-stop group radius
aaa accounting network default start-stop group radius
enable password tryit1
!
! Change the global retransmission value for all RADIUS servers.
radius-server retransmit 4
!
! Configure per-server specific timeout, retransmission, and key values.
! Change the default auth-port and acct-port values.
radius-server host 172.16.1.1 auth-port 1612 acct-port 1616 timeout 3 retransmit 3 key
radkey
!
! Configure per-server specific timeout and key values. This server uses the global
! retransmission value.
radius-server host 172.29.39.46 timeout 6 key rad123
```

Multiple RADIUS Server Entries for the Same Server IP Address Example

The following example shows how to configure the network access server to recognize several RADIUS host entries with the same IP address. Two different host entries on the same RADIUS server are configured for the same services—authentication and accounting. The second host entry configured acts as fail-over backup to the first one. (The RADIUS host entries will be tried in the order they are configured.)

```
! This command enables AAA.
aaa new-model
! The next command configures default RADIUS parameters.
aaa authentication ppp default group radius
! The next set of commands configures multiple host entries for the same IP address.
radius-server host 172.20.0.1 auth-port 1000 acct-port 1001
radius-server host 172.20.0.1 auth-port 2000 acct-port 2000
```

RADIUS Server Group Examples

The following example shows how to create server group *radgroup1* with three different RADIUS server members, each using the default authentication port (1645) and accounting port (1646):

```
aaa group server radius radgroup1
 server 172.16.1.11
 server 172.17.1.21
 server 172.18.1.31
```

The following example shows how to create server group *radgroup2* with three RADIUS server members, each with the same IP address but with unique authentication and accounting ports:

```
aaa group server radius radgroup2
 server 172.16.1.1 auth-port 1000 acct-port 1001
 server 172.16.1.1 auth-port 2000 acct-port 2001
 server 172.16.1.1 auth-port 3000 acct-port 3001
```

Multiple RADIUS Server Entries Using AAA Server Groups Example

The following example shows how to configure the network access server to recognize two different RADIUS server groups. One of these groups, group1, has two different host entries on the same RADIUS server configured for the same services. The second host entry configured acts as failover backup to the first one. Each group is individually configured for deadtime; deadtime for group 1 is one minute, and deadtime for group 2 is two minutes.

Note In cases where both global commands and server commands are used, the server command will take precedence over the global command.

```
! This command enables AAA.
aaa new-model
! The next command configures default RADIUS parameters.
aaa authentication ppp default group group1
! The following commands define the group1 RADIUS server group and associate servers
! with it and configures a deadtime of one minute.
aaa group server radius group1
 server 1.1.1.1 auth-port 1645 acct-port 1646
 server 2.2.2.2 auth-port 2000 acct-port 2001
 deadtime 1
```

```
! The following commands define the group2 RADIUS server group and associate servers
! with it and configures a deadtime of two minutes.
aaa group server radius group2
 server 2.2.2.2 auth-port 2000 acct-port 2001
 server 3.3.3.3 auth-port 1645 acct-port 1646
 deadtime 2
! The following set of commands configures the RADIUS attributes for each host entry
! associated with one of the defined server groups.
radius-server host 1.1.1.1 auth-port 1645 acct-port 1646
radius-server host 2.2.2.2 auth-port 2000 acct-port 2001
radius-server host 3.3.3.3 auth-port 1645 acct-port 1646
```

AAA Server Group Selection Based on DNIS Example

The following example shows how to select RADIUS server groups based on DNIS to provide specific AAA services:

```
! This command enables AAA.
aaa new-model
!
! The following set of commands configures the RADIUS attributes for each server
! that will be associated with one of the defined server groups.
radius-server host 172.16.0.1 auth-port 1645 acct-port 1646 key cisco1
radius-server host 172.17.0.1 auth-port 1645 acct-port 1646 key cisco2
radius-server host 172.18.0.1 auth-port 1645 acct-port 1646 key cisco3
radius-server host 172.19.0.1 auth-port 1645 acct-port 1646 key cisco4
radius-server host 172.20.0.1 auth-port 1645 acct-port 1646 key cisco5

! The following commands define the sg1 RADIUS server group and associate servers
! with it.
aaa group server radius sg1
  server 172.16.0.1
  server 172.17.0.1
! The following commands define the sg2 RADIUS server group and associate a server
! with it.
aaa group server radius sg2
  server 172.18.0.1
! The following commands define the sg3 RADIUS server group and associate a server
! with it.
aaa group server radius sg3
  server 172.19.0.1
! The following commands define the default-group RADIUS server group and associate
! a server with it.
aaa group server radius default-group
  server 172.20.0.1
!
! The next set of commands configures default-group RADIUS server group parameters.
aaa authentication ppp default group default-group
aaa accounting network default start-stop group default-group
!
```

```
! The next set of commands enables DNIS mapping and maps DNIS numbers to the defined
! RADIUS server groups. In this configuration, all PPP connection requests using
! DNIS 7777 are sent to the sg1 server group. The accounting records for these
! connections (specifically, start-stop records) are handled by the sg2 server group.
! Calls with a DNIS of 8888 use server group sg3 for authentication and server group
! default-group for accounting. Calls with a DNIS of 9999 use server group
! default-group for authentication and server group sg3 for accounting records
! (stop records only). All other calls with DNIS other than the ones defined use the
! server group default-group for both authentication and stop-start accounting records.
aaa dnis map enable
aaa dnis map 7777 authentication ppp group sg1
aaa dnis map 7777 accounting network start-stop group sg2
aaa dnis map 8888 authentication ppp group sg3
aaa dnis map 9999 accounting network stop-only group sg3
```

AAA Preauthentication Examples

The following example shows a simple configuration that specifies that the DNIS number be used for preauthentication:

```
aaa preauth
 group radius
 dnis required
```

The following example shows a configuration that specifies that both the DNIS number and the CLID number be used for preauthentication. DNIS preauthentication will be performed first, followed by CLID preauthentication.

```
aaa preauth
 group radius
 dnis required
 clid required
```

The following example specifies that preauthentication be performed on all DNIS numbers except the two DNIS numbers specified in the DNIS group called “hawaii”:

```
aaa preauth
 group radius
 dnis required
 dnis bypass hawaii

dialer dnis group hawaii
 number 12345
 number 12346
```

The following example shows a sample AAA configuration with DNIS preauthentication:

```
aaa new-model
aaa authentication login CONSOLE none
aaa authentication login RADIUS_LIST group radius
aaa authentication login TAC_PLUS group tacacs+ enable
aaa authentication login V.120 none
aaa authentication enable default enable group tacacs+
aaa authentication ppp RADIUS_LIST if-needed group radius
aaa authorization exec RADIUS_LIST group radius if-authenticated
aaa authorization exec V.120 none
aaa authorization network default group radius if-authenticated
aaa authorization network RADIUS_LIST if-authenticated group radius
aaa authorization network V.120 group radius if-authenticated
aaa accounting suppress null-username
aaa accounting exec default start-stop group radius
aaa accounting commands 0 default start-stop group radius
```

```
aaa accounting network default start-stop group radius
aaa accounting connection default start-stop group radius
aaa accounting system default start-stop group radius
aaa preauth
 dnis password Cisco-DNIS
aaa nas port extended
!
radius-server configure-nas
radius-server host 10.0.0.0 auth-port 1645 acct-port 1646 non-standard
radius-server host 10.255.255.255 auth-port 1645 acct-port 1646 non-standard
radius-server retransmit 2
radius-server deadtime 1
radius-server attribute nas-port format c
radius-server unique-ident 18
radius-server key MyKey
```

Note To configure preauthentication, you must also set up preauthentication profiles on the RADIUS server.

RADIUS User Profile with RADIUS Tunneling Attributes Example

The following example shows a RADIUS user profile (Merit Daemon format) that includes RADIUS tunneling attributes. This entry supports two tunnels, one for L2F and the other for L2TP. The tag entries with :1 support L2F tunnels, and the tag entries with :2 support L2TP tunnels.

```
cisco.com Password = "cisco", Service-Type = Outbound
    Service-Type = Outbound,
    Tunnel-Type = :1:L2F,
    Tunnel-Medium-Type = :1:IP,
    Tunnel-Client-Endpoint = :1:"10.0.0.2",
    Tunnel-Server-Endpoint = :1:"10.0.0.3",
    Tunnel-Client-Auth-Id = :1:"l2f-cli-auth-id",
    Tunnel-Server-Auth-Id = :1:"l2f-svr-auth-id",
    Tunnel-Assignment-Id = :1:"l2f-assignment-id",
    Cisco-Avpair = "vpdn:nas-password=l2f-cli-pass",
    Cisco-Avpair = "vpdn:gw-password=l2f-svr-pass",
    Tunnel-Preference = :1:1,
    Tunnel-Type = :2:L2TP,
    Tunnel-Medium-Type = :2:IP,
    Tunnel-Client-Endpoint = :2:"10.0.0.2",
    Tunnel-Server-Endpoint = :2:"10.0.0.3",
    Tunnel-Client-Auth-Id = :2:"l2tp-cli-auth-id",
    Tunnel-Server-Auth-Id = :2:"l2tp-svr-auth-id",
    Tunnel-Assignment-Id = :2:"l2tp-assignment-id",
    Cisco-Avpair = "vpdn:l2tp-tunnel-password=l2tp-tnl-pass",
    Tunnel-Preference = :2:2
```

Guard Timer Examples

The following example shows an ISDN guard timer that is set at 8000 milliseconds. A call will be rejected if the RADIUS server has not responded to a preauthentication request when the timer expires.

```
interface serial1/0/0:23
 isdn guard-timer 8000 on-expiry reject

aaa preauth
 group radius
 dnis required
```

The following example shows a CAS guard timer that is set at 20,000 milliseconds. A call will be accepted if the RADIUS server has not responded to a preauthentication request when the timer expires.

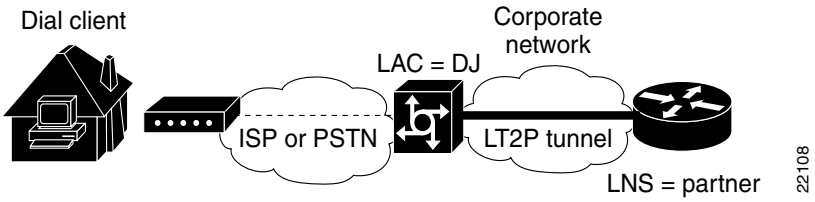
```
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 ds0-group 0 timeslots 1-24 type e&m-fgb dtmf dnis
 cas-custom 0
 call guard-timer 20000 on-expiry accept

aaa preauth
 group radius
 dnis required
```

L2TP Access Concentrator Examples

The following example shows a basic L2TP configuration for the L2TP access concentrator (LAC) for the topology shown in Figure 12. The local name is not defined, so the host name used is the local name. Because the L2TP tunnel password is not defined, the username password is used. In this example, VPDN is configured locally on the LAC and does not take advantage of the new RADIUS tunnel attributes.

Figure 12 Topology for Configuration Examples




```
! Enable AAA globally.
aaa new-model
! Enable AAA authentication for PPP and list the default method to use for PPP
! authentication.
aaa authentication ppp default local
! Define the username as "DJ."
username DJ password 7 030C5E070A00781B
! Enable VPDN.
vpdn enable
! Define VPDN group number 1.
vpdn-group 1
! Allow the LAC to respond to dialin requests using L2TP from IP address 172.21.9.13
! domain "cisco.com."
 request dialin
  protocol l2tp
  domain cisco.com
 initiate-ip to 172.21.9.13
 local name nas-1
```

The following example shows how to configure the LAC if RADIUS tunnel attributes are supported. In this example, there is no local VPDN configuration on the LAC; the LAC, instead, is configured to query the remote RADIUS security server.

```
! Enable global AAA securities services.
aaa new-model
```

```
! Enable AAA authentication for PPP and list RADIUS as the default method to use
! for PPP authentication.
aaa authentication ppp default group radius local
! Enable AAA (network) authorization and list RADIUS as the default method to use for
! authorization.
aaa authorization network default group radius
! Define the username as "DJ."
username DJ password 7 030C5E070A00781B
! Enable VPDN.
vpdn enable
! Configure the LAC to interface with the remote RADIUS security server.
radius host 171.69.1.1 auth-port 1645 acct-port 1646
radius-server key cisco
```

L2TP Network Server Examples

The following example shows a basic L2TP configuration with corresponding comments on the L2TP network server (LNS) for the topology shown in Figure 12:

```
! Enable AAA globally.
aaa new-model
! Enable AAA authentication for PPP and list the default method to use for PPP
! authentication.
aaa authentication ppp default local
! Define the username as "partner."
username partner password 7 030C5E070A00781B
! Create virtual-template 1 and assign all values for virtual access interfaces.
interface Virtual-Template1
! Borrow the IP address from interface ethernet 1.
 ip unnumbered Ethernet0
! Disable multicast fast switching.
 no ip mroute-cache
! Use CHAP to authenticate PPP.
 ppp authentication chap
! Enable VPDN.
vpdn enable
! Create vpdn-group number 1.
vpdn-group 1
! Accept all dialin l2tp tunnels from virtual-template 1 from remote peer DJ.
 accept dialin l2tp virtual-template 1 remote DJ
  protocol any
   virtual-template 1
  terminate-from hostname nas1
local name hgw1
```

The following example shows how to configure the LNS with a basic L2F and L2TP configuration using RADIUS tunneling attributes:

```
aaa new-model
aaa authentication login default none
aaa authentication login console none
aaa authentication ppp default local group radius
aaa authorization network default group radius if-authenticated
!
username l2f-cli-auth-id password 0 l2f-cli-pass
username l2f-svr-auth-id password 0 l2f-svr-pass
username l2tp-svr-auth-id password 0 l2tp-tnl-pass
!
vpdn enable
vpdn search-order domain
!
vpdn-group 1
```

```
accept-dialin
protocol l2f
virtual-template 1
terminate-from hostname l2f-cli-auth-id
local name l2f-svr-auth-id
!
vpdn-group 2
accept-dialin
protocol l2tp
virtual-template 2
terminate-from hostname l2tp-cli-auth-id
local name l2tp-svr-auth-id
!
interface Ethernet1/0
ip address 10.0.0.3 255.255.255.0
no ip route-cache
no ip mroute-cache
!
interface Virtual-Template1
ip unnumbered Ethernet1/0
ppp authentication pap
!
interface Virtual-Template2
ip unnumbered Ethernet1/0
ppp authentication pap
!
radius-server host 1.1.1.1 auth-port 1645 acct-port 1646
radius-server key <deleted>
!
```

AAA Dead-Server Detection

The AAA Dead-Server Detection feature allows you to configure the criteria that are to be used to mark a RADIUS server as dead. If no criteria are explicitly configured, the criteria will be computed dynamically on the basis of the number of outstanding transactions. Using this feature will result in less deadtime and quicker packet processing.

Feature History for AAA Dead-Server Detection

Release	Modification
12.3(6)	This feature was introduced.
12.3(7)T	This feature was integrated into Cisco IOS Release 12.3(7)T.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for AAA Dead-Server Detection, page 187
- Restrictions for AAA Dead-Server Detection, page 188
- Information About AAA Dead-Server Detection, page 188
- How to Configure AAA Dead-Server Detection, page 188
- Configuration Examples for AAA Dead-Server Detection, page 190
- Additional References, page 192
- Command Reference, page 193

Prerequisites for AAA Dead-Server Detection

- You must have access to a RADIUS server.
- You should be familiar with configuring a RADIUS server.
- You should be familiar with configuring authentication, authorization, and accounting (AAA).

- Before a server can be marked as dead, you must first configure the **radius-server deadtime** command. If this command is not configured, even if the criteria are met for the server to be marked as dead, the server state will be the "up" state.

Restrictions for AAA Dead-Server Detection

- Original transmissions are not counted in the number of consecutive timeouts that must occur on the router before the server is marked as dead—only the number of retransmissions are counted.

Information About AAA Dead-Server Detection

To configure the AAA Dead-Server Detection feature, you should understand the following concept:

- Criteria for Marking a RADIUS Server As Dead, page 188

Criteria for Marking a RADIUS Server As Dead

The AAA Dead-Server Detection feature allows you to determine the criteria that are used to mark a RADIUS server as dead. That is, you can configure the minimum amount of time, in seconds, that must elapse from the time that the router last received a valid packet from the RADIUS server to the time the server is marked as dead. If a packet has not been received since the router booted, and there is a timeout, the time criterion will be treated as though it has been met.

In addition, you can configure the number of consecutive timeouts that must occur on the router before the RADIUS server is marked as dead. If the server performs both authentication and accounting, both types of packets are included in the number. Improperly constructed packets are counted as though they are timeouts. Only retransmissions are counted, not the initial transmission. (Each timeout causes one retransmission to be sent.)

Note Both the time criterion and the tries criterion must be met for the server to be marked as dead.

The RADIUS dead-server detection configuration will result in the prompt detection of RADIUS servers that have stopped responding. This configuration will also result in the avoidance of servers being improperly marked as dead when they are "swamped" (responding slowly) and the avoidance of the state of servers being rapidly changed from dead to live to dead again. This prompt detection of nonresponding RADIUS servers and the avoidance of swamped and dead-to-live-to-dead-again servers will result in less deadtime and quicker packet processing.

How to Configure AAA Dead-Server Detection

This section contains the following procedures:

- Configuring AAA Dead-Server Detection, page 189 (required)
- Verifying AAA Dead-Server Detection, page 190 (optional)

Configuring AAA Dead-Server Detection

To configure AAA dead-server detection, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **radius-server deadtime** *minutes*
5. **radius-server dead-criteria** [**time** *seconds*] [**tries** *number-of-tries*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router (config)# aaa new-model`	Enables the AAA access control model.
Step 4	**radius-server deadtime** *minutes* **Example:** `Router (config)# radius-server deadtime 5`	Improves RADIUS response times when some servers might be unavailable and causes the unavailable servers to be skipped immediately.
Step 5	**radius-server dead-criteria** [**time** *seconds*] [**tries** *number-of-tries*] **Example:** `Router (config)# radius-server dead-criteria time 5 tries 4`	Forces one or both of the criteria—used to mark a RADIUS servr as dead—to be the indicated constant.

Troubleshooting Tips

After you have configured AAA dead-server detection, you should verify your configuration using the **show running-config** command. This verification is especially important if you have used the **no** form of the **radius-server dead-criteria** command. The output of the **show running-config** command must show the same values in the "Dead Criteria Details" field that you configured using the **radius-server dead-criteria** command.

Verifying AAA Dead-Server Detection

To verify your AAA dead-server detection configuration, perform the following steps. The **show** and **debug** commands may be used in any order.

SUMMARY STEPS

1. **enable**
2. **debug aaa dead-criteria transactions**
3. **show aaa dead-criteria**
4. **show aaa servers**
5. **show aaa server-private**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug aaa dead-criteria transactions** **Example:** `Router# debug aaa dead-criteria transactions`	Displays AAA dead-criteria transaction values.
Step 3	**show aaa dead-criteria** **Example:** `Router# show aaa dead-criteria`	Displays dead-criteria information for a AAA server.
Step 4	**show aaa servers** **Example:** `Router# show aaa servers`	Displays information about the AAA servers.
Step 5	**show aaa server-private** **Example:** `Router# show aaa server-private`	Displays the status of all private RADIUS servers.

Configuration Examples for AAA Dead-Server Detection

This section provides the following configuration examples:

- Configuring AAA Dead-Server Detection: Example, page 191
- debug aaa dead-criteria transactions Command: Example, page 191
- show aaa dead-criteria Command: Example, page 191

Configuring AAA Dead-Server Detection: Example

The following example shows that the router will be considered dead after 5 seconds and four tries:

```
Router (config)# aaa new-model
Router (config)# radius-server deadtime 5
Router (config)# radius-server dead-criteria time 5 tries 4
```

debug aaa dead-criteria transactions Command: Example

The following output example shows dead-criteria transaction information for a particular server group:

```
Router# debug aaa dead-criteria transactions

AAA Transaction debugs debugging is on

*Nov 14 23:44:17.403: AAA/SG/TRANSAC: Computed Retransmit Tries: 22, Current Max Tries: 22
*Nov 14 23:44:17.403: AAA/SG/TRANSAC: Computed Dead Detect Interval: 25s, Current Max
Interval: 25s
*Nov 14 23:44:17.403: AAA/SG/TRANSAC: Estimated Outstanding Transactions: 6, Current Max
Transactions: 6
```

show aaa dead-criteria Command: Example

The following output example shows that dead-server-detection information has been requested for a RADIUS server at the IP address 172.19.192.80:

```
Router# show aaa dead-criteria radius 172.19.192.80 radius

RADIUS Server Dead Critieria:
============================
Server Details:
    Address : 172.19.192.80
    Auth Port : 1645
    Acct Port : 1646
Server Group : radius
Dead Criteria Details:
    Configured Retransmits : 62
    Configured Timeout : 27
    Estimated Outstanding Transactions: 5
    Dead Detect Time : 25s
    Computed Retransmit Tries: 22
    Statistics Gathered Since Last Successful Transaction
=====================================================
Max Computed Outstanding Transactions: 5
Max Computed Dead Detect Time: 25s
Max Computed Retransmits : 22
```

Additional References

The following sections provide references related to AAA Dead-Server Detection.

Related Documents

Related Topic	Document Title
Configuring RADIUS	"Configuring RADIUS" chapter of *Cisco IOS Security Configuration Guide*
Configuring AAA	"Authentication, Authorization, and Accounting (AAA)" section of *Cisco IOS Security Configuration Guide*
Security commands	*Cisco IOS Security Commands*, Release 12.3 T

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **debug aaa dead-criteria transactions**
- **radius-server dead-criteria**
- **show aaa dead-criteria**
- **show aaa server-private**
- **show aaa servers**

ACL Default Direction

Feature History

Release	Modification
12.2(4)T	This feature was introduced.

This feature module describes the ACL Default Direction feature for Cisco IOS Release 12.2(4)T. It includes the following sections:

- Feature Overview, page 195
- Supported Platforms, page 196
- Supported Standards, MIBs, and RFCs, page 196
- Prerequisites, page 197
- Configuration Tasks, page 197
- Configuration Examples, page 198
- Command Reference, page 198

Feature Overview

The ACL Default Direction feature allows you to change the filter direction (where filter direction is not specified) to inbound packets only; that is, you can configure your server to filter packets that are coming toward the network.

This feature introduces the **radius-server attribute 11 direction default** command, which allows you to change the default direction of filters for your access control lists (ACL) via RADIUS. (RADIUS attribute 11 (Filter-Id) indicates the name of the filter list for the user.) Enabling this command allows you to change the filter direction to inbound—which stops traffic from entering a router, and reduces resource consumption—rather than keeping the outbound default direction, which waits until the traffic is about to leave the network before filtering occurs.

Benefits

The ACL Default Direction feature allows you to change the default direction, which is outbound, of filters for your access control lists to inbound via the **radius-server attribute 11 direction default** command.

Related Documents

- *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2
- *Cisco IOS Security Configuration Guide*, Release 12.2
- *Cisco IOS Security Command Reference*, Release 12.2
- "Configuring IP Services" section of the chapter "IP Addressing and Services" of the *Cisco IOS IP Configuration Guide*, Release 12.2
- RFC 2865, *Remote Authentication Dial-In User Service (RADIUS)*

Supported Platforms

- Cisco 7200 series

Platform Support Through Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Feature Navigator. Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image.

To access Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Feature Navigator is updated when major Cisco IOS software releases and technology releases occur. As of May 2001, Feature Navigator supports M, T, E, S, and ST releases. You can access Feature Navigator at the following URL:

http://www.cisco.com/go/fn

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

No new or modified RFCs are supported by this feature.

Prerequisites

Before you can change the default direction of filters from RADIUS, you must perform the following tasks:

- Configure your network access server (NAS) for authentication, authorization, and accounting (AAA) and to accept incoming calls.

  For more information, refer to the AAA chapters of the *Cisco IOS Security Configuration Guide*, Release 12.2 and the *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2.

- Create a filter on your NAS.

  For more information, refer to the section "Configuring IP Services" of the chapter "IP Addressing and Services" of the *Cisco IOS IP Configuration Guide*, Release 12.2

- Add a filter definition for a RADIUS user; for example, Filter-Id = "myfilter".

Configuration Tasks

See the following sections for configuration tasks for the ACL Default Direction feature. Each task in the list is identified as either required or optional.

- Configuring RADIUS Attribute 11 (Filter-Id) (required)
- Verifying RADIUS Attribute 11 (Filter-Id) (optional)

Configuring RADIUS Attribute 11 (Filter-Id)

To configure the default direction of filters from RADIUS via attribute 11, use the following command in global configuration mode:

Command	Purpose
`Router#(config) radius-server attribute 11 default direction [inbound | outbound]`	Specifies the default direction of filters from RADIUS to inbound or outbound.

Verifying RADIUS Attribute 11 (Filter-Id)

To verify the default direction of filters from RADIUS and to verify that RADIUS attribute 11 is being sent in access accept requests, use the following privileged EXEC commands:

Command	Purpose
`Router# more system:running-config`	Displays the contents of the current running configuration file. (Note that the **more system:running-config** command has replaced the **show running-config** command.)
`Router# debug radius`	Displays information associated with RADIUS. The output of this command shows whether attribute 11 is being sent in access accept requests.

Configuration Examples

This section provides the following configuration examples:

- Default Direction of Filters via RADIUS Attribute 11 (Filter-Id) Configuration Example
- RADIUS User Profile with Filter-Id Example

Default Direction of Filters via RADIUS Attribute 11 (Filter-Id) Configuration Example

The following example shows how to configure RADIUS attribute 11 to change the default direction of filters. In this example, the filtering is applied to inbound packets only.

```
radius-server attribute 11 direction default inbound
```

RADIUS User Profile with Filter-Id Example

The following is an example of a RADIUS user profile (Merit Daemon format) that includes RADIUS attribute 11 (Filter-Id):

```
client Password = "cisco"
        Service-Type = Framed,
        Framed-Protocol = PPP,
        Filter-Id = "myfilter.out"
```

The RADIUS user profile shown in this example produces the following reply from the NAS:

```
RADIUS: Send to unknown id 79 10.51.13.4:1645, Access-Request, len 85
RADIUS: authenticator 84 D3 B5 7D C2 5B 70 AD - 1E 5C 56 E8 3A 91 D0 6E
RADIUS: User-Name           [1]   8  "client"
RADIUS: CHAP-Password       [3]   19  *
RADIUS: NAS-Port            [5]   6   20030
RADIUS: NAS-Port-Type       [61]  6   ISDN                      [2]
RADIUS: Called-Station-Id   [30]  6   "4321"
RADIUS: Calling-Station-Id  [31]  6   "1234"
RADIUS: Service-Type        [6]   6   Framed                    [2]
RADIUS: NAS-IP-Address      [4]   6   9.1.73.74

RADIUS: Received from id 79 10.51.13.4:1645, Access-Accept, len 46
RADIUS: authenticator 9C 6C 66 E2 F1 42 D6 4B - C1 7D D4 5E 9D 09 BB A1
RADIUS: Service-Type        [6]   6   Framed                    [2]
RADIUS: Framed-Protocol     [7]   6   PPP                       [1]
RADIUS: Filter-Id           [11]  14
RADIUS: 6D 79 66 69 6C 74 65 72 2E 6F 75 74             [myfilter.out]
```

Command Reference

The following new command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **radius-server attribute 11 direction default**

Attribute Screening for Access Requests

The Attribute Screening for Access Requests feature allows you to configure your network access server (NAS) to filter attributes in outbound Access Requests to the RADIUS server for purposes of authentication or authorization.

Feature History for Attribute Screening for Access Requests

Release	Modification
12.3(3)B	This feature was introduced.
12.3(7)T	This feature was integrated into Cisco IOS Release 12.3(7)T.
12.2(27)SB	This feature was integrated into Cisco IOS Release 12.2(27)SB.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for Attribute Screening for Access Requests, page 200
- Restrictions for Attribute Screening for Access Requests, page 200
- Information About Attribute Screening for Access Requests, page 200
- How to Configure Attribute Screening for Access Requests, page 200
- Configuration Examples for Attribute Filtering for Access Requests, page 204
- Additional References, page 205
- Command Reference, page 206

Prerequisites for Attribute Screening for Access Requests

- You must be familiar with configuring attribute lists.

Restrictions for Attribute Screening for Access Requests

- Attributes 1 (Username), 2 (User-Password), and 3 (Chap-Password) cannot be filtered.

Information About Attribute Screening for Access Requests

To configure the Attribute Screening for Access Requests feature, you should understand the following concept:

- Configuring an NAS to Filter Attributes in Outbound Access Requests, page 200

Configuring an NAS to Filter Attributes in Outbound Access Requests

The Attribute Screening for Access Requests feature allows you to configure your NAS to filter attributes in outbound Access Requests to the RADIUS server for purposes of authentication or authorization. The filters can be configured on the NAS, or they can be downloaded via downloadable vendor-specific attributes (VSAs) from the authentication, authorization, and accounting (AAA) server.

The following are some examples of the downloadable VSAs:

```
Cisco:Cisco-Avpair="ppp-authen-type=chap"
Cisco:Cisco-Avpair="ppp-authen-list=group 1"
Cisco:Cisco-Avpair="ppp-author-list=group 1"
Cisco:Cisco-Avpair="vpdn:tunnel-id=B53"
Cisco:Cisco-Avpair="vpdn:ip-addresses=10.0.58.35"
```

Note You must be aware of which attributes you want to filter. Filtering certain key attributes can result in authentication failure (for example, attribute 60 should not be filtered).

How to Configure Attribute Screening for Access Requests

This section contains the following procedures:

- Configuring Attribute Screening for Access Requests, page 201
- Configuring a Router to Support Downloadable Filters, page 202
- Monitoring and Maintaining Attribute Filtering for Access Requests, page 203

Configuring Attribute Screening for Access Requests

To configure attribute screening for Access Requests, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius-server attribute list** *listname*
4. **attribute** *value1* [*value2* [*value3*...]]
5. **aaa group server radius** *group-name*
6. **authorization** [**request** | **reply**] [**accept** | ***reject***] *listname*

   or

   accounting [**request** | **reply**] [**accept** | **reject**] *listname*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**radius-server attribute list** *listname* **Example:** `Router (config)# radius-server attribute list attrlist`	Defines an attribute list.
Step 4	**attribute** *value1* [*value2*[*value3*...]] **Example:** `Router (config)# attribute 6-10, 12`	Adds attributes to an accept or reject list.
Step 5	**aaa group server radius** *group-name* **Example:** `Router (config)# aaa group server radius rad1`	Applies the attribute list to the AAA server group and enters server-group configuration mode.

	Command or Action	Purpose
Step 6	**authorization** [**request** | **reply**][**accept** | **reject**] *listname* or **accounting** [**request** | **reply**] [**accept** | **reject**] *listname* **Example:** `Router (config-sg-radius)# authorization request accept attrlist` or **Example:** `Router (config-sg-radius)# accounting request accept attrlist`	Filters attributes in outbound Access Requests to the RADIUS server for purposes of authentication or authorization. • The **request** keyword defines filters for outgoing authorization Access Requests. • The **reply** keyword defines filters for incoming authorization Accept and Reject packets and for outgoing accounting requests.

Configuring a Router to Support Downloadable Filters

To configure your router to support downloadable filters, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa authorization template**
4. **aaa authorization network default group radius**
5. **radius-server attribute list** *list-name*
6. **attribute** *value1* [*value2* [*value3*...]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**aaa authorization template** **Example:** `Router (config)# aaa authorization template`	Enables usage of a local or remote customer template on the basis of Virtual Private Network (VPN) routing and forwarding (VRF).
Step 4	**aaa authorization network default group radius** **Example:** `Router (config)# aaa authorization network default group radius`	Sets parameters that restrict user access to a network.
Step 5	**radius-server attribute list** *list-name* **Example:** `Router (config)# radius-server attribute list attlist`	Defines an accept or reject list name.
Step 6	**attribute** *value1* [*value2* [*value3...*]] **Example:** `Router (config)# attribute 10-14, 24`	Adds attributes to an accept or reject list.

Troubleshooting Tips

If attribute filtering is not working, ensure that the attribute list is properly defined.

Monitoring and Maintaining Attribute Filtering for Access Requests

To monitor and maintain attribute filtering, you can use the **debug radius** command.

SUMMARY STEPS

1. **enable**
2. **debug radius**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug radius** **Example:** `Router# debug radius`	Displays information associated with RADIUS, including filtering information.

Configuration Examples for Attribute Filtering for Access Requests

This section provides the following configuration examples:

- Attribute Filtering for Access Requests: Example, page 204
- Attribute Filtering User Profile: Example, page 204
- debug radius Command: Example, page 205

Attribute Filtering for Access Requests: Example

The following example shows that the attributes 30-31 that are defined in “all-attr” will be rejected in all outbound Access Request messages:

```
aaa group server radius ras
 server 172.19.192.238 auth-port 1745 acct-port 1746
 authorization request reject all-attr
!
.
.
.
radius-server attribute list all-attr
 attribute 30-31
!
.
.
.
```

Attribute Filtering User Profile: Example

The following is a sample user profile after attribute filtering has been configured for Access Requests:

```
cisco.com Password = "cisco"
Service-Type = Framed,
Framed-Protocol = PPP,
Cisco:Cisco-Avpair = :1:"rad-serv=172.19.192.87 key rad123",
Cisco:Cisco-Avpair = :1:"rad-serv-filter=authorization request reject range1",
Cisco:Cisco-Avpair = :1:"rad-serv-filter=accounting request reject range1",
Cisco:Cisco-Avpair = "ppp-authen-type=chap"
Cisco:Cisco-Avpair = "ppp-authen-list=group 1",
Cisco:Cisco-Avpair = "ppp-author-list=group 1",
Cisco:Cisco-Avpair = "ppp-acct-list=start-stop group 1",
Cisco:Cisco-Avpair = "vpdn:tunnel-id=B53",
Cisco:Cisco-Avpair = "vpdn:tunnel-type=l2tp",
Cisco:Cisco-Avpair = "vpdn:ip-addresses=10.0.58.35",
Cisco:Cisco-Avpair = "vpdn:l2tp-tunnel-password=cisco"


user2@cisco.com
Service-Type = Outbound,
Cisco:Cisco-Avpair = "vpdn:tunnel-id=B53",
Cisco:Cisco-Avpair = "vpdn:tunnel-type=l2tp",
Cisco:Cisco-Avpair = "vpdn:ip-addresses=10.0.58.35",
Cisco:Cisco-Avpair = "vpdn:l2tp-tunnel-password=cisco"
```

When a session for user2@cisco.com "comes up" at the Layer 2 Tunneling Protocol (L2TP) Network Server (LNS)—as is shown above—because the **aaa authorization template** command has been configured, a RADIUS request is sent to the server for Cisco.com. The server then sends an Access Accept message if authentication is successful, along with the VSAs that are configured as part of the Cisco.com profile. If filters are configured as part of the Cisco.com profile, these filters will be parsed and applied to the RADIUS requests for user2@cisco.com.

In the above profile example, filter range1 has been applied to the authorization and accounting requests.

debug radius Command: Example

If the attribute you are trying to filter is rejected, you will see an **debug radius** output statement similar to the following:

```
RADIUS: attribute 31 rejected
```

If you try to filter an attribute that cannot be filtered, you will see an output statement similar to the following:

```
RADIUS: attribute 1 cannot be rejected
```

Additional References

The following sections provide references related to Attribute Filtering for Access Requests.

Related Documents

Related Topic	Document Title
Authentication, authorization, and accounting (AAA)	"Authentication, Authorization, and Accounting (AAA)" section of the *Cisco IOS Security Configuration Guide*
Configuring RADIUS	"Configuring RADIUS" chapter of the *Cisco IOS Security Configuration Guide*.
Security commands	*Cisco IOS Security Command Reference*, Release 12.3 T.
RADIUS attribute lists	*RADIUS Attribute Screening*

Standards

Standards	Title
No new or modified standards are supported by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following modified command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **authorization (server-group)**

Enable Multilink PPP via RADIUS for Preauthentication User

Feature History

Release	Modification
12.2(11)T	This feature was introduced.
12.2(27)SB	This feature was integrated into Cisco IOS Release 12.2(27)SB.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

This feature module describes the Enable Multilink PPP via RADIUS for Preauthentication User feature in Cisco IOS Release 12.2(11)T. It includes the following sections:

- Feature Overview, page 207
- Supported Platforms, page 209
- Supported Standards, MIBs, and RFCs, page 210
- Prerequisites, page 210
- Configuration Tasks, page 210
- Configuration Examples, page 211
- Command Reference, page 212
- Glossary, page 213

Feature Overview

The Enable Multilink PPP via RADIUS for Preauthentication User feature allows you to selectively enable and disable Multilink PPP (MLP) negotiation for different users via RADIUS vendor-specific attribute (VSA) preauth:ppp-multilink=1.

You can enable MLP by configuring the **ppp multilink** command on an interface, but then this command enables MLP negotiation for all connections and users on that interface; that is, you cannot selectively enable or disable MLP negotiation for specific connections and users on an interface.

Note To enable this feature, the **ppp multilink** command should not be configured on the interface; this command will disable MLP by default. If the **ppp multilink** command is already configured on the interface, the attribute "preauth:ppp-multilink=1" will not override this command.

How MLP via RADIUS Works

Because MLP parameters are negotiated at the time of link control protocol (LCP) negotiation, RADIUS VSA preauth:ppp-multilink=1 should only be a part of preauthentication user authorization. You should add this VSA to the preauthentication profile of the user to enable MLP. Thus, MLP will be enabled only for preauthentication users whose profiles contain this VSA; MLP will be disabled for all other users. If the MLP VSA is received during PPP user authorization (as opposed to preauthentication user authorization), it will be too late to negotiate MLP, and MLP will not be enabled.

When this VSA is received during preauthentication user authorization, MLP negotiation for the user is enabled. MLP is enabled when the VSA value is 1. All attribute values other than 1 are ignored.

Roles of the L2TP Access Server and L2TP Network Server

With this feature, you do not need to configure MLP on the interface of the L2TP access server (LAC); during preauthentication user authorization, the LAC will selectively choose to enable MLP for preauthentication users who receive preauth:ppp-multilink=1. On the L2TP network server (LNS), you can control the maximum number of links allowed in the multilink bundle by sending RADIUS VSA multilink:max-links=n during PPP user authorization.

New Vendor-Specific Attributes

This feature introduces the following new VSAs:

- Cisco-AVpair = preauth:ppp-multilink=1

  Turns on MLP on the interface and is applied to the preauthentication profile.

- Cisco-AVpair = multilink:max-links=n

  Restricts the maximum number of links that a user can have in a multilink bundle and is used with the service=ppp attribute. The range of "n" is from 1 to 255.

- Cisco-AVpair = multilink:min-links=1

  Sets the minimum number of links for MLP. The range of "n" is from 0 to 255.

- Cisco-AVpair = multilink:load-threshold=n

  Sets the load threshold for the caller for which additional links are added or deleted from the multilink bundle. If the load exceeds the specified value, links are added; if the load drops below the specified value, links are deleted. This attribute is used with the service=ppp attribute. The range of "n" is from 1 to 255.

Note RADIUS VSAs multilink:max-links, multilink:min-links, and multilink:load-threshold serve the same purpose as TACACS+ per-user attributes, max-links, min-links, and load-threshold respectively.

Benefits

Selective Multilink PPP Configuration

MLP negotiation can be selectively enabled and disabled for different users by applying RADIUS VSA preauth:ppp-multilink=1 to the preauthentication profile.

Related Documents

- "RADIUS Attributes" appendix in the C*isco IOS Security Configuration Guide*, Release 12.2
- "TACACS+ Attribute-Value Pairs" appendix in the C*isco IOS Security Configuration Guide*, Release 12.2
- "Configuring RADIUS" chapter in the *Cisco IOS Security Configuration Guide*, Release 12.2
- "PPP Configuration" chapter in the *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2
- *Cisco IOS Dial Technologies Command Reference*, Release 12.2

Supported Platforms

- Cisco AS5300 series
- Cisco AS5350 series
- Cisco AS5400 series
- Cisco AS5800 series
- Cisco AS5850 series

Determining Platform Support Through Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Feature Navigator. Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image.

To access Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

No new or modified RFCs are supported by this feature.

Prerequisites

Before enabling MLP via RADIUS VSA preauth:ppp-multilink=1, you should perform the following tasks:

- Enable the network access server (NAS) to recognize and use VSAs as defined by RADIUS IETF attribute 26 by using the **radius-server vsa send** command.

  For more information about using VSAs, refer to the section "Configuring Router to Use Vendor-Specific RADIUS Attributes" of the chapter "Configuring RADIUS" in the *Cisco IOS Security Configuration Guide*, Release 12.2.

- Enable preauthentication.

  For information about configuring preauthentication, refer to the section "Configuring AAA Preauthentication" of the chapter "Configuring RADIUS" in the *Cisco IOS Security Configuration Guide*, Release 12.2.

Configuration Tasks

None

Verifying MLP Negotiation via RADIUS in Preauthentication

To display bundle information for the MLP bundles, use the **show ppp multilink** EXEC command.

```
Router# show ppp multilink

Virtual-Access1, bundle name is mlpuser
  Bundle up for 00:00:15
  Dialer interface is Serial0:23
  0 lost fragments, 0 reordered, 0 unassigned
  0 discarded, 0 lost received, 1/255 load
  0x0 received sequence, 0x0 sent sequence
  Member links: 1 (max 7, min 1)
    Serial0:22, since 00:00:15, no frags rcvd
```

Table 15 describes the significant fields shown when MLP is enabled.

Table 15 show ppp multilink Field Descriptions

Field	Description
Virtual-Access1	Multilink bundle virtual interface.
Bundle	Configured name of the multilink bundle.
Dialer Interface is Serial0:23	Name of the interface that dials the calls.
1/255 load	Load on the link in the range 1/255 to 255/255. (255/255 is a 100% load.)
Member links: 1	Number of child interfaces.

Configuration Examples

This section provides dialin VPDN configurations using Cisco VSA ppp-multilink examples:

- LAC for MLP Configuration Example
- LAC RADIUS Profile for Preauthentication Example
- LNS for MLP Configuration Example
- LNS RADIUS Profile Example

LAC for MLP Configuration Example

The following example is a sample configuration that can be used to configure a LAC for MLP via RADIUS:

```
! Enable preauthentication
aaa preauth
  group radius
  dnis required

!Enable VPDN
vpdn enable
!
vpdn-group 1
  request-dialin
  protocol l2tp
  dnis 56118
  initiate-to ip 10.0.1.22
  local name lac-router

! Don't need to configure multilink on the interface
! Multilink will be enabled by "ppp-multilink" attribute
interface Serial0:23
  ip address 15.0.1.7 255.0.0.0
  encapsulation ppp
  dialer-group 1
  isdn switch-type primary-5ess
  isdn calling-number 56118
  peer default ip address pool pool1
  no cdp enable
  ppp authentication chap
```

LAC RADIUS Profile for Preauthentication Example

The following example shows a LAC RADIUS profile for a preauthentication user who has applied the preauth:ppp-multilink=1 VSA:

```
56118   Password = “cisco”
        Service-Type = Outbound,
        Framed-Protocol = PPP,
        Framed-MTU = 1500,
        Cisco-Avpair = "preauth:auth-required=1",
        Cisco-Avpair = "preauth:auth-type=chap",
        Cisco-Avpair = "preauth:username=dnis:56118",
        Cisco-Avpair = "preauth:ppp-multilink=1"
```

LNS for MLP Configuration Example

The following example is a sample configuration that can be used to configure a LNS to limit the number of links in a MLP bundle:

```
! Enable VPDN
vpdn enable
!
vpdn-group 1
 accept-dialin
 protocol any
 virtual-template 1
 terminate-from hostname lac-router
 local name lns-router
!
! Configure multilink on interface
interface Virtual-Template 1
 ip unnumbered Ethernet 0/0
 ppp authentication chap
 ppp multilink
```

LNS RADIUS Profile Example

The following example shows a LNS RADIUS profile for specifying the maximum number of links in a multilink bundle. The following multilink VSAs should be specified during PPP user authorization.

```
mascot  password = "cisco"
        Service-Type = Framed,
        Framed-Protocol = PPP,
        Cisco-Avpair = "multilink:max-links=7"
        Cisco-Avpair = "multilink:min-links=1"
        Cisco-Avpair = "multilink:load-threshold=128"
```

Command Reference

This feature uses no new or modified commands. To see the command pages for the commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

Glossary

AAA—authentication, authorization, and accounting. Suite of network security services that provide the primary framework through which access control can be set up on your Cisco router or access server.

attribute—RADIUS Internet Engineering Task Force (IETF) attributes are the original set of 255 standard attributes that are used to communicate AAA information between a client and a server. Because IETF attributes are standard, the attribute data is predefined and well known; thus all clients and servers that exchange AAA information via IETF attributes must agree on attribute data such as the exact meaning of the attributes and the general bounds of the values for each attribute.

L2F—Layer 2 Forwarding. Protocol that supports the creation of secure virtual private dial-up networks over the Internet.

L2TP—Layer 2 Tunnel Protocol. A Layer 2 tunneling protocol that enables an ISP or other access service to create a virtual tunnel to link customer remote sites or remote users with corporate home networks. In particular, a network access server (NAS) at the ISP point of presence (POP) exchanges PPP messages with the remote users and communicates by L2F or L2TP requests and responses with the customer tunnel server to set up tunnels.

LAC—L2TP access concentrator. A network access server (NAS) to which the client directly connects and through which PPP frames are tunneled to the L2TP network server (LNS). The LAC need only implement the media over which L2TP is to operate to pass traffic to one or more LNSs. The LAC may tunnel any protocol carried within PPP. The LAC initiates incoming calls and receives outgoing calls. A LAC is analogous to an L2F network access server.

LNS—L2TP network server. A termination point for L2TP tunnels, and an access point where PPP frames are processed and passed to higher-layer protocols. An LNS can operate on any platform that terminates PPP. The LNS handles the server side of the L2TP protocol. L2TP relies only on the single medium over which L2TP tunnels arrive. The LNS initiates outgoing calls and receives incoming calls. An LNS is analogous to a home gateway in L2F technology.

MLP—Multilink PPP. MLP allows packets to be fragmented and the fragments to be sent at the same time over multiple point-to-point links to the same remote address. The multiple links come up in response to a defined dialer load threshold. The load can be calculated on inbound traffic, outbound traffic, or on either, as needed for the traffic between the specific sites. MLP provides bandwidth on demand and reduces transmission latency across WAN links.

MLP is designed to work over synchronous and asynchronous serial and BRI and PRI types of single or multiple interfaces that have been configured to support both dial-on-demand rotary groups and PPP encapsulation.

RADIUS—Remote Authentication Dial-In User Service. RADIUS is a distributed client/server system that secures networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco routers and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.

VSA—Vendor-Specific Attribute. VSAs derived from one IETF attribute—vendor-specific (attribute 26). Attribute 26 allows a vendor to create and implement an additional 255 attributes. That is, a vendor can create an attribute that does not match the data of any IETF attribute and encapsulate it behind attribute 26: essentially, Vendor-Specific = “protocol:attribute=value.”

Enhanced Test Command

Feature History

Release	Modification
12.2(4)T	This feature was introduced.

This feature module describes the Enhanced Test Command feature in Cisco IOS Release 12.2(4)T. It includes the following sections:

- Feature Overview, page 215
- Supported Platforms, page 216
- Supported Standards, MIBs, and RFCs, page 217
- Configuration Tasks, page 217
- Configuration Examples, page 218
- Command Reference, page 219
- Glossary, page 220

Feature Overview

The Enhanced Test Command feature introduces two new commands—**aaa user profile** and **aaa attribute**—that allow you to create a named user profile with calling line identification (CLID) or dialed number identification service (DNIS) attribute values, which can be associated with a **test aaa group** command.

Use the **aaa attribute** command to add CLID or DNIS attribute values to a user profile, which is created by using the **aaa user profile** command. The CLID or DNIS attribute values can be associated with the record that is going out with the user profile (via the **test aaa group** command), thereby providing the RADIUS server with access to CLID or DNIS attribute information for all incoming calls.

Benefits

The Enhanced Test Command feature allows you to add a named user profile with CLID or DNIS attribute values and associate the user profile with the **test aaa group** command. Thus, the attribute values that are added to the user profile go to the RADIUS server, and the RADIUS server can access CLID or DNIS information when it receives a RADIUS record.

Restrictions

The **test aaa group** command does not work with TACACS+.

Related Documents

- *Cisco IOS Security Configuration Guide*, Release 12.2
- *Cisco IOS Security Command Reference*, Release 12.2

Supported Platforms

- Cisco 7100 series
- Cisco 7200 series
- Cisco 7500 series

Platform Support Through Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Feature Navigator. Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image.

To access Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Feature Navigator is updated when major Cisco IOS software releases and technology releases occur. As of May 2001, Feature Navigator supports M, T, E, S, and ST releases. You can access Feature Navigator at the following URL:

http://www.cisco.com/go/fn

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

None

Configuration Tasks

See the following sections for configuration tasks for the Enhanced Test Command feature. Each task in the list is identified as either required or optional.

- Configuring a User Profile (required)
- Associating a User Profile with a test aaa group Command (required)
- Verifying Enhanced Test Command (optional)

Configuring a User Profile

To create a named user profile and add CLID or DNIS attribute values, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa user profile profile-name`	Creates a user profile.
Step 2	`Router(config-aaa-user)# aaa attribute {dnis | clid} attribute-value`	Adds DNIS or CLID attribute values to the user profile and enters AAA-user configuration mode.

Associating a User Profile with a test aaa group Command

To associate a user profile with a **test aaa group** command, use the following command in privilege EXEC mode:

Command	Purpose
`Router# test aaa group {group-name | radius} username password new-code [profile profile-name]`	Associates a DNIS or CLID named user profile with the record that is sent to the RADIUS server. **Note** The *profile-name* must match the *profile-name* specified in the **aaa user profile** command.

Verifying Enhanced Test Command

To verify the enhanced test command configurations, use the following privilege EXEC commands:

Command	Purpose
`Router# debug radius`	Displays information associated with RADIUS.
`Router# more system:running-config`	Displays the contents of the current running configuration file. (Note that the **more system:running-config** command has replaced the **show running-config** command.)

Configuration Examples

This section provides the following configuration example:

- User Profile Associated With a test aaa group command Example

User Profile Associated With a test aaa group command Example

The following example shows how to configure the dnis = dnisvalue user profile “prfl1” and associate it with a **test aaa group** command. In this example, the **debug radius** command has been enabled and the output follows the configuration.

```
aaa user profile prfl1
  aaa attribute dnis
  aaa attribute dnis dnisvalue
  no aaa attribute clid
! Attribute not found.
  aaa attribute clid clidvalue
  no aaa attribute clid
  exit
!
! Associate the dnis user profile with the test aaa group command.
test aaa group radius user1 pass new-code profile prfl1
!
!
!
```

```
! debug radius output, which shows that the dnis value has been passed to the radius
! server.
*Dec 31 16:35:48: RADIUS: Sending packet for Unique id = 0
 *Dec 31 16:35:48: RADIUS: Initial Transmit unknown id 8 171.69.71.21:1645,
Access-Request, len 68
 *Dec 31 16:35:48: RADIUS: code=Access-Request id=08 len=0068
         authenticator=1E CA 13 F2 E2 81 57 4C - 02 EA AF 9D 30 D9 97 90
         T=User-Password[2]                L=12 V=*
         T=User-Name[1]                    L=07 V="kalki"
         T=Called-Station-Id[30]           L=0B V="dnisvalue"
         T=Service-Type[6]                 L=06 V=Login                    [1]
         T=NAS-IP-Address[4]               L=06 V=10.0.1.81

 *Dec 31 16:35:48: RADIUS: Received from id 8 171.69.71.21:1645, Access-Accept, len 38
 *Dec 31 16:35:48: RADIUS: code=Access-Accept id=08 len=0038
```

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **aaa attribute**
- **aaa user profile**
- **test aaa group**

Glossary

attribute—RADIUS Internet Engineering Task Force (IETF) attributes are the original set of 255 standard attributes that are used to communicate AAA information between a client and a server. Because IETF attributes are standard, the attribute data is predefined and well known; thus all clients and servers who exchange AAA information via IETF attributes must agree on attribute data such as the exact meaning of the attributes and the general bounds of the values for each attribute.

CLID—calling line identification. Also called caller ID. CLID provides the number from which a call originates.

DNIS—dialed number identification service. DNIS provides the number that is dialed.

Framed-Route in RADIUS Accounting

The Framed-Route in RADIUS Accounting feature provides for the presence of Framed-Route (RADIUS attribute 22) information in RADIUS Accounting-Request accounting records. The Framed-Route information will be returned to the RADIUS server in the Accounting-Request packets. The Framed-Route information can be used to verify that a per-user route or routes have been applied for a particular static IP customer on the network access server (NAS).

Feature History for Framed-Route in RADIUS Accounting

Release	Modification
12.3(4)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for Framed-Route in RADIUS Accounting, page 221
- Information About Framed-Route in RADIUS Accounting, page 222
- How to Monitor Framed-Route in RADIUS Accounting, page 222
- Additional References, page 224
- Command Reference, page 225

Prerequisites for Framed-Route in RADIUS Accounting

- You should be familiar with configuring authentication, authorization, and accounting (AAA).
- You should be familiar with RADIUS servers and configuring RADIUS attribute screening.

Information About Framed-Route in RADIUS Accounting

This section includes the following concepts:

- Framed-Route, Attribute 22, page 222
- Framed-Route in RADIUS Accounting Packets, page 222

Framed-Route, Attribute 22

Framed-Route, attribute 22 as defined in Internet Engineering Task Force (IETF) standard RFC 2865, provides for routing information to be configured for the user on the NAS. The Framed-Route attribute information is usually sent from the RADIUS server to the NAS in Access-Accept packets. The attribute can appear multiple times.

Framed-Route in RADIUS Accounting Packets

The Framed-Route attribute information in RADIUS accounting packets shows per-user routes that have been applied for a particular static IP customer on the NAS. The Framed-Route attribute information is currently sent in Access-Accept packets. Effective with Cisco IOS Release 12.3(4)T, the Framed-Route attribute information will also be sent in Accounting-Request packets if it was provided in the Access-Accept packets and was applied successfully. Zero or more instances of the Framed-Route attribute may be present in the Accounting-Request packets.

Note If there is more than one Framed-Route attribute in an Access-Accept packet, there can also be more than one Framed-Route attribute in the Accounting-Request packet.

The Framed-Route information will be returned in Stop and Interim accounting records and in Start accounting records when accounting Delay-Start is configured.

No configuration is required to have the Frame-Route attribute information returned in the RADIUS accounting packets.

How to Monitor Framed-Route in RADIUS Accounting

You can use the **debug radius** command to monitor whether Framed-Route (attribute 22) information is being sent in RADIUS Accounting-Request packets.

Examples

This section provides the following example:

- debug radius Command Output: Example, page 223

debug radius Command Output: Example

In the following example, the **debug radius** command is used to verify that Framed-Route (attribute 22) information is being sent in the Accounting-Request packets (see the line 00:06:23: RADIUS: Framed-Route [22] 26 "10.80.0.1 255.255.255.255 10.60.0.1 100").

```
Router# debug radius

00:06:23: RADIUS:  Send to unknown id 0 10.1.0.2:1645, Access-Request, len 126
00:06:23: RADIUS:  authenticator 40 28 A8 BC 76 D4 AA 88 - 5A E9 C5 55 0E 50 84 37
00:06:23: RADIUS:  Framed-Protocol     [7]   6   PPP                       [1]
00:06:23: RADIUS:  User-Name           [1]   14  "nari@trw1001"
00:06:23: RADIUS:  CHAP-Password       [3]   19  *
00:06:23: RADIUS:  NAS-Port            [5]   6   1
00:06:23: RADIUS:  Vendor, Cisco       [26]  33
00:06:23: RADIUS:  Cisco AVpair        [1]   27  "interface=Virtual-Access1"
00:06:23: RADIUS:  NAS-Port-Type       [61]  6   Virtual                   [5]
00:06:23: RADIUS:  Service-Type        [6]   6   Framed                    [2]
00:06:23: RADIUS:  NAS-IP-Address      [4]   6   12.1.0.1
00:06:23: RADIUS:  Acct-Session-Id     [44]  10  "00000002"
00:06:23: RADIUS:  Received from id 0 10.1.0.2:1645, Access-Accept, len 103
00:06:23: RADIUS:  authenticator 5D 2D 9F 25 11 15 45 B2 - 54 BB 7F EB CE 79 20 3B
00:06:23: RADIUS:  Vendor, Cisco       [26]  33
00:06:23: RADIUS:  Cisco AVpair        [1]   27  "interface=Virtual-Access1"
00:06:23: RADIUS:  Service-Type        [6]   6   Framed                    [2]
00:06:23: RADIUS:  Framed-Protocol     [7]   6   PPP                       [1]
00:06:23: RADIUS:  Framed-IP-Netmask   [9]   6   255.255.255.255
00:06:23: RADIUS:  Framed-IP-Address   [8]   6   10.60.0.1
00:06:23: RADIUS:  Framed-Route     [22] 26 "10.80.0.1 255.255.255.255 10.60.0.1 100"
<=======
00:06:23: RADIUS:  Received from id 2
00:06:24: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1, changed state
to up
00:06:25: AAA/AUTHOR:  Processing PerUser AV route
00:06:25: Vi1 AAA/PERUSER/ROUTE: route string: IP route  10.80.0.1 255.255.255.255
10.60.0.1 100

00:06:25: RADIUS/ENCODE(00000002): Unsupported AAA attribute timezone
00:06:25: RADIUS(00000002): sending
00:06:25: RADIUS:  Send to unknown id 1 10.1.0.2:1646, Accounting-Request, len 278
00:06:25: RADIUS:  authenticator E0 CC 99 EB 49 18 B9 78 - 4A 09 60 0F 4E 92 24 C6
00:06:25: RADIUS:  Acct-Session-Id     [44]  10  "00000002"
00:06:25: RADIUS:  Tunnel-Server-Endpoi[67]  12  00:"10.1.1.1"
00:06:25: RADIUS:  Tunnel-Client-Endpoi[66]  12  00:"10.1.1.2"
00:06:25: RADIUS:  Tunnel-Assignment-Id[82]  15  00:"from_isdn101"
00:06:25: RADIUS:  Tunnel-Type         [64]  6   00:L2TP                   [3]
00:06:25: RADIUS:  Acct-Tunnel-Connecti[68]  12  "2056100083"
00:06:25: RADIUS:  Tunnel-Client-Auth-I[90]  10  00:"isdn101"
00:06:25: RADIUS:  Tunnel-Server-Auth-I[91]  6   00:"lns"
00:06:25: RADIUS:  Framed-Protocol     [7]   6   PPP                       [1]
00:06:25: RADIUS:  Framed-Route     [22] 39 "10.80.0.1 255.255.255.255 10.60.0.1 100"
<========
00:06:25: RADIUS:  Framed-IP-Address   [8]   6   10.60.0.1
00:06:25: RADIUS:  Vendor, Cisco       [26]  35
00:06:25: RADIUS:  Cisco AVpair        [1]   29  "connect-progress=LAN Ses Up"
00:06:25: RADIUS:  Authentic           [45]  6   RADIUS                    [1]
00:06:25: RADIUS:  User-Name           [1]   14  "nari@trw1001"
00:06:25: RADIUS:  Acct-Status-Type    [40]  6   Start                     [1]
00:06:25: RADIUS:  NAS-Port            [5]   6   1
00:06:25: RADIUS:  Vendor, Cisco       [26]  33
00:06:25: RADIUS:  Cisco AVpair        [1]   27  "interface=Virtual-Access1"
00:06:25: RADIUS:  NAS-Port-Type       [61]  6   Virtual                   [5]
00:06:25: RADIUS:  Service-Type        [6]   6   Framed                    [2]
```

```
00:06:25: RADIUS:  NAS-IP-Address       [4]   6   10.1.0.1
00:06:25: RADIUS:  Acct-Delay-Time      [41]  6   0
```

Additional References

The following sections provide references related to Framed-Route in RADIUS Accounting.

Related Documents

Related Topic	Document Title
Configuring AAA	"Authentication, Authorization, and Accounting (AAA)" section of the *Cisco IOS Security Configuration Guide*
Configuring RADIUS	The "Configuring RADIUS" chapter of "Part 2: Security Server Protocols" section of the *Cisco IOS Security Configuration Guide*
Configuring RADIUS attribute screening	*RADIUS Attribute Screening*
Security commands	*Cisco IOS Security Command Reference*, Release 12.3 T

Standards

Standards	Title
No new or modified standards are supported by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

This feature uses no new or modified commands. To see the command pages for the commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

Offload Server Accounting Enhancement

Feature History

Release	Modification
12.2(4)T	This feature was introduced.

This feature module describes the Offload Server Accounting Enhancement feature in Cisco IOS Release 12.2(4)T. It includes the following sections:

- Feature Overview, page 227
- Supported Platforms, page 228
- Supported Standards, MIBs, and RFCs, page 229
- Prerequisites, page 229
- Configuration Tasks, page 229
- Configuration Examples, page 230
- Command Reference, page 231
- Glossary, page 232

Feature Overview

The Offload Server Accounting Enhancement feature allows users to configure their network access servers (NAS) to synchronize authentication and accounting information—NAS-IP-Address (attribute 4) and Class (attribute 25)—with the offload server.

An offload server interacts with a NAS via Virtual Private Network (VPN) to perform required Point-to-Point Protocol (PPP) negotiation for calls. The NAS performs call preauthentication, wheras the offload server performs user authentication. T his feature allows the authentication and accounting data of the NAS to synchronize with the offload server as follows:

- During preauthentication, the NAS generates a unique session-id, adding the Acct-Session-Id (attribute 44) before the existing session-id (NAS-IP-Address), and retrieves a Class attribute. The new session-id is sent in preauthentication requests and resource accounting requests; the Class attribute is sent in resource accounting requests.

Note Unique session-ids are needed when multiple NASs are being processed by one offload server.

- The NAS-IP-Address, the Acct-Session-Id, and the Class attribute are transmitted to the offload server via Layer 2 Forwarding (L2F) options.

- The offload server will include the new, unique session-id in user access requests and user session accounting requests. The Class attribute that was passed from the NAS will be included in the user access request, but a new Class attribute will be received in the user access reply; this new Class attribute should be included in user session accounting requests.

Benefits

The Offload Server Accounting Enhancement feature allows users to maintain authentication and accounting information between their NAS and offload server.

Although NASs can already synchronize information with an offload server, this feature extends the functionality to include a unique session-id, adding the Acct-Session-Id (attribute 44) before the existing session-id (NAS-IP-Address), and Class (attribute 25) information collected by the NASs.

Related Documents

- "Configuring Virtual Private Networks" chapter, *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2
- *Cisco IOS Security Configuration Guide*, Release 12.2
- *Cisco IOS Security Command Reference*, Release 12.2

Supported Platforms

- Cisco 7100 series
- Cisco 7200 series
- Cisco 7500 series

Determining Platform Support Through Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Feature Navigator. Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image.

To access Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml.

RFCs

No new or modified RFCs are supported by this feature.

Prerequisites

Before configuring the Offload Server Accounting Enhancement feature, you must perform the following tasks:

- Enable AAA. (For more information, refer to chapter "Configuring Authentication" of the *Cisco IOS Security Configuration Guid*e, Release 12.2.)
- Enable VPN. (For more information, refer to the chapter "Configuring Virtual Private Networks" of the *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2.)

Configuration Tasks

See the following sections for configuration tasks for the Offload Server Accounting Enhancement feature. Each task in the list is identified as either required or optional.

- Configuring Unique Session IDs (required)
- Configuring Offload Server to Synchronize with NAS Clients (required)
- Verifying Offload Server Accounting (optional)

Configuring Unique Session IDs

To maintain unique session IDs among NASs, use the following global configuration command. When multiple NASs are being processed by one offload server, this feature must be enabled by all NASs and by the offload server to ensure a common and unique session-id.

Command	Purpose
`Router(config)# radius-server attribute 44 extend-with-addr`	Adds the accounting IP address in front of the existing AAA session ID. **Note** The unique session-id is different from other NAS session-ids because it adds the Acct-Session-Id (attribute 44) before the existing session-id (NAS-IP-Address).

Configuring Offload Server to Synchronize with NAS Clients

To configure the offload server to synchronize accounting session information with the NAS clients, use the following global configuration command:

Command	Purpose
`Router(config)# radius-server attribute 44 sync-with-client`	Configures the offload server to synchronize accounting session information with the NAS clients.

Verifying Offload Server Accounting

To verify whether the NAS has synchronized authentication and accounting data with the offload server, use the following commands in privileged EXEC mode:

Command	Purpose
`Router# more system:running-config`	Displays the contents of the current running configuration file. (Note that the **more system:running-config** command has replaced the **show running-config** command.)
`Router(config)# debug radius`	Displays information associated with RADIUS. The output of this command shows whether attribute 44 is being sent in access requests. The output, however, does not show the entire value for attribute 44. To view the entire value for attribute 44, refer to your RADIUS server log.

Configuration Examples

This section provides the following configuration examples:

- Unique Session ID Configuration Example
- Offload Server Synchronization with NAS Clients Example

Unique Session ID Configuration Example

The following example shows how to configure unique session IDs among NASs:

```
aaa new-model
aaa authentication ppp default group radius
radius-server host 10.100.1.34
radius-server attribute 44 include-in-access-req
radius-server attribute 44 extend-with-addr
```

Offload Server Synchronization with NAS Clients Example

The following example shows how to configure the offload server to synchronize accounting session information with NAS clients:

```
radius-server attribute 44 sync-with-client
```

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **radius-server attribute 44 extend-with-addr**
- **radius-server attribute 44 sync-with-client**

Glossary

AAA—authentication, authorization, and accounting. Suite of network security services that provide the primary framework through which access control can be set up on your Cisco router or access server.

Acct-Session-ID (attribute 44)—A unique accounting identifier that makes it easy to match start and stop records in a log file. Acct-Session ID numbers restart at 1 each time the router is power-cycled or the software is reloaded.

Class (attribute 25)—An accounting attribute. Arbitrary value that the network access server includes in all accounting packets for this user if the attribute is supplied by the RADIUS server.

L2F—Layer 2 Forwarding. A Layer 2 tunneling protocol that enables an ISP or other access service to create a virtual tunnel to link customer remote sites or remote users with corporate home networks. In particular, a network access server (NAS) at the ISP point of presence (POP) exchanges PPP messages with the remote users and communicates by L2F or L2TP requests and responses with the customer tunnel server to set up tunnels.

NAS—network access server. A Cisco platform (or collection of platforms, such as an AccessPath system) that interfaces between the packet world (for example, the Internet) and the circuit world (for example, the public switched telephone network).

NAS-IP Address (attribute 4)—Specifies the IP address of the network access server that is requesting authentication. The default value is 0.0.0.0/0.

PPP—Point-to-Point Protocol. Successor to SLIP that provides router-to-router and host-to-network connections over synchronous and asynchronous circuits. Whereas SLIP was designed to work with IP, PPP was designed to work with several network layer protocols, such as IP, IPX, and ARA. PPP also has built-in security mechanisms, such as CHAP and PAP. PPP relies on two protocols: LCP and NCP.

RADIUS—Remote Authentication Dial-In User Service. RADIUS is a distributed client/server system that secures networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco routers and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.

VPN—A system that permits dial-in networks to exist remotely to home networks, while giving the appearance of being directly connected. VPNs use L2TP and L2F to terminate the Layer 2 and higher parts of the network connection at the LNS instead of the LAC.

Per VRF AAA

The Per VRF AAA functionality enables AAA services to be based on VPN routing and forwarding (VRF) instances. The Provider Edge (PE) or Virtual Home Gateway (VHG) can now communicate directly with the customer's RADIUS server, which is associated with the customer's VPN, without having to go through a RADIUS proxy. Thus, ISPs can scale their VPN offerings more efficiently because they no longer have to use RADIUS proxies and they can provide their customers with the flexibility they demand.

Also, for Cisco IOS Release 12.2(15)T or later, you can use a customer template, which may be stored either locally or remotely, and AAA services can be performed on the information that is stored in the customer template.

Feature Specifications for the Per VRF AAA Feature

Feature History	
Release	**Modification**
12.2(1)DX	This feature was introduced on the Cisco 7200 series and the Cisco 7401ASR.
12.2(2)DD	This feature was integrated into Cisco IOS Release 12.2(2)DD. The **ip vrf forwarding** and **radius-server domain-stripping** commands were added.
12.2(4)B	This feature was integrated into Cisco IOS Release 12.2(4)B.
12.2(13)T	This feature was integrated into Cisco IOS Release 12.2(13)T.
12.2(15)T	This feature was integrated into Cisco IOS Release 12.2(15)T. The **aaa authorization template** command was added.
12.2(27)SB	This feature was integrated into Cisco IOS Release 12.2(27)SB.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Restrictions for Per VRF AAA, page 234
- Information About Per VRF AAA, page 234

- How to Configure Per VRF AAA, page 238
- Configuration Examples for Per VRF AAA, page 250
- Additional References, page 254
- Command Reference, page 256
- Glossary, page 257

Restrictions for Per VRF AAA

- Per VRF AAA is supported only for RADIUS servers.
- Because all functionalities must be consistent between the NAS and the AAA servers, the operational parameters should be defined once per VRF rather than set per server group.
- The ability to configure a customer template either locally or remotely is available only for Cisco IOS Release 12.2(15)T and later.

Information About Per VRF AAA

To use Per VRF AAA, you should understand the following concepts:

- Per VRF AAA Functionality Overview, page 234
- Benefits of Per VRF AAA, page 235
- New Vendor-Specific Attributes (VSAs), page 235

Per VRF AAA Functionality Overview

To support AAA on a per customer basis, some AAA features must be made VRF aware. That is, ISPs must be able to define operational parameters—such as AAA server groups, method lists, system accounting, and protocol-specific parameters—and bind those parameters to a particular VRF instance. Defining and binding the operational parameters can be accomplished using one or more of the following methods:

- Virtual private dialup network (VPDN) virtual template or dialer interfaces that are configured for a specific customer
- Locally defined customer templates—Per VPN with customer definitions. The customer template is stored locally on the VHG. This method can be used to associate a remote user with a specific VPN based on the domain name or dialed number identification service (DNIS) and provide the VPN-specific configuration for virtual access interface and all operational parameters for the customer AAA server.
- Remotely defined customer templates—Per VPN with customer definitions that are stored on the service provider AAA server in a RADIUS profile. This method is used to associate a remote user with a specific VPN based on the domain name or DNIS and provide the VPN-specific configuration for the virtual access interface and all operational parameters for the AAA server of the customer.

Note The ability to configure locally or remotely defined customer templates is available only with Cisco IOS Release 12.2(15)T and later.

Benefits of Per VRF AAA

Configuration Support

ISPs can partition AAA services on a per VRF basis. Thus, ISPs can allow their customers to control some of their own AAA services.

Server Group List Extension

The list of servers in server groups is extended to include the definitions of private servers in addition to references to the hosts in the global configuration, allowing access to both customer servers and global service provider servers simultaneously.

New Vendor-Specific Attributes (VSAs)

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating vendor-specific information between the network access server and the RADIUS server by using the vendor-specific attribute (attribute 26). Attribute 26 encapsulates vendor specific attributes, thereby, allowing vendors to support their own extended attributes otherwise not suitable for general use.

The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. Cisco's vendor-ID is 9, and the supported option has vendor-type 1, which is named "cisco-avpair." The value is a string of the following format:

```
protocol : attribute sep value *
```

"Protocol" is a value of the Cisco "protocol" attribute for a particular type of authorization. "Attribute" and "value" are an appropriate attribute-value (AV) pair defined in the Cisco TACACS+ specification, and "sep" is "=" for mandatory attributes and "*" for optional attributes. This allows the full set of features available for TACACS+ authorization to also be used for RADIUS.

Table 16 summarizes the VSAs that are now supported with Per VRF AAA.

Table 16 Newly Supported VSAs for Per VRF AAA

VSA Name	Value Type	Decription
Note Each VSA must have the prefix "template:" before the VSA name, unless a different prefix is explicitly stated.		
account-delay	string	This VSA must be "on." The functionality of this VSA is equal to the **aaa accounting delay-start** command for the customer template.
account-send-stop	string	This VSA must be "on." The functionality of this VSA is equal to the **aaa accounting send stop-record authentication failure** command.
attr-44	string	This VSA must be "access-req." The functionality of this VSA is equal to the **radius-server attribute 44 include-in-access-req** command.
ip-addr	string	This VSA specifies the IP address, followed by the mask that the router uses to indicate its own IP address and mask in negotiation with the client; for example, ip-addr=1.2.3.4 255.255.255.255
ip-unnumbered	string	This VSA specifies the name of an interface on the router. The functionality of this VSA is equal to the **ip unnumbered** command, which specifies an interface name such as "Loopback 0."
ip-vrf	string	This VSA specifies which VRF will be used for the packets of the end user. This VRF name match the name that is used on the router via the **ip vrf forwarding** command.
peer-ip-pool	string	This VSA specifies the name of an IP address pool from which an address will be allocated for the peer. This pool should be configured using the **ip local pool** command or should be automatically downloadable via RADIUS.
ppp-acct-list	string	This VSA defines the accounting method list that is to be used for PPP sessions. The VSA syntax is as follows: "ppp-acct-list=[start-stop | stop-only | none] group X [group Y] [broadcast]." It is equal to the **aaa accounting network mylist** command functionality. The user must specify at least one of the following options: start-stop, stop-only, or none. If either start-stop or stop-only is specified, the user must specify at least one, but not more than four, group arguments. Each group name must consist of integers. The servers in the group should have already been identified in the access-accept via the VSA "rad-serv." After each group has been specified, the user can specify the broadcast option

Table 16 Newly Supported VSAs for Per VRF AAA (continued)

VSA Name	Value Type	Decription
ppp-authen-list	string	This VSA defines which authentication method list is to be used for PPP sessions and, if more than one method is specified, in what order the methods should be used. The VSA syntax is as follows: "ppp-authen-list=[groupX | local | local-case | none | if-needed]," which is equal to the **aaa authentication ppp mylist** command functionality. The user must specify at least one, but no more than four, authentication methods. If a server group is specified, the group name must be an integer. The servers in the group should have already been identified in the access-accept via the VSA "rad-serv."
ppp-authen-type	string	This VSA allows the end user to specify at least one of the following authentication types: pap, chap, eap, ms-chap, ms-chap-v2, any, or a combination of the available types that is separated by spaces. The end user will be permitted to log in using only the methods that are specified in this VSA. PPP will attempt these authentication methods in the order presented in the attribute.
ppp-author-list	string	This VSA defines the authorization method list that is to be used for PPP sessions. It indicates which methods will be used and in what order. The VSA syntax is as follows: "ppp-author-list=[groupX] [local] [if-authenticated] [none]," which is equal to the **aaa authoirzation network mylist** command functionality. The user must specify at least one, but no more than four, authorization methods. If a server group is specified, the group name must be an integer. The servers in the group should have already been identified in the access-accept via the VSA "rad-serv."

Note The RADIUS VSAs—rad-serv, rad-server-filter, rad-serv-source-if, and rad-serv-vrf—must have the prefix "aaa:" before the VSA name.

Table 16 Newly Supported VSAs for Per VRF AAA (continued)

VSA Name	Value Type	Decription
rad-serv	string	This VSA indicates the IP address, key, timeout, and retransmit number of a server, as well as the group of the server. The VSA syntax is as follows: "rad-serv=a.b.c.d [key SomeKey] [auth-port X] [acct-port Y] [retransmit V] [timeout W]." Other than the IP address, all paramters are optional and can be issued in any order. If the optional parmaters are not specified, their default values will be used. The key cannot contain any spaces; for "retransmit V," "V" can range from 1-100; for "timeout W," the "W" can range from 1-1000.
rad-serv-filter	string	The VSA syntax is as follows: "rad-serv-filter=authorization | accounting-request | reply-accept | reject-filtername." The filtername must be defined via the **radius-server attribute list filtername** command.
rad-serv-source-if	string	This VSA specifies the name of the interface that is used for transmitting RADIUS packets. The specified interface must match the interface configured on the router.
rad-serv-vrf	string	This VSA specifies the name of the VRF that is used for transmitting RADIUS packets. The VRF name should match the name that was specified via the **ip vrf forwarding** command.

How to Configure Per VRF AAA

This following sections contain procedures for possible deployment scenarios for using Per VRF AAA:

- Configuring Per VRF AAA, page 239 (required)
- Configuring Per VRF AAA Using Local Customer Templates, page 244 (optional)
- Configuring Per VRF AAA Using Remote Customer Templates, page 247 (optional)
- Verifying VRF Routing Configurations, page 249 (optional)
- Troubleshooting Per VRF AAA Configurations, page 250 (optional)

Configuring Per VRF AAA

This section contains the following procedures:

- Configuring AAA, page 239
- Configuring Server Groups, page 239
- Configuring Authentication, Authorization, and Accounting for Per VRF AAA, page 240
- Configuring RADIUS-Specific Commands for Per VRF AAA, page 242
- Configuring Interface-Specific Commands for Per VRF AAA, page 242

Configuring AAA

To enable AAA, you should complete the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables AAA globally.

Configuring Server Groups

To configure server groups, you should complete the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa group server radius** *groupname*

4. **server-private** *ip-address* [**auth-port** *port-number* | **acct-port** *port-number*] [**non-standard**] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string*]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa group server radius** *groupname* **Example:** `Router(config)# aaa group server radius v2.44.com`	Groups different RADIUS server hosts into distinct lists and distinct methods. Enters server-group configuration mode.
Step 4	`server-private` *ip-address* [**auth-port** *port-number* | **acct-port** *port-number*] [**non-standard**] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string*] **Example:** `Router(config-sg-radius)# server-private 10.10.130.2 auth-port 1600 acct-port 1666 key ww`	Configures the IP address of the private RADIUS server for the group server. **Note** If private server parameters are not specified, global configurations will be used. If global configurations are not specified, default values will be used.
Step 5	**exit** **Example:** `Router(config-sg-radius)# exit`	Exits from server-group configuration mode; returns to global configuration mode.

Configuring Authentication, Authorization, and Accounting for Per VRF AAA

To configure AAA for Per VRF AAA, you should complete the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa authentication ppp** {**default** | *list-name*} *method1* [*method2...*]
4. **aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} *method1* [*method2...*]
5. **aaa accounting system default vrf** *vrf-name*] {**start-stop** | **stop-only** | **wait-start** | **none**} [**broadcast**] **group** *groupname*

6. **aaa accounting delay-start** [**vrf** *vrf-name*]
7. **aaa accounting send stop-record authentication failure** [**vrf** *vrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa authentication ppp** {**default** | *list-name*} *method1* [*method2...*] **Example:** `Router(config)# aaa authentication ppp method_list_v2.44.com group v2.44.com`	Specifies one or more AAA authentication methods for use on serial interfaces that are running PPP.
Step 4	**aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} *method1* [*method2...*] **Example:** `Router(config)# aaa authorization network method_list_v2.44.com group v2.44.com`	Sets parameters that restrict user access to a network.
Step 5	**aaa accounting system default** [**vrf** *vrf-name*] {**start-stop** | **stop-only** | **wait-start** | **none**} [**broadcast**] **group** *groupname* **Example:** `Router(config)# aaa accounting system default vrf v2.44.com start-stop group v2.44.com`	Enables AAA accounting of requested services for billing or security purposes when you use RADIUS.
Step 6	**aaa accounting delay-start vrf** [*vrf-name*] **Example:** `Router(config)# aaa acounting delay-start vrf v2.44.com`	Displays generation of the start accounting records until the user IP address is established.
Step 7	**aaa accounting send stop-record authentication failure** [**vrf** *vrf-name*] **Example:** `Router(config)# aaa accounting send stop-record authentication failure vrf v2.44.com`	Generates accounting “stop” records for users who fail to authenticate at login or during session negotiation.

Configuring RADIUS-Specific Commands for Per VRF AAA

To configure RADIUS-specific commands for Per VRF AAA, you should complete the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip radius source-interface** *subinterface-name* [**vrf** *vrf-name*]
4. **radius-server attribute 44 include-in-access-req** [**vrf** *vrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip radius source-interface** *subinterface-name* [**vrf** *vrf-name*] **Example:** `Router(config)# ip radius source-interface loopback55`	Forces RADIUS to use the IP address of a specified interface for all outgoing RADIUS packets and enables the specification on a per-VRF basis.
Step 4	**radius-server attribute 44 include-in-access-req** [**vrf** *vrf-name*] **Example:** `Router(config)# radius-server attribute 44 include-in-access-req vrf v2.44.com`	Sends RADIUS attribute 44 in access request packets before user authentication and enables the specification on a per-VRF basis.

Configuring Interface-Specific Commands for Per VRF AAA

To configure interface-specific commands for Per VRF AAA, you should complete the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **ip vrf forwarding** *vrf-name*
5. **ppp authentication** {*protocol1* [*protocol2...*]} *listname*

6. **ppp authorization** *list-name*
7. **ppp accounting default**
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface loopback11`	Configures an interface type and enters interface configuration mode.
Step 4	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-sg)# ip vrf forwarding v2.44.com`	Associates a VRF with an interface.
Step 5	**ppp authentication** {*protocol1* [*protocol2...*]} *listname* **Example:** `Router(config)# ppp authentication chap callin V2_44_com`	Enables Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP) or both and specifies the order in which CHAP and PAP authentication are selected on the interface.
Step 6	**ppp authorization** *list-name* **Example:** `Router(config)# ppp authorization V2_44_com`	Enables AAA authorization on the selected interface.
Step 7	**ppp accounting default** **Example:** `Router(config)# ppp accounting default`	Enables AAA accounting services on the selected interface.
Step 8	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Configuring Per VRF AAA Using Local Customer Templates

This section contains the following procedures:

- Prerequisites, page 244
- Configuring Authorization for Per VRF AAA with Local Customer Templates, page 244
- Configuring Local Customer Templates, page 245

Prerequisites

Before configuring authorization for Per VRF AAA with local templates, you should perform the following tasks:

- Configure AAA. (Perform the tasks as outlined in the section "Configuring AAA.")
- Configure Server Groups (Perform the tasks as outlined in the section "Configuring Server Groups.")
- Configure AAA for Per VRF AAA. (Perform the tasks as outlined in the section "Configuring Authentication, Authorization, and Accounting for Per VRF AAA.")

Configuring Authorization for Per VRF AAA with Local Customer Templates

To configure authorization for Per VRF AAA with local templates, you should complete the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa authorization template**
4. **aaa authorization network default local**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**aaa authorization template** **Example:** `Router(config)# aaa authorization template`	Enables the use of local or remote templates.
Step 4	**aaa authorization network default local** **Example:** `Router(config)# aaa authorization network default local`	Specifies local as the default method for authorization.

Configuring Local Customer Templates

To configure local customer templates, you should complete the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vpdn search-order domain**
4. **template** *name* [**default** | **exit** | **multilink** | **no** | **peer** | **ppp**]
5. **peer default ip address pool** *pool-name*
6. **ppp authentication** {*protocol1* [*protocol2...*]} [**if-needed**] [*list-name* | **default**] [**callin**] [**one-time**]
7. **ppp authorization** [**default** | *list-name*]
8. **aaa accounting** {**auth-proxy** | **system** | **network** | **exec** | **connection** | **commands** *level*} {**default** | *list-name*} [**vrf** *vrf-name*] {**start-stop** | **stop-only** | **wait-start** | **none**} [**broadcast**] **group** *groupname*
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vpdn search-order domain** **Example:** `Router (config)# vpdn search-order domain`	Looks up the profiles based on domain.

	Command or Action	Purpose
Step 4	**template** *name* [**default** | **exit** | **multilink** | **no** | **peer** | **ppp**] **Example:** `Router (config)# template v2.44.com`	Creates a customer profile template and assigns a unique name that relates to the customer that will be receiving it. Enters template configuration mode. **Note** Steps 5, 6, and 7 are optional. Enter **multilink**, **peer**, and **ppp** keywords appropriate to customer application requirements.
Step 5	**peer default ip address pool** *pool-name* **Example:** `Router(config-template)# peer default ip address pool v2_44_com_pool`	(Optional) Specifies that the customer profile to which this template is attached will use a local IP address pool with the specified name.
Step 6	**ppp authentication** {*protocol1* [*protocol2...*]} [**if-needed**] [*list-name* | **default**] [**callin**] [**one-time**] **Example:** `Router(config-template)# ppp authentication chap`	(Optional) Sets the PPP link authentication method.
Step 7	**ppp authorization** [**default** | *list-name*] **Example:** `Router(config-template)# ppp authorization v2_44_com`	(Optional) Sets the PPP link authorization method.
Step 8	**aaa accounting** {**auth-proxy** | **system** | **network** | **exec** | **connection** | **commands** *level*} {**default** | *list-name*} [**vrf** *vrf-name*] {**start-stop** | **stop-only** | **wait-start** | **none**} [**broadcast**] **group** *groupname* **Example:** `Router(config-template)# aaa accounting v2_44_com`	(Optional) Enables AAA operational parameters for the specified customer profile.
Step 9	**exit** **Example:** `Router(config-template)# exit`	Exits from template configuration mode; returns to global configuration mode.

Configuring Per VRF AAA Using Remote Customer Templates

This section contains the following procedures:

- Prerequisites, page 247
- Configuring Authentication for Per VRF AAA with Remote Customer Profiles, page 247
- Configuring Authorization for Per VRF AAA with Remote Customer Profiles, page 248
- Configuring the RADIUS Profile on the SP RADIUS Server, page 249

Prerequisites

Before configuring authorization for Per VRF AAA with remote customer templates, you should perform the following tasks:

- Configure AAA. (Perform the tasks as outlined in the section "Configuring Per VRF AAA.")
- Configure Server Groups. (Perform the tasks as outlined in the section "Configuring Server Groups."

Configuring Authentication for Per VRF AAA with Remote Customer Profiles

To configure authentication for Per VRF AAA with remote customer profiles, you should perform the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa authentication ppp** {**default** | *list-name*} *method1* [*method2...*]
4. **aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} *method1* [*method2...*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**aaa authentication ppp** {**default** | *list-name*} *method1* [*method2...*] **Example:** `Router# ppp authentication ppp default group radius`	Specifies one or more authentication, authorization, and accounting (AAA) authentication methods for use on serial interfaces that are running PPP.
Step 4	**aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} *method1* [*method2...*] **Example:** `Router# aaa authorization network default group sp`	Sets parameters that restrict user access to a network.

Configuring Authorization for Per VRF AAA with Remote Customer Profiles

To configuring authorization for Per VRF AAA with remote customer profiles, you should perform the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa authorization template**
4. **aaa authorization network default sp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**aaa authorization template** **Example:** `Router(config)# aaa authorization template`	Enables use of local or remote templates.
Step 4	**aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} *method1* [*method2...*] **Example:** `Router(config)# aaa authorization network default sp`	Specifies the server group that is named as the default method for authorization.

Configuring the RADIUS Profile on the SP RADIUS Server

Configure the RADIUS profile on the SP RADIUS server. See the section "Per VRF AAA Using a Remote RADIUS Customer Template Example" for an example of how to update the RADIUS profile.

Verifying VRF Routing Configurations

To verify VRF routing configurations, you need to complete the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **show ip route vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**show ip route vrf** *vrf-name* **Example:** `Router(config)# show ip route vrf northvrf`	Displays the IP routing table associated with a VRF.

Troubleshooting Per VRF AAA Configurations

To troubleshoot the Per VRF AAA feature, use at least one of the following commands in EXEC mode:

Command	Purpose
`Router# debug aaa accounting`	Displays information on accountable events as they occur.
`Router# debug aaa authentication`	Displays information on AAA authentication.
`Router# debug aaa authorization`	Displays information on AAA authorization.
`Router# debug ppp negotiation`	Displays information on traffic and exchanges in an internetwork implementing PPP.
`Router# debug radius`	Displays information associated with RADIUS.
`Router# debug vpdn event`	Displays Layer 2 Transport Protocol (L2TP) errors and events that are a part of normal tunnel establishment or shutdown for VPNs.
`Router# debug vpdn error`	Displays debug traces for VPN.

Configuration Examples for Per VRF AAA

This section contains the following configuration examples:

- Per VRF AAA Example, page 251
- Per VRF AAA Using a Locally Defined Customer Template Example, page 251
- Per VRF AAA Using a Remote RADIUS Customer Template Example, page 252
- Locally Configured Customer Template with RADIUS Attribute Screening and Broadcast Accounting Example, page 252
- Remotely Configured Customer Template with RADIUS Attribute Screening and Broadcast Accounting Example, page 253

Per VRF AAA Example

The following example shows how to configure Per VRF AAA using a AAA server group with associated private servers:

```
aaa new-model

aaa authentication ppp method_list_v1.55.com group v1.55.com
aaa authorization network method_list_v1.55.com group v1.55.com
aaa accounting network method_list_v1.55.com start-stop group v1.55.com
aaa accounting system default vrf v1.55.com start-stop group v1.55.com
aaa accounting delay-start vrf v1.55.com
aaa accounting send stop-record authentication failure vrf v1.55.com

aaa group server radius v1.55.com
    server-private 10.10.132.4 auth-port 1645 acct-port 1646 key ww
    ip vrf forwarding v1.55.com

ip radius source-interface loopback55
radius-server attribute 44 include-in-access-req vrf v1.55.com
```

Per VRF AAA Using a Locally Defined Customer Template Example

The following example shows how to configure Per VRF AAA using a locally defined customer template with a AAA server group that has associated private servers:

```
aaa new-model
aaa authentication ppp method_list_v1.55.com group v1.55.com
aaa authorization network method_list_v1.55.com group v1.55.com
aaa authorization network default local
aaa authorization template
aaa accounting network method_list_v1.55.com start-stop group v1.55.com
aaa accounting system default vrf v1.55.com start-stop group v1.55.com

aaa group server radius V1_55_com
  server-private 10.10.132.4 auth-port 1645 acct-port 1646 key ww
  ip vrf forwarding V1.55.com

template V1.55.com
  peer default ip address pool V1_55_com_pool
  ppp authentication chap callin V1_55_com
  ppp authorization V1_55_com
  ppp accounting V1_55_com
  aaa accounting delay-start
  aaa accounting send stop-record authentication failure
  radius-server attribute 44 include-in-access-req
  ip vrf forwarding v1.55.com
  ip radius source-interface Loopback55
```

Per VRF AAA Using a Remote RADIUS Customer Template Example

The following examples shows how to configure Per VRF AAA using a remotely defined customer template on the SP RADIUS server with a AAA server group that has associated private servers:

```
aaa new-model
aaa authentication ppp default group radius
aaa authorization template
aaa authorization network default group sp

aaa group server radius sp
   server 3.3.3.3

radius-server host 3.3.3.3 auth-port 1645 acct-port 1646 key sp_key
```

The following RADIUS server profile is configured on the SP RADIUS server:

```
cisco-avpair = "aaa:rad-serv#1=10.10.132.4 key ww"
cisco-avpair = "aaa:rad-serv-vrf#1=V1.55.com"
cisco-avpair = "aaa:rad-serv-source-if#1=Loopback 55"
cisco-avpair = "template:ppp-authen-list=group 1"
cisco-avpair = "template:ppp-author-list=group 1"
cisco-avpair = "template:ppp-acct-list= start-stop group 1"
cisco-avpair = "template:account-delay=on"
cisco-avpair = "template:account-send-stop=on"
cisco-avpair = "template:rad-attr44=access-req"
cisco-avpair = "template:peer-ip-pool=V1.55-pool"
cisco-avpair = "template:ip-vrf=V1.55.com"
cisco-avpair = "template:ip-unnumbered=Loopback 55"
framed-protocol = ppp
service-type = framed
```

Locally Configured Customer Template with RADIUS Attribute Screening and Broadcast Accounting Example

The following example shows how to create a locally configured template for a single customer, configuring additional features including RADIUS attribute screening and broadcast accounting:

```
aaa authentication ppp default local group radius
aaa authentication ppp V1_55_com group V1_55_com
aaa authorization template
aaa authorization network default local group radius
aaa authorization network V1_55_com group V1_55_com
aaa accounting network V1_55_com start-stop broadcast group V1_55_com group SP_AAA_server

aaa group server radius SP_AAA_server
 server 10.10.100.7 auth-port 1645 acct-port 1646

aaa group server radius V1_55_com
 server-private 10.10.132.4 auth-port 1645 acct-port 1646
 authorization accept min-author
 accounting accept usage-only
 ip vrf forwarding V1.55.com

ip vrf V1.55.com
 rd 1:55
 route-target export 1:55
 route-target import 1:55
```

```
template V1.55.com
 peer default ip address pool V1.55-pool
 ppp authentication chap callin V1_55_com
 ppp authorization V1_55_com
 ppp accounting V1_55_com
 aaa accounting delay-start
 aaa accounting send stop-record authentication failure
 radius-server attribute 44 include-in-access-req

vpdn-group V1.55
 accept-dialin
  protocol l2tp
  virtual-template 13
 terminate-from hostname lac-lb-V1.55
 source-ip 10.10.104.12
 lcp renegotiation always
 l2tp tunnel password 7 060506324F41

interface Virtual-Template13
 ip vrf forwarding V1.55.com
 ip unnumbered Loopback55
 ppp authentication chap callin
 ppp multilink

ip local pool V1.55-pool 42.1.55.10 42.1.55.19 group V1.55-group

ip radius source-interface Loopback0
ip radius source-interface Loopback55 vrf V1.55.com

radius-server attribute list min-author
 attribute 6-7,22,27-28,242
radius-server attribute list usage-only
 attribute 1,40,42-43,46

radius-server host 10.10.100.7 auth-port 1645 acct-port 1646 key ww
radius-server host 10.10.132.4 auth-port 1645 acct-port 1646 key ww
```

Remotely Configured Customer Template with RADIUS Attribute Screening and Broadcast Accounting Example

The following example shows how to create a remotely configured template for a single customer, configuring additional features including RADIUS attribute screening and broadcast accounting:

```
aaa authentication ppp default local group radius
aaa authorization template
aaa authorization network default local group radius

ip vrf V1.55.com
 rd 1:55
 route-target export 1:55
 route-target import 1:55

vpdn-group V1.55
 accept-dialin
  protocol l2tp
  virtual-template 13
 terminate-from hostname lac-lb-V1.55
 source-ip 10.10.104.12
 lcp renegotiation always
 l2tp tunnel password 7 060506324F41
```

```
interface Virtual-Template13
 no ip address
 ppp authentication chap callin
 ppp multilink

ip local pool V1.55-pool 42.1.55.10 42.1.55.19 group V1.55-group

radius-server attribute list min-author
 attribute 6-7,22,27-28,242
radius-server attribute list usage-only
 attribute 1,40,42-43,46
```

The customer template is stored as a RADIUS server profile for v1.55.com.

```
cisco-avpair = "aaa:rad-serv#1=10.10.132.4 key ww"
cisco-avpair = "aaa:rad-serv-vrf#1=V1.55.com"
cisco-avpair = "aaa:rad-serv-source-if#1=Loopback 55"
cisco-avpair = "aaa:rad-serv#2=10.10.100.7 key ww"
cisco-avpair = "aaa:rad-serv-source-if#2=Loopback 0"
cisco-avpair = "template:ppp-authen-list=group 1"
cisco-avpair = "template:ppp-author-list=group 1"
cisco-avpair = "template:ppp-acct-list= start-stop group 1 group 2 broadcast"
cisco-avpair = "template:account-delay=on"
cisco-avpair = "template:account-send-stop=on"
cisco-avpair = "template:rad-attr44=access-req"
cisco-avpair = "aaa:rad-serv-filter#1=authorization accept min-author"
cisco-avpair = "aaa:rad-serv-filter#1=accounting accept usage-only"
cisco-avpair = "template:peer-ip-pool=V1.55-pool"
cisco-avpair = "template:ip-vrf=V1.55.com"
cisco-avpair = "template:ip-unnumbered=Loopback 55"
framed-protocol = ppp
service-type = framed
```

Additional References

The following sections provide references related to Per VRF AAA.

Related Documents

Related Topic	Document Title
AAA	*Cisco IOS Security Configuration Guide*, Release 12.2
Broadcast Accounting	AAA Broadcast Accounting Feature Guide.
Cisco IOS Security Commands	*Cisco IOS Security Command Reference*, Release 12.2
Cisco IOS Switching Services Commands	*Cisco IOS Switching Services Command Reference*, Release 12.2
Configuring AAA Server Groups	AAA Server Group feature module, Release 12.0(5)T.
Configuring Multiprotocol Label Switching	Configuring Multiprotocol Label Switching" chapter of the *Cisco IOS Switching Services Configuration Guide*, Release 12.2
Configuring Virtual Templates	"Virtual Templates, Profiles, and Networks" chapter of the *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2

Related Topic	Document Title
RADIUS Attribute Screening	RADIUS Attribute Screening Feature Guide.
RADIUS Debug Enhancements	RADIUS Debug Enhancements Feature Guide.

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **aaa accounting**
- **aaa accounting delay-start**
- **aaa accounting send stop-record authentication failure**
- **aaa authorization template**
- **ip radius source-interface**
- **ip vrf forwarding (server-group)**
- **radius-server attribute 44 extend-with-addr**
- **radius-server domain-stripping**
- **server-private (RADIUS)**

Glossary

AAA—authentication, authorization, and accounting. A framework of security services that provide the method for identifying users (authentication), for remote access control (authorization), and for collecting and sending security server information used for billing, auditing, and reporting (accounting).

L2TP—Layer 2 Tunnel Protocol. A Layer 2 tunneling protocol that enables an ISP or other access service to create a virtual tunnel to link customer remote sites or remote users with corporate home networks. In particular, a network access server (NAS) at the ISP point of presence (POP) exchanges PPP messages with the remote users and communicates by L2F or L2TP requests and responses with the customer tunnel server to set up tunnels.

PE—Provider Edge. Networking devices that are located on the edge of a service provider network.

RADIUS—Remote Authentication Dial-In User Service. RADIUS is a distributed client/server system that secures networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco routers and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.

SP—service provider.

VHG—Virtual Home Gateway.

VPDN—Virtual Private Dialup Network.

VPN—Virtual Private Network. A system that permits dial-in networks to exist remotely to home networks, while giving the appearance of being directly connected. VPNs use L2TP and L2F to terminate the Layer 2 and higher parts of the network connection at the LNS instead of the LAC.

VRF—Virtual Route Forwarding. Initially, a router has only one global default routing/forwarding table. VRFs can be viewed as multiple disjoined routing/forwarding tables, where the routes of a user have no correlation with the routes of another user.

RFC-2867 RADIUS Tunnel Accounting

The RFC-2867 RADIUS Tunnel Accounting introduces six new RADIUS accounting types that are used with the RADIUS accounting attribute Acct-Status-Type (attribute 40), which indicates whether an accounting request marks the beginning of user service (start) or the end (stop).

This feature also introduces two new virtual private virtual private dialup network (VPDN) commands that help users better troubleshoot VPDN session events.

Feature History for RFC-2867 RADIUS Tunnel Accounting

Release	Modification
12.2(15)B	This feature was introduced on the Cisco 6400 series, Cisco 7200 series, and the Cisco 7400 series routers.
12.3(4)T	This feature was integrated into Cisco IOS Release 12.3(4)T.
12.2(27)SB	This feature was integrated into Cisco IOS Release 12.2(27)SB.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Restrictions for RFC-2867 RADIUS Tunnel Accounting, page 259
- Information About RFC-2867 RADIUS Tunnel Accounting, page 260
- How to Configure RADIUS Tunnel Accounting, page 264
- Configuration Examples for RADIUS Tunnel Accounting, page 266
- Additional References, page 270
- Command Reference, page 271

Restrictions for RFC-2867 RADIUS Tunnel Accounting

RADIUS tunnel accounting works only with L2TP tunnel support.

Information About RFC-2867 RADIUS Tunnel Accounting

To use RADIUS tunnel attributes and commands, you should understand the following concepts:

- Benefits of RFC-2867 RADIUS Tunnel Accounting, page 260
- RADIUS Attributes Support for RADIUS Tunnel Accounting, page 260

Benefits of RFC-2867 RADIUS Tunnel Accounting

Without RADIUS tunnel accounting support, VPDN with network accounting, which allows users to determine tunnel-link status changes, did not report all possible attributes to the accounting record file. Now that all possible attributes can be displayed, users can better verify accounting records with their Internet Service Providers (ISPs).

RADIUS Attributes Support for RADIUS Tunnel Accounting

Table 17 outlines the new RADIUS accounting types that are designed to support the provision of compulsory tunneling in dialup networks; that is, these attribute types allow you to better track tunnel status changes.

Note The accounting types are divided into two separate tunnel types so users can decide if they want tunnel type, tunnel-link type, or both types of accounting.

Table 17 RADIUS Accounting Types for the Acct-Status-Type Attribute

Type-Name	Number	Description	Additional Attributes[1]
Tunnel-Start	9	Marks the beginning of a tunnel setup with another node.	• User-Name (1)—from client • NAS-IP-Address (4)—from AAA • Acct-Delay-Time (41)—from AAA • Event-Timestamp (55)—from AAA • Tunnel-Type (64)—from client • Tunnel-Medium-Type (65)—from client • Tunnel-Client-Endpoint (66)—from client • Tunnel-Server-Endpoint (67)—from client • Acct-Tunnel-Connection (68)—from client
Tunnel-Stop	10	Marks the end of a tunnel connection to or from another node.	• User-Name (1)—from client • NAS-IP-Address (4)—from AAA • Acct-Delay-Time (41)—from AAA • Acct-Input-Octets (42)—from AAA • Acct-Output-Octets (43)—from AAA • Acct-Session-Id (44)—from AAA • Acct-Session-Time (46)—from AAA • Acct-Input-Packets (47)—from AAA • Acct-Output-Packets (48)—from AAA • Acct-Terminate-Cause (49)—from AAA • Acct-Multi-Session-Id (51)—from AAA • Event-Timestamp (55)—from AAA • Tunnel-Type (64)—from client • Tunnel-Medium-Type (65)—from client • Tunnel-Client-Endpoint (66)—from client • Tunnel-Server-Endpoint (67)—from client • Acct-Tunnel-Connection (68)—from client • Acct-Tunnel-Packets-Lost (86)—from client

Table 17 RADIUS Accounting Types for the Acct-Status-Type Attribute (continued)

Type-Name	Number	Description	Additional Attributes[1]
Tunnel-Reject	11	Marks the rejection of a tunnel setup with another node.	• User-Name (1)—from client • NAS-IP-Address (4)—from AAA • Acct-Delay-Time (41)—from AAA • Acct-Terminate-Cause (49)—from client • Event-Timestamp (55)—from AAA • Tunnel-Type (64)—from client • Tunnel-Medium-Type (65)—from client • Tunnel-Client-Endpoint (66)—from client • Tunnel-Server-Endpoint (67)—from client • Acct-Tunnel-Connection (68)—from client
Tunnel-Link-Start	12	Marks the creation of a tunnel link. Only some tunnel types (Layer 2 Transport Protocol [L2TP]) support the multiple links per tunnel; this value should be included only in accounting packets for tunnel types that support multiple links per tunnel.	• User-Name (1)—from client • NAS-IP-Address (4)—from AAA • NAS-Port (5)—from AAA • Acct-Delay-Time (41)—from AAA • Event-Timestamp (55)—from AAA • Tunnel-Type (64)—from client • Tunnel-Medium-Type (65)—from client • Tunnel-Client-Endpoint (66)—from client • Tunnel-Server-Endpoint (67)—from client • Acct-Tunnel-Connection (68)—from client

Table 17 RADIUS Accounting Types for the Acct-Status-Type Attribute (continued)

Type-Name	Number	Description	Additional Attributes[1]
Tunnel-Link-Stop	13	Marks the end of a tunnel link. Only some tunnel types (L2TP) support the multiple links per tunnel; this value should be included only in accounting packets for tunnel types that support multiple links per tunnel.	• User-Name (1)—from client • NAS-IP-Address (4)—from AAA • NAS-Port (5)—from AAA • Acct-Delay-Time (41)—from AAA • Acct-Input-Octets (42)—from AAA • Acct-Output-Octets (43)—from AAA • Acct-Session-Id (44)—from AAA • Acct-Session-Time (46)—from AAA • Acct-Input-Packets (47)—from AAA • Acct-Output-Packets (48)—from AAA • Acct-Terminate-Cause (49)—from AAA • Acct-Multi-Session-Id (51)—from AAA • Event-Timestamp (55)—from AAA • NAS-Port-Type (61)—from AAA • Tunnel-Type (64)—from client • Tunnel-Medium-Type (65)—from client • Tunnel-Client-Endpoint (66)—from client • Tunnel-Server-Endpoint (67)—from client • Acct-Tunnel-Connection (68)—from client • Acct-Tunnel-Packets-Lost (86)—from client
Tunnel-Link-Reject	14	Marks the rejection of a tunnel setup for a new link in an existing tunnel. Only some tunnel types (L2TP) support the multiple links per tunnel; this value should be included only in accounting packets for tunnel types that support multiple links per tunnel.	• User-Name (1)—from client • NAS-IP-Address (4)—from AAA • Acct-Delay-Time (41)—from AAA • Acct-Terminate-Cause (49)—from AAA • Event-Timestamp (55)—from AAA • Tunnel-Type (64)—from client • Tunnel-Medium-Type (65)—from client • Tunnel-Client-Endpoint (66)—from client • Tunnel-Server-Endpoint (67)—from client • Acct-Tunnel-Connection (68)—from client

1. If the specified tunnel type is used, these attributes should also be included in the accounting request packet.

How to Configure RADIUS Tunnel Accounting

This section contains the following procedures

- Enabling Tunnel Type Accounting Records, page 264
- Verifying RADIUS Tunnel Accounting, page 266

Enabling Tunnel Type Accounting Records

Use this task to configure your LAC to send tunnel and tunnel-link accounting records to be sent to the RADIUS server.

VPDN Tunnel Events

Two new command line interfaces (CLIs)—vpdn session accounting network (tunnel-link-type records) and vpdn tunnel accounting network (tunnel-type records)—are supported to help identify the following events:

- A VPDN tunnel is brought up or destroyed
- A request to create a VPDN tunnel is rejected
- A user session within a VPDN tunnel is brought up or brought down
- A user session create request is rejected

Note The first two events are tunnel-type accounting records: authentication, authorization, and accounting (AAA) sends Tunnel-Start, Tunnel-Stop, or Tunnel-Reject accounting records to the RADIUS server. The next two events are tunnel-link-type accounting records: AAA sends Tunnel-Link-Start, Tunnel-Link-Stop, or Tunnel-Link-Reject accounting records to the RADIUS server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa accounting network** {**default** | *list-name*} {**start-stop** | **stop-only** | **wait-start** | none} **group** *groupname*
4. **vpdn enable**
5. **vpdn tunnel accounting network** *list-name*
6. **vpdn session accounting network** *list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# aaa accounting network {default | list-name} {start-stop | stop-only | wait-start | none} group groupname`	Enables network accounting. • **default**—If the default network accounting method-list is configured and no additional accounting configurations are enabled on the interface, network accounting is enabled by default. If either the **vpdn session accounting network** command or the **vpdn tunnel accounting network** command is linked to the **default** method-list, all tunnel and tunnel-link accounting records are enabled for those sessions. • *list-name*—The *list-name* defined in the **aaa accounting** command must be the same as the *list-name* defined in the VPDN command; otherwise, accounting will not occur.
Step 4	`Router(config)# vpdn enable`	Enables virtual private dialup networking on the router and informs the router to look for tunnel definitions in a local database and on a remote authorization server (if applicable).
Step 5	`Router(config)# vpdn tunnel accounting network list-name`	Enables Tunnel-Start, Tunnel-Stop, and Tunnel-Reject accounting records. • *list-name*—The *list-name* must match the *list-name* defined in the **aaa accounting** command; otherwise, network accounting will not occur.
Step 6	`Router(config)# vpdn session accounting network list-name`	Enables Tunnel-Link-Start, Tunnel-Link-Stop, and Tunnel-Link-Reject accounting records. • *list-name*—The *list-name* must match the *list-name* defined in the **aaa accounting** command; otherwise, network accounting will not occur.

What To Do Next

After you have enabled RADIUS tunnel accounting, you can verify your configuration via the following optional task "Verifying RADIUS Tunnel Accounting."

Verifying RADIUS Tunnel Accounting

Use either one or both of the following optional steps to verify your RADIUS tunnel accounting configuration.

SUMMARY STEPS

1. **enable**
2. **show accounting**
3. **show vpdn** [**session** | **tunnel**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`Router# show accounting`	Displays the active accountable events on the network and helps collect information in the event of a data loss on the accounting server.
Step 3	`Router# show vpdn [session] [tunnel]`	Displays information about active L2TP tunnel and message identifiers in a VPDN. • **session**—Displays a summary of the status of all active tunnels. • **tunnel**—Displays information about all active L2TP tunnels in summary-style format.

Configuration Examples for RADIUS Tunnel Accounting

This section provides the following configuration examples:

- Configuring RADIUS Tunnel Accounting on LAC: Example, page 266
- Configuring RADIUS Tunnel Accounting on LNS: Example, page 268

Configuring RADIUS Tunnel Accounting on LAC: Example

The following example shows how to configure your L2TP access concentrator (LAC) to send tunnel and tunnel-link accounting records to the RADIUS server:

```
aaa new-model
!
!
aaa authentication ppp default group radius
aaa authorization network default local
aaa accounting network m1 start-stop group radius
aaa accounting network m2 stop-only group radius
```

```
aaa session-id common
enable secret 5 $1$IDjH$iL7puCja1RMlyOM.JAeuf/
enable password lab
!
username ISP_LAC password 0 tunnelpass
!
!
resource-pool disable
!
!
ip subnet-zero
ip cef
no ip domain-lookup
ip host dirt 171.69.1.129
!
vpdn enable
vpdn tunnel accounting network m1
vpdn session accounting network m1
vpdn search-order domain dnis
!
vpdn-group 1
 request-dialin
  protocol l2tp
  domain cisco.com
 initiate-to ip 10.1.26.71
 local name ISP_LAC
!
isdn switch-type primary-5ess
!
!
fax interface-type fax-mail
mta receive maximum-recipients 0
!
controller T1 7/4
 framing esf
 linecode b8zs
 pri-group timeslots 1-24
!
!
!
interface FastEthernet0/0
 ip address 10.1.27.74 255.255.255.0
 no ip mroute-cache
 duplex half
 speed auto
 no cdp enable
!
interface FastEthernet0/1
 no ip address
 no ip mroute-cache
 shutdown
 duplex auto
 speed auto
 no cdp enable
!
interface Serial7/4:23
 ip address 60.0.0.2 255.255.255.0
 encapsulation ppp
 dialer string 2000
 dialer-group 1
 isdn switch-type primary-5ess
 ppp authentication chap
!
```

```
interface Group-Async0
 no ip address
 shutdown
 group-range 1/00 3/107
!
ip default-gateway 10.1.27.254
ip classless
ip route 0.0.0.0 0.0.0.0 10.1.27.254
no ip http server
ip pim bidir-enable
!
!
dialer-list 1 protocol ip permit
no cdp run
!
!
radius-server host 172.19.192.26 auth-port 1645 acct-port 1646 key rad123
radius-server retransmit 3
call rsvp-sync
!
```

Configuring RADIUS Tunnel Accounting on LNS: Example

The following example shows how to configure your L2TP network server (LNS) to send tunnel and tunnel-link accounting records to the RADIUS server:

```
aaa new-model
!
!
aaa accounting network m1 start-stop group radius
aaa accounting network m2 stop-only group radius
aaa session-id common
enable secret 5 $1$ftf.$wE6Q5Yv6hmQiwL9pizPCg1
!
username ENT_LNS password 0 tunnelpass
username user1@cisco.com password 0 lab
username user2@cisco.com password 0 lab
spe 1/0 1/7
 firmware location system:/ucode/mica_port_firmware
spe 2/0 2/9
 firmware location system:/ucode/mica_port_firmware
!
!
resource-pool disable
clock timezone est 2
!
ip subnet-zero
no ip domain-lookup
ip host CALLGEN-SECURITY-V2 64.24.80.28 3.47.0.0
ip host dirt 171.69.1.129
!
vpdn enable
vpdn tunnel accounting network m1
vpdn session accounting network m1
!
vpdn-group 1
accept-dialin
  protocol l2tp
  virtual-template 1
 terminate-from hostname ISP_LAC
 local name ENT_LNS
!
```

```
isdn switch-type primary-5ess
!
!
!
!
!
!
!
fax interface-type modem
mta receive maximum-recipients 0
!
interface Loopback0
 ip address 70.0.0.101 255.255.255.0
!
interface Loopback1
 ip address 80.0.0.101 255.255.255.0
!
interface Ethernet0
 ip address 10.1.26.71 255.255.255.0
 no ip mroute-cache
 no cdp enable
!
interface Virtual-Template1
 ip unnumbered Loopback0
 peer default ip address pool vpdn-pool1
 ppp authentication chap
!
interface Virtual-Template2
 ip unnumbered Loopback1
 peer default ip address pool vpdn-pool2
 ppp authentication chap
!
interface FastEthernet0
 no ip address
 no ip mroute-cache
 shutdown
 duplex auto
speed auto
 no cdp enable
!
ip local pool vpdn-pool1 70.0.0.1 70.0.0.100
ip local pool vpdn-pool2 80.0.0.1 80.0.0.100
ip default-gateway 10.1.26.254
ip classless
ip route 0.0.0.0 0.0.0.0 10.1.26.254
ip route 90.1.1.2 255.255.255.255 10.1.26.254
no ip http server
ip pim bidir-enable
!
!
dialer-list 1 protocol ip permit
no cdp run
!
!
radius-server host 172.19.192.80 auth-port 1645 acct-port 1646 key rad123
radius-server retransmit 3
call rsvp-sync
```

Additional References

The following sections provide references related to RFC-2867 RADIUS Tunnel Accounting.

Related Documents

Related Topic	Document Title
RADIUS attributes	The appendix "RADIUS Attributes" in the *Cisco IOS Security Configuration Guide*
Vpdn	The chapter "Configuring Virtual Private Networks" in the *Cisco IOS Dial Technologies Configuration Guide*
Network accounting	The chapter "Configuring Accounting" in the *Cisco IOS Security Configuration Guide*

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2867	*RADIUS Accounting Modifications for Tunnel Protocol Support*

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **aaa accounting**
- **vpdn session accounting network**
- **vpdn tunnel accounting network**

RADIUS Attribute Screening

Feature History

Release	Modification
12.2(1)DX	This feature was introduced.
12.2(2)DD	This feature was integrated into Cisco IOS Release 12.2(2)DD.
12.2(4)B	This feature was integrated into Cisco IOS Release 12.2(4)B.
12.2(4)T	This feature was integrated into 12.2(4)T.
12.2(13)T	Platform support was added for the Cisco 7401 ASR router.

This feature module describes the RADIUS Attribute Screening feature in Cisco IOS Release 12.2(13)T. It includes the following sections:

- Feature Overview, page 273
- Supported Platforms, page 275
- Supported Standards, MIBs, and RFCs, page 276
- Prerequisites, page 276
- Configuration Tasks, page 276
- Configuration Examples, page 278
- Command Reference, page 280
- Glossary, page 281

Feature Overview

The RADIUS Attribute Screening feature allows users to configure a list of “accept” or “reject” RADIUS attributes on the network access server (NAS) for purposes such as authorization or accounting.

If a NAS accepts and processes *all* RADIUS attributes received in an Access-Accept packet, unwanted attributes may be processed, creating a problem for wholesale providers who do not control their customers’ authentication, authorization, and accounting (AAA) servers. For example, there may be attributes that specify services to which the customer has not subscribed, or there may be attributes that may degrade service for other wholesale dial users. The ability to configure the NAS to restrict the use of specific attributes has therefore become a requirement for many users.

The RADIUS Attribute Screening feature should be implemented in one of the following ways:

- To allow the NAS to accept and process all standard RADIUS attributes for a particular purpose, except for those on a configured reject list
- To allow the NAS to reject (filter out) all standard RADIUS attributes for a particular purpose, except for those on a configured accept list

Benefits

The RADIUS Attribute Screening feature provides the following benefits:

- Users can configure an accept or reject list consisting of a selection of attributes on the NAS for a specific purpose so unwanted attributes are not accepted and processed.
- Users may wish to configure an accept list that includes only relevant accounting attributes, thereby reducing unnecessary traffic and allowing users to customize their accounting data.

Restrictions

NAS Requirements

To enable this feature, your NAS should be configured for authorization with RADIUS groups.

Accept or Reject Lists Limitations

The two filters used to configure accept or reject lists are mutually exclusive; therefore, a user can configure only one access list or one reject list for each purpose, per server group.

Vendor-Specific Attributes

This feature does not support vendor-specific attribute (VSA) screening; however, a user can specify attribute 26 (Vendor-Specific) in an accept or reject list, which will accept or reject all VSAs.

Required Attributes Screening Recommendation

It is recommended that users do not reject the following required attributes:

- For authorization:
  - 6 (Service-Type)
  - 7 (Framed-Protocol)
- For accounting:
  - 4 (NAS-IP-Address)
  - 40 (Acct-Status-Type)
  - 41 (Acct-Delay-Time)
  - 44 (Acct-Session-ID)

If an attribute is required, the rejection will be refused, and the attribute will be allowed to pass through.

Note The user will not receive an error at the point of configuring a reject list for required attributes because the list does not specify a purpose—authorization or accounting. The server will determine whether an attribute is required when it is known what the attribute is to be used for.

Related Documents

- *Cisco IOS Security Command Reference*, Release 12.2
- *Cisco IOS Security Configuration Guide*, Release 12.2

Supported Platforms

For Cisco IOS Releases 12.2(1)DX, 12.2(2)DD, 12.2(4)B, 12.2(4)T, and 12.2(13)T

- Cisco 7200 series

For Cisco IOS Release 12.2(13)T Only

- Cisco 7401 ASR

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that are supported on specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

No new or modified RFCs are supported by this feature.

Prerequisites

Before configuring a RADIUS accept or reject list, you must enable AAA.

For more information, refer to the AAA chapters in the *Cisco IOS Security Configuration Guide*, Release 12.2.

Configuration Tasks

See the following section for configuration tasks for the RADIUS Attribute Screening feature. Each task in the list is identified as either optional or required.

- Configuring RADIUS Attribute Screening (required)
- Verifying RADIUS Attribute Screening (optional)

Configuring RADIUS Attribute Screening

To configure a RADIUS attribute accept or reject list for authorization or accounting, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 4	`Router(config)# aaa authentication ppp default group group-name`	Specifies one or more AAA authentication methods for use on serial interfaces running PPP.
Step 5	`Router(config)# aaa authorization network default group group-name`	Sets parameters that restrict network access to the user.
Step 6	`Router(config)# aaa group server radius group-name`	Groups different RADIUS server hosts into distinct lists and distinct methods.
Step 7	`Router(config-sg-radius)# server ip-address`	Configures the IP address of the RADIUS server for the group server,
Step 8	`Router(config-sg-radius)# authorization [accept | reject] listname` and/or `Router(config-sg-radius)# accounting [accept | reject] listname`	Specifies a filter for the attributes that are returned in an Access-Accept packet from the RADIUS server. and/or Specifies a filter for the attributes that are to be sent to the RADIUS server in an accounting request. **Note** The **accept** keyword indicates that all attributes will be rejected except for the attributes specified in the *listname*. The **reject** keyword indicates that all attributes will be accepted except for the attributes specified in the *listname* and all standard attributes.
Step 9	`Router(config-sg-radius)# exit`	Exits server-group configuration mode.
Step 10	`Router(config)# radius-server host {hostname | ip-address} [key string]`	Specifies a RADIUS server host.
Step 11	`Router(config)# radius-server attribute list listname`	Defines the list name given to the set of attributes defined in the **attribute** command. **Note** The *listname* must be the same as the *listname* defined in Step 5.
Step 12	`Router(config-sg-radius)# attribute value1 [value2 [value3...]]`	Adds attributes to the configured accept or reject list. **Note** This command can be used multiple times to add attributes to an accept or reject list.

Verifying RADIUS Attribute Screening

To verify an accept or reject list, use one of the following commands in privileged EXEC mode:

Command	Purpose
`Router# debug aaa accounting`	Displays information on accountable events as they occur.
`Router# debug aaa authentication`	Displays information on AAA authentication.
`Router# show radius statistics`	Displays the RADIUS statistics for accounting and authentication packets.

Configuration Examples

This section provides the following configuration examples:

- Authorization Accept Example
- Accounting Reject Example
- Authorization Reject and Accounting Accept Example
- Rejecting Required Attributes Example

Authorization Accept Example

The following example shows how to configure an accept list for attribute 6 (Service-Type) and attribute 7 (Framed-Protocol); all other attributes (including VSAs) are rejected for RADIUS authorization.

```
aaa new-model
aaa authentication ppp default group radius-sg
aaa authorization network default group radius-sg
aaa group server radius radius-sg
    server 1.1.1.1
    authorization accept min-author
!
radius-server host 1.1.1.1 key mykey1
radius-server attribute list min-author
    attribute 6-7
```

Accounting Reject Example

The following example shows how to configure a reject list for attribute 66 (Tunnel-Client-Endpoint) and attribute 67 (Tunnel-Server-Endpoint); all other attributes (including VSAs) are accepted for RADIUS accounting.

```
aaa new-model
aaa authentication ppp default group radius-sg
aaa authorization network default group radius-sg
aaa group server radius radius-sg
    server 1.1.1.1
    accounting reject tnl-x-endpoint
!
radius-server host 1.1.1.1 key mykey1
radius-server attribute list tnl-x-endpoint
    attribute 66-67
```

Authorization Reject and Accounting Accept Example

The following example shows how to configure a reject list for RADIUS authorization and configure an accept list for RADIUS accounting. Although you cannot configure more than one accept or reject list per server group for authorization or accounting, you can configure one list for authorization and one list for accounting per server group.

```
aaa new-model
aaa authentication ppp default group radius-sg
aaa authorization network default group radius-sg
aaa group server radius radius-sg
    server 1.1.1.1
    authorization reject bad-author
    accounting accept usage-only
!
radius-server host 1.1.1.1 key mykey1
radius-server attribute list usage-only
    attribute 1,40,42-43,46
!
radius-server attribute list bad-author
    attribute 22,27-28,56-59
```

Rejecting Required Attributes Example

The following example shows debug output for the **debug aaa accounting** command. In this example, required attributes 44, 40, and 41 have been added to the reject list “standard.”

```
Router# debug aaa authorization

AAA/ACCT(6): Accounting method=radius-sg (radius)
RADIUS: attribute 44 cannot be rejected
RADIUS: attribute 61 rejected
RADIUS: attribute 31 rejected
RADIUS: attribute 40 cannot be rejected
RADIUS: attribute 41 cannot be rejected
```

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **accounting (server-group configuration)**
- **authorization (server-group configuration)**
- **attribute (server-group configuration)**
- **radius-server attribute list**

Glossary

AAA—authentication, authorization, and accounting. Suite of network security services that provide the primary framework through which access control can be set up on your Cisco router or access server.

attribute—RADIUS Internet Engineering Task Force (IETF) attributes are the original set of 255 standard attributes that are used to communicate AAA information between a client and a server. Because IETF attributes are standard, the attribute data is predefined and well known; thus all clients and servers who exchange AAA information via IETF attributes must agree on attribute data such as the exact meaning of the attributes and the general bounds of the values for each attribute.

NAS—network access server. A Cisco platform (or collection of platforms, such as an AccessPath system) that interfaces between the packet world (for example, the Internet) and the circuit world (for example, the Public Switched Telephone Network).

RADIUS—Remote Authentication Dial-In User Service. RADIUS is a distributed client/server system that secures networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco routers and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.

VSA—vendor-specific attribute. VSAs are derived from one IETF attribute—vendor-specific (attribute 26). Attribute 26 allows a vendor to create and implement an additional 255 attributes. That is, a vendor can create an attribute that does not match the data of any IETF attribute and encapsulate it behind attribute 26: essentially, Vendor-Specific ="protocol:attribute=value".

RADIUS Centralized Filter Management

Feature History for Radius Centralized Filter Management

Release	Modification
12.2(13)T	This feature was introduced.
12.2(27)SB	This feature was integrated into Cisco IOS Release 12.2(27)SB.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Feature Overview, page 284
- Supported Standards, MIBs, and RFCs, page 285
- Prerequisites, page 286
- Configuration Tasks, page 286
- Monitoring and Maintaining the Filter Cache, page 288
- Configuration Examples, page 288
- Command Reference, page 290

Feature Overview

Before the RADIUS Centralized Filter Management feature, wholesale providers (who provide premium charges for customer services such as access control lists [ACLs]) were unable to prevent customers from applying exhaustive ACLs, which could impact router performance and other customers. This feature introduces a centralized administration point—a filter server—for ACL management. The filter server acts as a centralized RADIUS repository for ACL configuration.

Whether or not the RADIUS server that is used as the filter server is the same server that is used for access authentication, the network access server (NAS) will initiate a second access request to the filter server. If configured, the NAS will use the filter-ID name as the authentication username and the filter server password for the second access request. The RADIUS server will attempt to authenticate the filter-ID name, returning any required filtering configuration in the access-accept response.

Because downloading ACLs is time consuming, a local cache is maintained on the NAS. If an ACL name exists on the local cache, that configuration will be used without consulting the filter server.

Note An appropriately configured cache should minimize delays; however, the first dialin user to require a filter will always experience a longer delay because the ACL configuration is retrieved for the first time.

Cache Management

A global filter cache is maintained on the NAS of recently downloaded ACLs; thus, users no longer have to repeatedly request the same ACL configuration information from a potentially overloaded RADIUS server. Users are required to flush the cache when the following criteria have been met:

- After an entry becomes associated with a newly active call, the idle timer that is associated with that entry will be reset, if configured to do so.
- After the idle-time stamp of an entry expires, the entry will be removed.
- After the global cache of entries reaches a specified maximum number, the entry whose idle-timer is closest to the idle time limit will be removed.

A single timer is responsible for managing all cache entries. The timer is started after the first cache entry is created, and it runs periodically until reboot. The period of the timer will correspond to the minimum granularity offered when configuring cache idle timers, which is one expiration per minute. A single timer prevents users from having to manage individual timers per cache entry.

Note The single timer introduces a lack of precision in timer expiration. There is an average error of approximately 50 percent of the timer granularity. Although decreasing the timer granularity will decrease the average error, the decreased timer granularity will negatively impact performance. Because precise timing is not required for cache management, the error delay should be acceptable.

New Vendor-Specific Attribute Support

This feature introduces support for three new vendor-specific attributes (VSAs), which can be divided into the following two categories:

- User profile extensions

  - Filter-Required (50)—Specifies whether the call should be permitted if the specified filter is not found. If present, this attribute will be applied after any authentication, authorization, and accounting (AAA) filter method-list.
- Pseudo-user profile extensions
  - Cache-Refresh (56)—Specifies whether cache entries should be refreshed each time an entry is referenced by a new session. This attribute corresponds to the **cache refresh** command.
  - Cache-Time (57)—Specifies the idle time out, in minutes, for cache entries. This attribute corresponds to the **cache clear age** command.

Note All RADIUS attributes will override any command-line interface (CLI) configurations.

Benefits

This feature allows users to centrally manage filters at a RADIUS server, thereby, offloading ACL configuration and management to a centralized repository.

Restrictions

Multiple method lists are not supported in this feature; only a single global filter method list can be configured.

Related Documents

- The chapters “Configuring Authorization” and “Configuring RADIUS” in the *Cisco IOS Security Configuration Guide*, Release 12.2
- The chapter “Authorization Commands” in the *Cisco IOS Security Command Reference*, Release 12.2

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

None

Prerequisites

- You may need to add a dictionary file to your server if it does not support the new RADIUS VSAs. For a sample dictionary and vendors file, see the section "RADIUS Dictionary and Vendors File Example" later in this document.

  If you need to add a dictionary file, ensure that your RADIUS server is nonstandard and that it can send the newly introduced VSAs.
- You want to set up RADIUS network authentication so a remote user can dial in and get IP connectivity.

Configuration Tasks

See the following sections for configuration tasks for the Centralized Filter Management feature. Each task in the list is identified as either required or optional.

- Configuring the RADIUS ACL Filter Server (required)
- Configuring the Filter Cache (required)
- Verifying the Filter Cache (optional)

Configuring the RADIUS ACL Filter Server

To enable the RADIUS ACL filter server, use the following command in global configuration mode:

Command	Purpose
`Router(config)# aaa authorization cache filterserver default methodlist[methodlist2...]`	Enables AAA authorization caches and the downloading of an ACL configuration from a RADIUS filter server. • **default**—The default authorization list. • *methodlist* [*methodlist2...*]—One of the keywords listed on the password command page.

Configuring the Filter Cache

To configure the filter cache, use the following commands beginning in global configuration:

	Command	Purpose
Step 1	`Router(config)# aaa cache filter`	Enables filter cache configuration and enters AAA filter configuration mode.
Step 2	`Router(config-aaa-filter)# password {0 | 7} password`	(Optional) Specifies the optional password that is to be used for filter server authentication requests. **0**—Specifies that an unencrypted password will follow. **7**—Specifies that a hidden password will follow. *password*—The unencrypted (clear text) password. **Note** If a password is not specified, the default password ("cisco") is enabled.
Step 3	`Router(config-aaa-filter)# cache disable`	(Optional) Disables the cache.
Step 4	`Router(config-aaa-filter)# cache clear age minutes`	(Optional) Specifies, in minutes, when cache entries expire and the cache is cleared. *minutes*—Any value between 0 to 4294967295. **Note** If a time is not specified, the default (1400 minutes [1 day]) is enabled.
Step 5	`Router(config-aaa-filter)# cache refresh`	(Optional) Refreshes a cache entry when a new session begins. This command is enabled by default. To disable this functionality, use the **no cache refresh** command.
Step 6	`Router(config-aaa-filter)# cache max number`	(Optional) Limits the absolute number of entries the cache can maintain for a particular server. *number*—The maximum number of entries the cache can contain. Any value between 0 to 4294967295. **Note** If a number is not specified, the default (100 entries) is enabled.

Verifying the Filter Cache

To display the cache status, use the **show aaa cache filterserver** EXEC command. The following is sample output for the **show aaa cache filterserver** command:

```
Router# show aaa cache filterserver

Filter     Server          Age Expires Refresh Access-Control-Lists
-------------------------------------------------------------------------------
aol        1.2.3.4           0    1440     100 ip in icmp drop
                                               ip out icmp drop
                                               ip out forward tcp dstip 1.2.3...
msn        1.2.3.4         N/A   Never       2 ip in tcp drop
msn2       1.2.3.4         N/A   Never       2 ip in tcp drop
vone       1.2.3.4         N/A   Never       0 ip in tcp drop
```

Note The **show aaa cache filterserver** command shows how many times a particular filter has been referenced or refreshed. This function may be used in administration to determine which filters are actually being used.

Troubleshooting Tips

To help troubleshoot your filter cache configurations, use the privileged EXEC **debug aaa cache filterserver** command. To view sample output for the **debug aaa cache filterserver** command, refer to the section "Debug Output Example" later in this document.

Monitoring and Maintaining the Filter Cache

To monitor and maintain filter caches, use at least one of the following EXEC commands:

Command	Purpose
Router# **clear aaa cache filterserver acl** [*filter-name*]	Clears the cache status for a particular filter or all filters.
Router# **show aaa cache filterserver**	Displays the cache status.

Configuration Examples

This section provides the following configuration examples:

- NAS Configuration Example
- RADIUS Server Configuration Example
- RADIUS Dictionary and Vendors File Example
- Debug Output Example

NAS Configuration Example

The following example shows how to configure the NAS for cache filtering. In this example, the server group "mygroup" in contacted first. If there is no response, the default RADIUS server will then be contacted. If there still is no response, the local filters care contacted. Finally, the call is accepted if the filter cannot be resolved.

```
aaa authorization cache filterserver group mygroup group radius local none
!
aaa group server radius mygroup
 server 1.2.3.4
 server 1.2.3.5
!
radius-server host 1.2.3.4
!
aaa cache filter
 password mycisco
 no cache refresh
```

```
 cache max 100
!
```

RADIUS Server Configuration Example

The following example is a sample RADIUS configuration that is for a remote user "user1" dialing into the NAS:

```
myfilter Password = "cisco"
        Service-Type = Outbound,
        Ascend:Ascend-Call-Filter = "ip in drop srcip 7.0.0.1/32 dstip 7.0.0.10/32 icmp",
        Ascend:Ascend-Call-Filter = "ip in drop srcip 7.0.0.1/32 dstip 7.0.0.10/32 tcp
        dstport = telnet",
        Ascend:Ascend-Cache-Refresh = Refresh-No,
        Ascend:Ascend-Cache-Time = 15

user1 Password = "cisco"
        Service-Type = Framed,
        Filter-Id = "myfilter",
        Ascend:Ascend-Filter-Required = Filter-Required-Yes,
```

RADIUS Dictionary and Vendors File Example

The following example is a sample RADIUS dictionary file for the new VSAs. In this example, the dictionary file is for a Merit server.

```
dictionary file:
Ascend.attr Ascend-Filter-Required 50 integer (*, 0, NOENCAPS)
Ascend.attr Ascend-Cache-Refresh   56 integer (*, 0, NOENCAPS)
Ascend.attr Ascend-Cache-Time      57 integer (*, 0, NOENCAPS)

Ascend.value Ascend-Cache-Refresh Refresh-No  0
Ascend.value Ascend-Cache-Refresh Refresh-Yes 1

Ascend.value Ascend-Filter-Required  Filter-Required-No  0
Ascend.value Ascend-Filter-Required  Filter-Required-Yes 1

vendors file:
50     50
56     56
57     57
```

Debug Output Example

The following is sample output from the **debug aaa cache filterserver** command:

```
Router# debug aaa cache filterserver

AAA/FLTSV: need "myfilter" (fetch), call 0x612DAC64
AAA/FLTSV: send req, call 0x612DAC50
AAA/FLTSV: method SERVER_GROUP myradius
AAA/FLTSV: recv reply, call 0x612DAC50 (PASS)
AAA/FLTSV: create cache
AAA/FLTSV: add attr "call-inacl"
AAA/FLTSV: add attr "call-inacl"
AAA/FLTSV: add attr "call-inacl"
```

```
AAA/FLTSV: skip attr "filter-cache-refresh"
AAA/FLTSV: skip attr "filter-cache-time"
AAA/CACHE: set "AAA filtserv cache" entry "myfilter" refresh? no
AAA/CACHE: set "AAA filtserv cache" entry "myfilter" cachetime 15
AAA/FLTSV: add attr to list "call-inacl" call 0x612DAC64
AAA/FLTSV: add attr to list "call-inacl" call 0x612DAC64
AAA/FLTSV: add attr to list "call-inacl" call 0x612DAC64
AAA/FLTSV: PASS call 0x612DAC64
AAA/CACHE: timer "AAA filtserv cache", next in 10 secs (0 entries)
AAA/CACHE: timer "AAA filtserv cache", next in 10 secs (1 entry)
AAA/CACHE: destroy "AAA filtserv cache" entry "myfilter"
AAA/CACHE: timer "AAA filtserv cache", next in 10 secs (0 entries)
```

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **aaa authorization cache filterserver**
- **aaa cache filter**
- **cache clear age**
- **cache disable**
- **cache max**
- **cache refresh**
- **clear aaa cache filterserver acl**
- **debug aaa cache filterserver**
- **password**
- **show aaa cache filterserver**

RADIUS Debug Enhancements

Feature History

Release	Modification
12.2(11)T	This feature was introduced on the Cisco 1400 series, Cisco 1600 series, Cisco 1700 series, Cisco 2500 series, Cisco 2600 series, Cisco 3600 series, Cisco 7100, Cisco 7200, Cisco 7500, Cisco AS5300, Cisco AS5800, Cisco Catalyst 5000, Cisco MC3810, and Cisco MGX8850 platforms.

This document describes the Remote Authentication Dial-In User Services (RADIUS) Debug Enhancements feature in Cisco IOS Release 12.2(11)T. It includes the following sections:

- Feature Overview, page 291
- Supported Platforms, page 292
- Supported Standards, MIBs, and RFCs, page 293
- Prerequisites, page 293
- Configuration Tasks, page 294
- Configuration Examples, page 297
- Command Reference, page 299
- Glossary, page 300

Feature Overview

This document details the RADIUS Debug Enhancements feature. RADIUS is a distributed client/server system that provides the following functionality:

- secures networks against unauthorized access
- enables authorization of specific service limits
- provides accounting information so that services can be billed

In the Cisco implementation, RADIUS clients run on Cisco routers and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.

The **debug radius** command displays information associated with RADIUS. Prior to the RADIUS Debug Enhancements feature, **debug radius** output was available only in an expanded, hexadecimal string format, resulting in displays that were difficult to interpret and analyze. Moreover, attribute value displays were truncated, particularly for vendor-specific attributes (VSAs).

The new feature provides enhanced RADIUS display including the following:

- Packet dump in a more readable, user-friendly ASCII format than before
- Complete display of attribute values without truncation
- Ability to select a brief RADIUS **debug** output display

Benefits

- Provides RADIUS debug display in a user-friendly format
- Supports complete RADIUS debug information
- Provides the default display of packet dump in ASCII format
- Allows a compact debugging output option that is useful for high-traffic, operational environments

Restrictions

Only Internet Engineering Task Force (IETF) attributes and Cisco VSAs used in voice applications are supported. For unsupported attributes, “undebuggable” is displayed.

Related Features and Technologies

- Cisco IOS security
- RADIUS authentication, authorization, and accounting (AAA)
- Cisco Voice over IP (VoIP)

Related Documents

- *Configuration Guide for AAA Billing Features in Cisco Voice-Enabled Routers and Access Servers*
- *Cisco IOS Security Configuration Guide*, Release 12.2, “Configuring RADIUS” chapter
- *RADIUS Vendor-Specific Attributes Voice Implementation Guide,* Release 12.1
- *Cisco IOS Debug Command Reference*, Release 12.2

Supported Platforms

- Cisco 1400 series
- Cisco 1600 series
- Cisco 1700 series
- Cisco 2500 series
- Cisco 2600 series
- Cisco 3600 series
- Cisco 7100 series

- Cisco 7200 series
- Cisco 7500 series
- Cisco AS5300
- Cisco AS5800
- Cisco Catalyst 5000
- Cisco MC3810
- Cisco MGX8850

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

No new or modified RFCs are supported by this feature.

Prerequisites

- Establish a working IP network. For more information about configuring IP, refer to the part “IP Overview,” and the “Configuring IP Addressing,” and “Configuring IP Services” chapters in the Cisco IOS Release 12.0 *Network Protocols Configuration Guide, Part 1*.
- Configure VoIP. For more information about configuring VoIP, refer to

Cisco IOS Voice, Video, and Fax Configuration Guide, Release 12.2.

- Configure the gateway as a RADIUS client. Refer to the chapter "Configuring the RADIUS Client Gateway" in the *RADIUS Vendor-Specific Attributes Voice Implementation Guide*, Release 12.1.

Configuration Tasks

See the following sections for configuration tasks for the RADIUS Debug Enhancements feature. Each task in the list is identified as either required or optional.

- Configuring Default Debug ASCII Display (optional)
- Configuring Debug Display in Brief Format (optional)
- Configuring Debug Display in Hex Format (optional)
- Verifying the debug radius Command (optional)

Configuring Default Debug ASCII Display

The complete ASCII format debug display with no truncation is enabled by default; no configuration tasks are required to enable this feature. To reenable the feature if it was disabled by using the **no debug radius** command, use the following command in privileged EXEC mode:

Command	Purpose
`Router# debug radius`	Enables RADIUS debugging output.

Note Prior to Cisco IOS Release 12.2(11)T, the **debug radius** command enabled truncated debugging output in hexadecimal notation, rather than ASCII. To enable debugging output in hex format, use the **debug radius hex** command.

Configuring Debug Display in Brief Format

Debugging output is available in a compact output that displays only basic information. To enable this display option, use the following command in privileged EXEC mode:

Command	Purpose
`Router# debug radius brief`	Enables RADIUS debugging output displaying only the client/server interaction and minimum packet information.

Configuring Debug Display in Hex Format

Debugging output is available in hexadecimal notation. To enable this display option, use the following command in privileged EXEC mode:

Command	Purpose
`Router# debug radius hex`	Enables RADIUS debugging output in hexadecimal notation.

Verifying the debug radius Command

Use the **show debug** command to verify RADIUS output options.

```
5300# show debug

Radius protocol debugging is on
Radius packet hex dump debugging is on
5300_43#
17:26:52: RADIUS: ustruct sharecount=3
17:26:52: Radius: radius_port_info() success=0 radius_nas_port=1
17:26:52: RADIUS: Initial Transmit ISDN 0:D:23 id 10 10.0.0.0:1824, Accounting-Request,
len 361
17:26:52:         Attribute 4 6 01081D03
17:26:52:         Attribute 26 19 00000009020D4953444E20303A443A3233
17:26:52:         Attribute 61 6 00000000
17:26:52:         Attribute 1 12 34303835323734323036
17:26:52:         Attribute 30 7 3532393831
17:26:52:         Attribute 31 12 34303835323734323036
17:26:52:         Attribute 40 6 00000001
17:26:52:         Attribute 6 6 00000001
17:26:52:         Attribute 26 27 0000000921156833323332D67772D69643D353330305F34332E
17:26:52:         Attribute 26 57
0000000901336833323332D696E636F6D696E672D636F6E662D69643D3846334133313633204234393830303046
20302033424537314238
17:26:52:         Attribute 26 31
000000091A19683332332D63616C6C2D6F726967696E3D616E73776572
17:26:52:         Attribute 26 32
000000091B1A683332332D63616C6C2D747970653D54656C6570686F6E79
17:26:52:         Attribute 26 56
0000000919326833323332D73657475702D74696D653D2A30393A32363A35322E383830205053542053617420204A
616E20312032303030
17:26:52:         Attribute 26 48
00000009182A683332332D636F6E662D69643D3846334133313633204234393830303046203020334245373142
38
17:26:52:         Attribute 44 10 3030303030303035
17:26:52:         Attribute 41 6 00000000
17:26:52: %ISDN-6-CONNECT: Interface Serial0:22 is now connected to 4085274206
17:26:52: RADIUS: Received from id 10 1.7.157.1:1824, Accounting-response, len 20
17:27:01: RADIUS: ustruct sharecount=3
17:27:01: Radius: radius_port_info() success=0 radius_nas_port=1
17:27:01: RADIUS: Initial Transmit ISDN 0:D:23 id 11 10.0.0.1:1823, Access-Request, len
173
17:27:01:         Attribute 4 6 01081D03
17:27:01:         Attribute 26 19 00000009020D4953444E20303A443A3233
17:27:01:         Attribute 61 6 00000000
17:27:01:         Attribute 1 8 313233343536
17:27:01:         Attribute 26 48
00000009182A683332332D636F6E662D69643D3846334133313633204234393830303046203020334245373142
38
```

```
17:27:01:         Attribute 31 12 34303835323734323036
17:27:01:         Attribute 2 18 C980D8D0E9A061B3D783C61AA6F27214
17:27:01:         Attribute 26 36
00000009011E683332332D6976722D6F75743D7472616E73616374696F6E49443A33
17:27:01: RADIUS: Received from id 11 1.7.157.1:1823, Access-Accept, len 115
17:27:01:         Attribute 6 6 00000001
17:27:01:         Attribute 26 29 0000000965176833323332D6372656469742D616D6F756E743D3435
17:27:01:         Attribute 26 27 000000096615683332332D6372656469742D74696D653D3333
17:27:01:         Attribute 26 26 00000009671468333233 2D72657475726E2D636F64653D30
17:27:01:         Attribute 25 7 6C6F63616C
17:27:01: RADIUS: saved authorization data for user 61AA0698 at 6215087C
17:27:09: %ISDN-6-DISCONNECT: Interface Serial0:22  disconnected from 4085554206 , call
lasted 17 seconds
17:27:09: RADIUS: ustruct sharecount=2
17:27:09: Radius: radius_port_info() success=0 radius_nas_port=1
17:27:09: RADIUS: Sent class "local" at 621508E8 from user 61AA0698
17:27:09: RADIUS: Initial Transmit ISDN 0:D:23 id 12 10.0.0.0:1824, Accounting-Request,
len 776
17:27:09:         Attribute 4 6 01081D03
17:27:09:         Attribute 26 19 00000009020D4953444E20303A443A3233
17:27:09:         Attribute 61 6 00000000
17:27:09:         Attribute 1 8 313233343536
17:27:09:         Attribute 30 7 3532393831
17:27:09:         Attribute 31 12 34303835323734323036
17:27:09:         Attribute 40 6 00000002
17:27:09:         Attribute 25 7 6C6F63616C
17:27:09:         Attribute 45 6 00000001
17:27:09:         Attribute 6 6 00000001
17:27:09:         Attribute 26 27 000000092115683332332D67772D69643D353330305F34332E
17:27:09:         Attribute 26 57
000000090133683332332D696E636F6D696E672D636F6E662D69643D3846334133313633204234393830303046
203020334245373142 38
17:27:09:         Attribute 26 31
000000091A19683332332D63616C6C2D6F726967696E3D616E73776572
17:27:09:         Attribute 26 32
000000091B1A683332332D63616C6C2D747970653D54656C6570686F6E79
17:27:09:         Attribute 26 56
000000091932683332332D73657475702D74696D653D2A30393A32363A35322E383830205053542053617420536174204A
616E20312032303030
17:27:09:         Attribute 26 58
000000091C34683332332D636F6E6E6563742D74696D653D2A30393A32363A35322E393037205053542053617420536174
204A616E20312032303030
17:27:09:         Attribute 26 61
000000091D37683332332D646973636F6E6E6E6563742D74696D653D2A30393A32373A31302E3133372050535420
536174204A616E20312032303030
17:27:09:         Attribute 26 32
000000091E1A683332332D646973636F6E6E6E6563742D63617573653D3130
17:27:09:         Attribute 26 28 000000091F16683332332D766F6963652D7175616C6974793D30
17:27:09:         Attribute 26 48
00000009182A683332332D636F6E662D69643D3846334133313633204234393830303046203020334245373142
38
17:27:09:         Attribute 44 10 3030303030303035
17:27:09:         Attribute 42 6 00000000
17:27:09:         Attribute 43 6 00012CA0
17:27:09:         Attribute 47 6 00000000
17:27:09:         Attribute 48 6 000001E1
17:27:09:         Attribute 46 6 00000011
17:27:09:         Attribute 26 30 0000000901187375627363726962657 23D526567756C61724C696E65
17:27:09:         Attribute 26 35
00000009011D683332332D6976722D6F75743D5461726966663A556E6B6E6F776E
17:27:09:         Attribute 26 22 0000000901107072652D62797465732D696E3D30
17:27:09:         Attribute 26 23 0000000901117072652D62797465732D6F75743D30
17:27:09:         Attribute 26 21 00000009010F7072652D70616B732D696E3D30
17:27:09:         Attribute 26 22 0000000901107072652D70616B732D6F75743D30
```

```
17:27:09:         Attribute 26 22 0000000901106E61732D72782D73706565643D30
17:27:09:         Attribute 26 22 0000000901106E61732D74782D73706565643D30
17:27:09:         Attribute 41 6 00000000
17:27:09: RADIUS: Received from id 12 10.0.0.0:1824, Accounting-response, len 20
```

Configuration Examples

This section provides the following configuration examples:

- Default debug radius Command Example
- Compact Debugging Output Example

Default debug radius Command Example

The following sample output shows the default RADIUS output in ASCII notation, generated by the **debug radius** command:

```
router# debug radius

Radius protocol debugging is on
Radius packet hex dump debugging is off
router#
00:02:50: RADIUS: ustruct sharecount=3
00:02:50: Radius: radius_port_info() success=0 radius_nas_port=1
00:02:50: RADIUS: Initial Transmit ISDN 0:D:23 id 0 10.0.0.0:1824, Accounting-Request, len
358
00:02:50: RADIUS: NAS-IP-Address [4] 6 10.0.0.1
00:02:50: RADIUS: Vendor, Cisco [26] 19 VT=02 TL=13 ISDN 0:D:23
00:02:50: RADIUS: NAS-Port-Type [61] 6 Async
00:02:50: RADIUS: User-Name [1] 12 "4085274206"
00:02:50: RADIUS: Called-Station-Id [30] 7 "52981"
00:02:50: RADIUS: Calling-Station-Id [31] 12 "4085554206"
00:02:50: RADIUS: Acct-Status-Type [40] 6 Start
00:02:50: RADIUS: Service-Type [6] 6 Login
00:02:50: RADIUS: Vendor, Cisco [26] 27 VT=33 TL=21 h323-gw-id=5300_43.
00:02:50: RADIUS: Vendor, Cisco [26] 55 VT=01 TL=49 h323-incoming-conf-id=8F3A3163
B4980003 0 29BD0
00:02:50: RADIUS: Vendor, Cisco [26] 31 VT=26 TL=25 h323-call-origin=answer
00:02:50: RADIUS: Vendor, Cisco [26] 32 VT=27 TL=26 h323-call-type=Telephony
00:02:50: RADIUS: Vendor, Cisco [26] 57 VT=25 TL=51 h323-setup-time=*16:02:48.681 PST Fri
Dec 31 1999
00:02:50: RADIUS: Vendor, Cisco [26] 46 VT=24 TL=40 h323-conf-id=8F3A3163 B4980003 029BD0
00:02:50: RADIUS: Acct-Session-Id [44] 10 "00000002"
00:02:50: RADIUS: Delay-Time [41] 6 0
00:02:51: RADIUS: Received from id 0 10.0.0.0:1824, Accounting-response, len 20
00:02:51: %ISDN-6-CONNECT: Interface Serial0:22 is now connected to 4085554206
00:03:01: RADIUS: ustruct sharecount=3
00:03:01: Radius: radius_port_info() success=0 radius_nas_port=1
00:03:01: RADIUS: Initial Transmit ISDN 0:D:23 id 1 1.7.157.1:1823, Access-Request, len
171
00:03:01: RADIUS: NAS-IP-Address [4] 6 10.0.0.1
00:03:01: RADIUS: Vendor, Cisco [26] 19 VT=02 TL=13 ISDN 0:D:23
00:03:01: RADIUS: NAS-Port-Type [61] 6 Async
00:03:01: RADIUS: User-Name [1] 8 "123456"
00:03:01: RADIUS: Vendor, Cisco [26] 46 VT=24 TL=40 h323-conf-id=8F3A3163 B4980003 0 29BD0
00:03:01: RADIUS: Calling-Station-Id [31] 12 "4085274206"
00:03:01: RADIUS: User-Password [2] 18 *
00:03:01: RADIUS: Vendor, Cisco [26] 36 VT=01 TL=30 h323-ivr-out=transactionID:0
```

```
00:03:01: RADIUS: Received from id 1 1.7.157.1:1823, Access-Accept, len 115
00:03:01: RADIUS: Service-Type [6] 6 Login
00:03:01: RADIUS: Vendor, Cisco [26] 29 VT=101 TL=23 h323-credit-amount=45
00:03:01: RADIUS: Vendor, Cisco [26] 27 VT=102 TL=21 h323-credit-time=33
00:03:01: RADIUS: Vendor, Cisco [26] 26 VT=103 TL=20 h323-return-code=0
00:03:01: RADIUS: Class [25] 7 6C6F63616C
00:03:01: RADIUS: saved authorization data for user 62321E14 at 6233D258
00:03:13: %ISDN-6-DISCONNECT: Interface Serial0:22 disconnected from 4085274206, call
lasted 22 seconds
00:03:13: RADIUS: ustruct sharecount=2
00:03:13: Radius: radius_port_info() success=0 radius_nas_port=1
00:03:13: RADIUS: Sent class "local" at 6233D2C4 from user 62321E14
00:03:13: RADIUS: Initial Transmit ISDN 0:D:23 id 2 10.0.0.0:1824, Accounting-Request, len
775
00:03:13: RADIUS: NAS-IP-Address [4] 6 10.0.0.1
00:03:13: RADIUS: Vendor, Cisco [26] 19 VT=02 TL=13 ISDN 0:D:23
00:03:13: RADIUS: NAS-Port-Type [61] 6 Async
00:03:13: RADIUS: User-Name [1] 8 "123456"
00:03:13: RADIUS: Called-Station-Id [30] 7 "52981"
00:03:13: RADIUS: Calling-Station-Id [31] 12 "4085554206"
00:03:13: RADIUS: Acct-Status-Type [40] 6 Stop
00:03:13: RADIUS: Class [25] 7 6C6F63616C
00:03:13: RADIUS: Undebuggable [45] 6 00000001
00:03:13: RADIUS: Service-Type [6] 6 Login
00:03:13: RADIUS: Vendor, Cisco [26] 27 VT=33 TL=21 h323-gw-id=5300_43.
00:03:13: RADIUS: Vendor, Cisco [26] 55 VT=01 TL=49 h323-incoming-conf-id=8F3A3163
B4980003 0 29BD0
00:03:13: RADIUS: Vendor, Cisco [26] 31 VT=26 TL=25 h323-call-origin=answer
00:03:13: RADIUS: Vendor, Cisco [26] 32 VT=27 TL=26 h323-call-type=Telephony
00:03:13: RADIUS: Vendor, Cisco [26] 57 VT=25 TL=51 h323-setup-time=*16:02:48.681 PST Fri
Dec 31 1999
00:03:13: RADIUS: Vendor, Cisco [26] 59 VT=28 TL=53 h323-connect-time=*16:02:48.946
PST Fri Dec 31 1999
00:03:13: RADIUS: Vendor, Cisco [26] 62 VT=29 TL=56 h323-disconnect-time=*16:03:11.306
PST Fri Dec 31 1999
00:03:13: RADIUS: Vendor, Cisco [26] 32 VT=30 TL=26 h323-disconnect-cause=10
00:03:13: RADIUS: Vendor, Cisco [26] 28 VT=31 TL=22 h323-voice-quality=0
00:03:13: RADIUS: Vendor, Cisco [26] 46 VT=24 TL=40 h323-conf-id=8F3A3163 B4980003 0 29BD0
00:03:13: RADIUS: Acct-Session-Id [44] 10 "00000002"
00:03:13: RADIUS: Acct-Input-Octets [42] 6 0
00:03:13: RADIUS: Acct-Output-Octets [43] 6 88000
00:03:13: RADIUS: Acct-Input-Packets [47] 6 0
00:03:13: RADIUS: Acct-Output-Packets [48] 6 550
00:03:13: RADIUS: Acct-Session-Time [46] 6 22
00:03:13: RADIUS: Vendor, Cisco [26] 30 VT=01 TL=24 subscriber=RegularLine
00:03:13: RADIUS: Vendor, Cisco [26] 35 VT=01 TL=29 h323-ivr-out=Tariff:Unknown
00:03:13: RADIUS: Vendor, Cisco [26] 22 VT=01 TL=16 pre-bytes-in=0
00:03:13: RADIUS: Vendor, Cisco [26] 23 VT=01 TL=17 pre-bytes-out=0
00:03:13: RADIUS: Vendor, Cisco [26] 21 VT=01 TL=15 pre-paks-in=0
00:03:13: RADIUS: Vendor, Cisco [26] 22 VT=01 TL=16 pre-paks-out=0
00:03:13: RADIUS: Vendor, Cisco [26] 22 VT=01 TL=16 nas-rx-speed=0
00:03:13: RADIUS: Vendor, Cisco [26] 22 VT=01 TL=16 nas-tx-speed=0
00:03:13: RADIUS: Delay-Time [41] 6 0
00:03:13: RADIUS: Received from id 2 10.0.0.0:1824, Accounting-response, len 20
```

Compact Debugging Output Example

A new EXEC command, **debug radius brief,** enables this abbreviated output option. The following sample output displays only the client/server interaction and minimum packet information (packet type, ID and so forth).

```
router# debug radius brief

Radius protocol debugging is on
Radius packet hex dump debugging is off
Radius protocol in brief format debugging is on
00:05:21: RADIUS: Initial Transmit ISDN 0:D:23 id 6 10.0.0.1:1824, Accounting-Request, len
358
00:05:21: %ISDN-6-CONNECT: Interface Serial0:22 is now connected to 4085554206
00:05:26: RADIUS: Retransmit id 6
00:05:31: RADIUS: Tried all servers.
00:05:31: RADIUS: No valid server found. Trying any viable server
00:05:31: RADIUS: Tried all servers.
00:05:31: RADIUS: No response for id 7
00:05:31: RADIUS: Initial Transmit ISDN 0:D:23 id 8 10.0.0.0:1823, Access-Request, len 171
00:05:36: RADIUS: Retransmit id 8
00:05:36: RADIUS: Received from id 8 10.0.0.0:1823, Access-Accept, len 115
00:05:47: %ISDN-6-DISCONNECT: Interface Serial0:22 disconnected from 4085554206, call
lasted 26 seconds
00:05:47: RADIUS: Initial Transmit ISDN 0:D:23 id 9 10.0.0.1:1824, Accounting-Request, len
775
00:05:47: RADIUS: Received from id 9 1.7.157.1:1824, Accounting-response, len 20
```

Command Reference

The following modified command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **debug radius**

Glossary

AAA — authentication, authorization, and accounting. Pronounced “triple a.”

ASCII — American Standard Code for Information Interchange. 8-bit code for character representation (7 bits plus parity).

attribute — Form of information items provided by the X.500 Directory Service. The directory information base consists of entries, each containing one or more attributes. Each attribute consists of a type identifier together with one or more values.

IETF — Internet Engineering Task Force. Task force consisting of over 80 working groups responsible for developing Internet standards. The IETF operates under the auspices of ISOC.

RADIUS — Remote Authentication Dial-In User Service. Database for authenticating modem and ISDN connections and for tracking connection time.

VoIP — Voice over IP. The capability to carry normal telephony-style voice over an IP-based internet with POTS-like functionality, reliability, and voice quality. VoIP enables a router to carry voice traffic (for example, telephone calls and faxes) over an IP network. In VoIP, the DSP segments the voice signal into frames, which then are coupled in groups of two and stored in voice packets. These voice packets are transported using IP in compliance with ITU-T specification H.323.

VSA — vendor-specific attribute. An attribute that has been implemented by a particular vendor. It uses the attribute Vendor-Specific to encapsulate the resulting AV pair: essentially, Vendor-Specific = protocol:attribute = value.

RADIUS Logical Line ID

Feature History

Release	Modification
12.2(13)T	This feature was introduced.
12.2(15)B	This feature was integrated into Cisco IOS Release 12.2(15)B.

This document describes the RADIUS Logical Line ID feature in Cisco IOS Release 12.2(15)B. It includes the following sections:

- Feature Overview, page 301
- Supported Platforms, page 302
- Supported Standards, MIBs, and RFCs, page 303
- Configuration Tasks, page 304
- Configuration Examples, page 305
- Command Reference, page 306

Feature Overview

The RADIUS Logical Line ID feature enables users to track their customers on the basis of the physical lines in which the customers' calls originate. Thus, users can better maintain the profile database of their customers as the customers move from one physical line to another.

Logical Line Identification (LLID) is an alphanumeric string (which must be a minimum of one character and a maximum of 253 characters) that is a logical identification of a subscriber line. LLID is maintained in a RADIUS server customer profile database. This customer profile database is connected to a L2TP access concentrator (LAC) and is separate from the RADIUS server that the LAC and L2TP Network Server (LNS) use for the authentication and authorization of incoming users. When the customer profile database receives a preauthorization request from the LAC, the server sends the LLID to the LAC as the Calling-Station-ID attribute (attribute 31).

The LAC sends a preauthorization request to the customer profile database when the LAC is configured for preauthorization. Configure the LAC for preauthorization using the **subscriber access pppoe pre-authorize** command.

Note Downloading the LLID is referred to as preauthorization because it occurs before normal virtual private dialup network (VPDN) authorization downloads L2TP tunnel information.

The customer profile database consists of user profiles for each user connected to the LAC. Each user profile contains the NAS-IP-Address (attribute #4) and the NAS-Port-ID (attribute #5.) When the LAC is configured for preauthorization, it queries the customer profile database using the username. (The username, which is in an authentication, authorization, and accounting (AAA) request, has physical line information.) When a match is found in the customer profile database, the customer profile database sends the LLID in the user profile. The LLID is defined in the username as the Calling-Station-ID attribute.

Benefits

Stability and Security

This feature provides users with a virtual port that will not change as customers move. Thus, the LLID can also be used for additional security checks.

Restrictions

RADIUS Server Compatibility

Although this feature can be used with any vendor's RADIUS server, some RADIUS servers may require modifications to their dictionary files to allow the Calling-Station-ID attribute to be returned in access-accept messages. For example, the Merit RADIUS server will not support LLID downloading unless you modify its dictionary as follows: “ATTRIBUTE Calling-Station-Id 31 string (*, *)”

Support Restrictions

- This feature supports only RADIUS; TACACS+ is not supported.
- This feature can be applied only toward PPP over Ethernet over ATM (PPPoEoATM) and PPP over Ethernet over VLAN (PPPoEoVLAN) (Dot1Q) calls; no other calls, such as ISDN, can be used.

Related Documents

- The chapter “Configuring Broadband Access: PPP and Routed Bridge Encapsulation” in the *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.2
- The section “Configuring AAA Preauthentication” in the chapter “Configuring RADIUS” in the *Cisco IOS Security Configuration Guide*, Release 12.2
- *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2

Supported Platforms

For platforms supported in Cisco IOS Release 12.2(13)T and 12.2(15)B, consult Cisco Feature Navigator.

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that are supported on specific platforms. To obtain updated information about platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. In the release section, you can compare releases side by side to display both the features unique to each software release and the features that releases in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL: http://www.cisco.com/register

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

None

Configuration Tasks

See the following sections for configuration tasks for the RADIUS Logical Line ID feature. Each task in the list is identified as either required or optional.

- Configuring Preauthorization (required)
- Configuring the LLID in a RADIUS User Profile (required)
- Verifying Logical Line ID (optional)

Configuring Preauthorization

To download the LLID and configure the LAC for preauthorization, use the following commands in global configuration mode:

Command	Purpose
`Router (config)# ip radius source-interface interface-name`	Specifies the IP address portion of the username for the preauthorization request.
`Router (config)# subscriber access pppoe pre-authorize nas-port-id list-name`	Enables the LLID to be downloaded so the LAC can be configured for preauthorization.

Configuring the LLID in a RADIUS User Profile

To configure the user profile for preauthorization, add a NAS port user to the customer profile database and add the RADIUS IETF attribute 31 (Calling-Station-ID) to the user profile.

Command	Purpose
`UserName=nas_port: ip-address:slot/module/port/vpi.vci`	(Optional) Adds a PPPoE over ATM NAS port user.
`User-Name=nas-port: ip-address:slot/module/port/vlan-id`	(Optional) Adds a PPPoE over VLAN NAS port user.
`Calling-Station-Id = "string (*,*)"`	Adds attribute 31 to the user profile. string—One or more octets, containing the phone number that the user placed the call from.

Verifying Logical Line ID

To verify feature functionality, use the following command in EXEC mode:

Command	Purpose
`Router# debug radius`	Checks to see that RADIUS attribute 31 is the LLID in the accounting-request on LAC and in the access-request and accounting-request on the LNS.

Configuration Examples

This section provides the following configuration examples:

- LAC for Preauthorization Configuration Example
- RADIUS User Profile for LLID Example

LAC for Preauthorization Configuration Example

The following example shows how to configure your LAC for preauthorization by downloading the LLID:

```
aaa new-model
aaa group server radius sg_llid
 server 128.107.164.106 auth-port 1645 acct-port 1646
aaa group server radius sg_water
 server 128.107.164.106 auth-port 1645 acct-port 1646
aaa authentication ppp default group radius
aaa authorization confg-commands
aaa authorization network default group sg_water
aaa authorization network mlist_llid group sg_llid
aaa session-id common
!
username s7200_2 password 0 lab
username s5300 password 0 lab
username sg_water password 0 lab
vpdn enable
!
vpdn-group 2
  request-dialin
  protocol 12tp
 domain water.com
 initiate-to ip 30.1.1.1
 local name s7200_2
!
vpdn-group 3
 accept dialin
  procotol pppoe
  virtual-template 1
!
! Enable the LLID to be downloaded.
subscriber access pppoe pre-authorize nas-port-id mlist_llid
!
interface Loopback0
 ip address 20.1.1.2 255.255.255.0
!
```

```
interface Loopback1
 ip address 30.1.1.1 255.255.255.0
!
interface Ethernet1/0
 ip address 80.1.1.1 255.255.255.0 secondary
 ip address 10.0.58.111 255.255.255.0
 no cdp enable
!
iterface ATM4/0
 no ip address
 no atm ilmi-keepalive
!
interface ATM4/0.1 point-to-point
 pvc 1/100
  encapsulation aa15snap
  protocol pppoe
!
interface virtual-template1
 no ip unnumbered Loopback0
 no peer default ip address
 ppp authentication chap
!
radius-server host 128.107.164.120 auth-port 1645 acct-port 1646 key rad123
radius-server host 128.107.164.106 auth-port 1645 acct-port 1646 key rad123
ip radius source-interface Loopback1
```

RADIUS User Profile for LLID Example

The following example shows how to configure the user profile for LLID querying for PPPoEoVLAN and PPPoEoATM and add attribute 31:

```
pppoeovlan
----------
nas-port:12.1.0.3:6/0/0/0    Password = "cisco",
    Service-Type = Outbound,
    Calling-Station-ID = "cat-example"

pppoeoa
--------
nas-port:12.1.0.3:6/0/0/1.100    Password = "cisco",
    Service-Type = Outbound,
    Calling-Station-ID = "cat-example"
```

Command Reference

This section documents the new command. All other commands used with this feature are documented in the Cisco IOS Release 12.2 command reference publications.

- **subscriber access pppoe pre-authorize**

RADIUS NAS-IP-Address Attribute Configurability

The RADIUS NAS-IP-Address Attribute Configurability feature allows you to configure an arbitrary IP address to be used as RADIUS attribute 4, NAS-IP-Address, without changing the source IP address in the IP header of the RADIUS packets. This feature may be used for situations in which service providers are using a cluster of small network access servers (NASs) to simulate a large NAS to improve scalability. This feature allows the NASs to behave as a single RADIUS client from the perspective of the RADIUS server.

Feature History for RADIUS NAS-IP-Address Attribute Configurability

Release	Modification
12.3(3)B	This feature was introduced.
12.3(7)T	This feature was integrated into Cisco IOS Release 12.3(7)T.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for RADIUS NAS-IP-Address Attribute Configurability, page 308
- Restrictions for RADIUS NAS-IP-Address Attribute Configurability, page 308
- Information About RADIUS NAS-IP-Address Attribute Configurability, page 308
- How to Configure RADIUS NAS-IP-Address Attribute Configurability, page 309
- Configuration Examples for RADIUS NAS-IP-Address Attribute Configurability, page 311
- Additional References, page 312
- Command Reference, page 313

Prerequisites for RADIUS NAS-IP-Address Attribute Configurability

- You must be familiar with IP Security (IPSec).
- You must be familiar with configuring both RADIUS servers and authentication, authorization, and accounting (AAA). Before configuring this feature, you must first set up the RADIUS server and the AAA lists.

Restrictions for RADIUS NAS-IP-Address Attribute Configurability

The following restrictions apply if a cluster of RADIUS clients are being used to simulate a single RADIUS client for scalability. Solutions, or workarounds, to the restrictions are also provided.

- RADIUS attribute 44, Acct-Session-Id, may overlap among sessions from different NASs.

  There are two solutions. Either the **radius-server attribute 44 extend-with-addr** or **radius-server unique-ident** command can be used on NAS routers to specify different prepending numbers for different NAS routers.
- RADIUS server-based IP address pool for different NASs must be managed.

  The solution is to configure different IP address pool profiles for different NASs on the RADIUS server. Different NASs use different pool usernames to retrieve them.
- RADIUS request message for sessions from different NASs must be differentiated.

  One of the solutions is to configure different format strings for RADIUS attribute 32, NAS-Identifier, using the **radius-server attribute 32 include-in-access-req** command on different NASs.

Information About RADIUS NAS-IP-Address Attribute Configurability

To configure the RADIUS NAS-IP-Address Attribute Configurability feature, you should understand the following concepts:

- Problem Definition and Solution Background Information, page 308
- Using the RADIUS NAS-IP-Address Attribute Configurability Feature, page 309

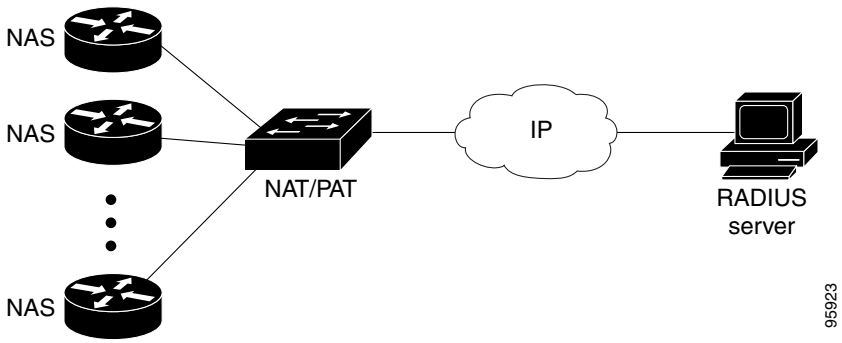
Problem Definition and Solution Background Information

To simulate a large NAS RADIUS client using a cluster of small NAS RADIUS clients, as shown in Figure 13, a Network Address Translation (NAT) or Port Address Translation (PAT) device is inserted in a network. The device is placed between a cluster of NASs and the IP cloud that is connected to a RADIUS server. When RADIUS traffic from different NASs goes through the NAT or PAT device, the source IP addresses of the RADIUS packets are translated to a single IP address, most likely an IP address on a loopback interface on the NAT or PAT device. Different User Datagram Protocol (UDP)

source ports are assigned to RADIUS packets from different NASs. When the RADIUS reply comes back from the server, the NAT or PAT device receives it, uses the destination UDP port to translate the destination IP address back to the IP address of the NAS, and forwards the reply to the corresponding NAS.

Figure 13 demonstrates how the source IP addresses of several NASs are translated to a single IP address as they pass through the NAT or PAT device on the way to the IP cloud.

Figure 13 NAS Addresses Translated to a Single IP Address



RADIUS servers normally check the source IP address in the IP header of the RADIUS packets to track the source of the RADIUS requests and to maintain security. The NAT or PAT solution satisfies these requirements because only a single source IP address is used even though RADIUS packets come from different NAS routers.

However, when retrieving accounting records from the RADIUS database, some billing systems use RADIUS attribute 4, NAS-IP-Address, in the accounting records. The value of this attribute is recorded on the NAS routers as their own IP addresses. The NAS routers are not aware of the NAT or PAT that runs between them and the RADIUS server; therefore, different RADIUS attribute 4 addresses will be recorded in the accounting records for users from the different NAS routers. These addresses eventually expose different NAS routers to the RADIUS server and to the corresponding billing systems.

Using the RADIUS NAS-IP-Address Attribute Configurability Feature

The RADIUS NAS-IP-Address Attribute Configurability feature allows you to freely configure an arbitrary IP address as RADIUS NAS-IP-Address, RADIUS attribute 4. By manually configuring the same IP address, most likely the IP address on the loopback interface of the NAT or PAT device, for all the routers, you can hide a cluster of NAS routers behind the NAT or PAT device from the RADIUS server.

How to Configure RADIUS NAS-IP-Address Attribute Configurability

This section contains the following procedures:

- Configuring a RADIUS NAS-IP-Address Attribute Configurability, page 310
- Monitoring and Maintaining RADIUS NAS-IP-Address Attribute Configurability, page 310

Configuring a RADIUS NAS-IP-Address Attribute Configurability

Before configuring the RADIUS NAS-IP-Address Attribute Configurability feature, you must have configured the RADIUS servers or server groups and AAA method lists.

To configure the RADIUS NAS-IP-Address Attribute Configurability feature, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius-server attribute 4** *ip-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**radius-server attribute 4** *ip-address* **Example:** `Router (config)# radius-server attribute 4 10.2.1.1`	Configures an IP address to be used as the RADIUS NAS-IP-Address, attribute 4.

Monitoring and Maintaining RADIUS NAS-IP-Address Attribute Configurability

To monitor the RADIUS attribute 4 address that is being used inside the RADIUS packets, use the **debug radius** command.

SUMMARY STEPS

1. **enable**
2. **debug radius**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug radius** **Example:** `Router# debug radius`	Displays information associated with RADIUS.

Examples

The following sample output is from the **debug radius** command:

```
Router# debug radius

RADIUS/ENCODE(0000001C): acct_session_id: 29
RADIUS(0000001C): sending
RADIUS(0000001C): Send Access-Request to 10.0.0.10:1645 id 21645/17, len 81
RADIUS:  authenticator D0 27 34 C0 F0 C4 1C 1B - 3C 47 08 A2 7E E1 63 2F
RADIUS:  Framed-Protocol     [7]   6   PPP                       [1]
RADIUS:  User-Name           [1]   18  "shashi@pepsi.com"
RADIUS:  CHAP-Password       [3]   19  *
RADIUS:  NAS-Port-Type       [61]  6   Virtual                   [5]
RADIUS:  Service-Type        [6]   6   Framed                    [2]
RADIUS:  NAS-IP-Address      [4]   6   10.0.0.21
UDP: sent src=10.1.1.1(21645), dst=10.0.0.10(1645), length=109
UDP: rcvd src=10.0.0.10(1645), dst=10.1.1.1(21645), length=40
RADIUS: Received from id 21645/17 10.0.0.10:1645, Access-Accept, len 32
RADIUS:  authenticator C6 99 EC 1A 47 0A 5F F2 - B8 30 4A 4C FF 4B 1D F0
RADIUS:  Service-Type        [6]   6   Framed                    [2]
RADIUS:  Framed-Protocol     [7]   6   PPP                       [1]
RADIUS(0000001C): Received from id 21645/17
```

Configuration Examples for RADIUS NAS-IP-Address Attribute Configurability

This section provides the following configuration example:

- Configuring a RADIUS NAS-IP-Address Attribute Configurability: Example, page 311

Configuring a RADIUS NAS-IP-Address Attribute Configurability: Example

The following example shows that IP address 10.0.0.21 has been configured as the RADIUS NAS-IP-Address attribute:

```
radius-server attribute 4 10.0.0.21
radius-server host 10.0.0.10 auth-port 1645 acct-port 1646 key cisco
```

Additional References

The following sections provide references related to RADIUS NAS-IP-Address Attribute Configurability.

Related Documents

Related Topic	Document Title
Configuring AAA	"Authentication, Authorization, and Accounting (AAA)" section of *Cisco IOS Security Configuration Guide*
Configuring RADIUS	"Configuring RADIUS" chapter of *Cisco IOS Security Configuration Guide*
RADIUS commands	*Cisco IOS Security Command Reference*, Release 12.3 T
Other security commands	*Cisco IOS Security Command Reference*, Release 12.3 T

Standards

Standards	Title
No new or modified standards are supported by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **radius-server attribute 4**

RADIUS Route Download

Feature History

Release	Modification
12.2(8)T	This feature was introduced.
12.2(27)SB	This feature was integrated into Cisco IOS Release 12.2(27)SB.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Feature Overview, page 315
- Supported Platforms, page 316
- Supported Standards, MIBs, and RFCs, page 317
- Prerequisites, page 317
- Configuration Tasks, page 317
- Configuration Examples, page 318
- Command Reference, page 318

Feature Overview

The RADIUS Route Download feature allows users to configure their network access server (NAS) to send static route download requests to authorization, authentication, and accounting (AAA) servers specified by a named method list. Before this feature, RADIUS authorization for static route download requests could be sent only to AAA servers specified by the default method list.

This feature extends the functionality of the command to allow users to specify the name of the method list that will be used to direct static route download requests to the AAA servers. The **aaa route download** command may be used to specify a separate method list for downloading static routes. This method list can be added by using the **aaa authorization configuration** command.

Benefits

The RADIUS Route Download feature allows users to specify a separate method list for static route download requests; that is, the NAS can direct RADIUS authorization for static route download requests to servers specified by a method list in addition to the default method list.

Related Documents

- The chapter "AAA Overview" in the *Cisco IOS Security Configuration Guide*, Release 12.2
- The chapter "Configuring Large-Scale Dial-Out" in the *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2
- *Cisco IOS Dial Technologies Command Reference*, Release 12.2

Supported Platforms

- Cisco 800 series
- Cisco 805
- Cisco 806
- Cisco 820
- Cisco 828
- Cisco 1400 series
- Cisco 1600 series
- Cisco 1751
- Cisco 2420
- Cisco 2600 series
- Cisco 3620
- Cisco 3631
- Cisco 3640
- Cisco 3660
- Cisco 3725
- Cisco 3745
- Cisco 7100 series
- Cisco 7200 series
- Cisco 7500 series
- Cisco 7700 series
- Cisco CVA120
- Cisco MC3810
- Cisco uBR7200 series
- Route Processor Module (RPM)

- Universal Route Module (URM)

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

None

Prerequisites

AAA network security must be enabled before you perform the tasks in this feature. For information about enabling AAA, refer to the AAA section in the *Cisco IOS Security Configuration Guide*, Release 12.2.

Configuration Tasks

See the following sections for configuration tasks for the RADIUS Route Download feature. Each task in the list is identified as either required or optional.

- Configuring RADIUS Route Download (required)
- Verifying RADIUS Route Download (optional)

Configuring RADIUS Route Download

To configure the NAS to send static route download requests to the servers specified by a named method list, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa authorization configuration` *method-name* `[radius | tacacs+ | group` *group-name*`]`	Downloads static route configuration information from the AAA server using RADIUS.
Step 2	`Router(config)# aaa route download` *[time]* `[authorization` *method-list*`]`	Enables the static route download feature. Use the **authorization** *method-list* attributes to specify a named method list to which RADIUS authorization requests for static route downloads are sent.

Verifying RADIUS Route Download

To verify the routes that are installed, use the **show ip route** command in EXEC mode.

To display information that is associated with RADIUS, use the **debug radius** command in privileged EXEC mode.

Configuration Examples

This section provides the following configuration examples:

- RADIUS Route Download Configuration Example

RADIUS Route Download Configuration Example

The following example shows how to configure the NAS to send static route download requests to the servers specified by the method list named “foo”:

```
aaa new-model
aaa group server radius rad1
    server 2.2.2.2 auth-port 1645 acct-port 1646
!
aaa group server tacacs+ tac1
    server 3.3.3.3
!
aaa authorization configuration default group radius
aaa authorization configuration foo group rad1 group tac1
aaa route download 1 authorization foo

tacacs-server host 3.3.3.3
tacacs-server key cisco
tacacs-server administration
!
radius-server host 2.2.2.2 auth-port 1645 acct-port 1646
radius-server key cisco
```

Command Reference

The following modified command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **aaa route download**

RADIUS Support of 56-Bit Acct Session-Id

The RADIUS Support of 56-Bit Acct Session-Id feature introduces a new 32-bit authentication, authorization, and accounting (AAA) variable, acct-session-id-count. The first eight bits of the acct-session-id-count variable are reserved for the unique identifier variable, a unique number assigned to the accounting session which is preserved between reloads. The acct-session-id-count variable is used in addition to the existing 32-bit acct-session-id variable, RADIUS attribute 44, providing a total of 56 bits of to represent the actual Accounting Session Identifier (ID). Benefits of this feature include the following:

- The 8-bit unique identifier variable allows accounting sessionIDs to be identified if a reload occurs.
- The additional space provided by the acct-session-id-count variable can keep track of acct-session-id wrapping when there is a high volume of traffic, such as voice calls. By incrementing each time the acct-session-id variable wraps, the acct-session-id-count variable preserves accounting information.

Feature Specifications for RADIUS Support of 56-Bit Acct Session-Id

Feature History	
Release	**Modification**
12.3(2)T	This feature was introduced.
Supported Platforms	
Cisco AS5300, Cisco AS5350, Cisco AS5400, Cisco AS5800, Cisco AS5850	

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for RADIUS Support of 56-Bit Acct Session-Id, page 320
- Information About RADIUS Support of 56-Bit Acct Session-Id, page 320
- How to Configure RADIUS Support of 56-Bit Acct Session-Id, page 321
- Configuration Examples for RADIUS Support of 56-Bit Acct Session-Id, page 322

- Additional References, page 322
- Command Reference, page 323

Prerequisites for RADIUS Support of 56-Bit Acct Session-Id

AAA accounting must be configured. For more information about configuring AAA accounting, refer to the "Configuring Accounting" chapter in the *Cisco IOS Security Configuration Guide*, Release 12.2.

Information About RADIUS Support of 56-Bit Acct Session-Id

To configure the RADIUS Support of 56-bit Acct Session-Id feature, you must understand the following concepts:

- Acct-Session-Id Attribute, page 320
- Acct-Session-Id-Count Attribute, page 320
- Benefits of RADIUS Support of 56-Bit Acct Session-Id, page 321

Acct-Session-Id Attribute

RADIUS attribute 44, Accounting Session ID, is a unique accounting identifier that makes it easy to match start and stop records in a log file. Accounting session ID numbers restart at 1 each time the router is power-cycled or the software is reloaded. RADIUS attribute 44 is automatically enabled when AAA accounting is configured.

The acct-session-id variable is a 32-bit variable that can take on values from 00000000–FFFFFFFF.

Acct-Session-Id-Count Attribute

The new acct-session-id-count variable is a 32-bit variable. The first eight bits of the variable are reserved for the unique identifier variable, an identifier that allows the RADIUS server to identify an accounting session if a reload occurs. The remaining 24 bits of the acct-session-id-count variable acts as a counter variable. When the first acct-session-id variable is assigned, this counter variable is set to 1. The variable increments by 1 every time the acct-session-id variable wraps, preventing the loss of accounting information.

The acct-session-id-count variable can take on values from ##000000– ##FFFFFF, where ## represents the eight bits that are reserved for the unique identifier variable.

The acct-session-id-count and acct-session-id variables are concatenated before being sent to the RADIUS server, resulting in the acct-session variable being represented as the following:

##000000 00000000– ##FFFFFF FFFFFFFF

This allows a total of 56 bits to be used for acct-session-id space.

Benefits of RADIUS Support of 56-Bit Acct Session-Id

Allows RADIUS Servers to Identify Accounting Sessions After a Reload

The 8-bit unique identifier variable allows accounting session identities to be identified if a reload occurs.

Provides Acounting Information Space for High Volume Traffic

The additional space provided by the acct-session-id-count variable can keep track of acct-session-id wrapping when there is a high volume of traffic, such as voice calls. By incrementing each time the acct-session-id variable wraps, the acct-session-id-count variable preserves accounting information.

How to Configure RADIUS Support of 56-Bit Acct Session-Id

This section contains the following procedure:

- Configuring RADIUS Support of 56-Bit Acct Session-Id, page 321

Configuring RADIUS Support of 56-Bit Acct Session-Id

This task enables the acct-session-id-count variable containing the unique identifier variable.

SUMMARY STEPS

1. **enable**
2. **radius-server unique-ident** *id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**radius-server unique-ident** *id* **Example:** `Router(config)# radius-server unique-ident 5`	Enables the acct-session-id-count variable containing the unique identifier variable. • The *id* argument specifies the unique identifier represented by the first eight bits of the acct-session-id-count variable. Valid values range from 0 to 255.

Configuration Examples for RADIUS Support of 56-Bit Acct Session-Id

This section contains the following configuration example:

- Configuring RADIUS Support of 56-Bit Acct Session-Id Example, page 322

Configuring RADIUS Support of 56-Bit Acct Session-Id Example

The following example configures AAA authentication, enables RADIUS attribute 44 in access request packets, and enables the acct-session-id-count variable and sets the unique identifier variable to 5:

```
aaa new-model
aaa authentication ppp default group radius
radius-server host 10.100.1.34
radius-server unique-ident 5
```

Additional References

For additional information related to the RADIUS Support of 56-Bit Acct Session-Id feature, refer to the following references:

- Related Documents, page 322
- Standards, page 322
- MIBs, page 323
- RFCs, page 323
- Technical Assistance, page 323

Related Documents

Related Topic	Document Title
Additional information about configuring RADIUS	"Configuring RADIUS" chapter in the *Cisco IOS Security Configuration Guide*, Release 12.2
Additional information about configuring accounting	"Configuring Accounting" chapter in the *Cisco IOS Security Configuration Guide*, Release 12.2
Additional information about AAA RADIUS attributes	"RADIUS Attributes" section in the *Cisco IOS Security Configuration Guide*, Release 12.2
Additional RADIUS commands	The *Cisco IOS Security Command Reference*, Release 12.3 T

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

RFCs[1]	Title
RFC 2139	RADIUS Accounting

1. Not all supported RFCs are listed.

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **radius-server unique-ident**

RADIUS Tunnel Preference for Load Balancing and Fail-Over

Feature History

Release	Modification
12.2(4)T	This feature was introduced.

This document describes the RADIUS Tunnel Preference for Load Balancing and Fail-Over feature in Cisco IOS Release 12.2(4)T. It includes the following sections:

- Feature Overview, page 325
- Supported Platforms, page 328
- Supported Standards, MIBs, and RFCs, page 329
- Prerequisites, page 329
- Configuration Tasks, page 329
- Configuration Example, page 330
- Command Reference, page 330
- Glossary, page 331

Feature Overview

The RADIUS Tunnel Preference for Load Balancing and Fail-Over feature provides load balancing and fail-over virtual private dialup network (VPDN) home gateway (HGW) groups in a standardized fashion. This feature introduces new software functionality; no new command is associated with this feature.

Industry-Standard Rather Than Proprietary Attributes

Until Cisco IOS Release 12.2(4)T, load balancing and fail-over functionality for a Layer 2 Tunnel Protocol network server (LNS) was provided by the Cisco proprietary Vendor Specific Attribute (VSA). In a multivendor network environment, using VSA on a RADIUS server can cause interoperability issues among network access servers (NASs) manufactured by different vendors. Even though some RADIUS server implementations can send VSAs that the requesting NAS can understand, the user still must maintain different VSAs for the same purpose in a single-service profile.

A consensus regarding the tunnel attributes that are to be used in a multivendor network environment is defined in RFC 2868. In RFC 2868, Tunnel-Server-Endpoint, in conjunction with the Tunnel-Medium-Type, specifies the address to which the NAS should initiate a new session. If multiple Tunnel-Server-Endpoint attributes are defined in one tagged attribute group, they are interpreted as equal-cost load-balancing HGWs.

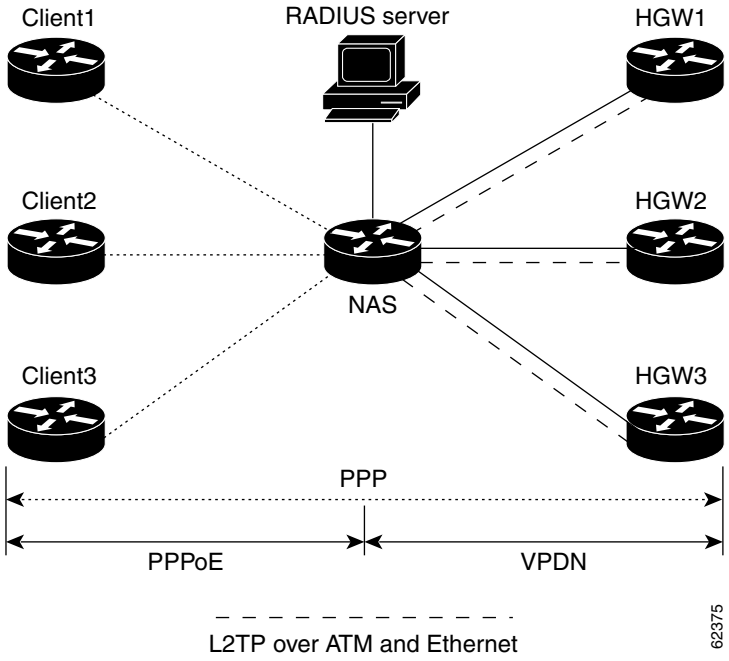
The Tunnel-Preference attribute defined in RFC 2868 can be used as a measure to form load balancing and fail-over HGW groups. When the Tunnel-Preference values of different tagged attribute groups are the same, the Tunnel-Server-Endpoint of those attribute groups is considered to have the same priority unless otherwise specified. When the Tunnel-Preference values of some attribute groups are higher (they have a lower preference) than other attribute groups, their Tunnel-Server-Endpoint attributes will have higher priority values. When an attribute group has a higher priority value, that attribute group will be used for fail-over in case the attribute groups with lower priority values are unavailable for the connections.

Until Cisco IOS Release 12.2(4)T, a specially formatted string would be transported within a Cisco VSA "vpdn:ip-addresses" string to a NAS for the purpose of HGW load balancing and fail-over. For example, 10.0.0.1 10.0.0.2 10.0.0.3/2.0.0.1 2.0.0.2 would be interpreted as IP addresses 10.0.0.1, 10.0.0.2, and 10.0.0.3 for the first group for load balancing. New sessions are projected to these three addresses based on the least-load-first algorithm. This algorithm uses its local knowledge to select an HGW that has the least load to initiate the new session. In this example, the addresses 2.0.0.1 and 2.0.0.2 in the second group have a lower priority and are applicable only when all HGWs specified in the first group fail to respond to the new connection request, thereby making 2.0.0.1 and 2.0.0.2 the fail-over addresses. See the section "Configuration Example" for an example of how to configure these fail-over addresses in a RADIUS tunnel profile.

Load Balancing and Fail-Over in a Multivendor Network

The RADIUS Tunnel Preference for Load Balancing and Fail-Over feature was designed for large multivendor networks that use VPDN Layer 2 tunnels over WAN links such as ATM and Ethernet, such as the configuration shown in Figure 14.

Figure 14 Typical Load Balancing and Fail-Over in a Multivendor Network



In the configuration shown in Figure 14, the NAS uses tunnel profiles downloaded from the RADIUS server to establish VPDN Layer 2 tunnels for load balancing and fail-over. The Point-to-Point over Ethernet (PPPoE) protocol is used as the client to generate PPP sessions.

Benefits

The RADIUS Tunnel Preference for Load Balancing and Fail-Over feature provides industry-standard load balancing and fail-over functionality for an LNS, rather than requiring the use of a Cisco proprietary VSA. The feature conforms to the tunnel attributes that are to be used in a multivendor network environment as defined in RFC 2868, thereby eliminating interoperability issues among NASs manufactured by different vendors.

Restrictions

The following restrictions and limitations apply to the RADIUS Tunnel Preference for Load Balancing and Fail-Over feature:

- This feature does not support VPDN dial-out networks; it is designed only for dial-in applications.
- The maximum number of LNSs allowed in the network is 1550, which is 50 per tag attribute group and a limit of 31 tags.
- This feature requires a RADIUS server implementation to support RFC 2868.

Related Features and Technologies

The RADIUS Tunnel Preference for Load Balancing and Fail-Over feature is used in VPDNs. Additionally, familiarity with the following technologies and protocols is recommended:

- ATM
- Ethernet
- L2TP and L2F
- PPP and PPPoE
- RADIUS servers

See the next section for a list of documentation that describes these technologies and protocols.

Related Documents

- "Basic Dial-in VPDN Configuration Using VPDN Groups" at http://www.cisco.com/warp/public/793/access_dial/2.html
- *Cisco IOS Dial Technologies Command Reference*, Release 12.2
- *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2, the chapters in the part "Virtual Templates, Profiles, and Networks"
- *Cisco IOS Security Command Reference*, Release 12.2
- *Cisco IOS Security Configuration Guide*, Release 12.2, the chapter "Configuring RADIUS" and the appendix "RADIUS Attributes"
- "Which VPN Solution is Right for You?" at http://www.cisco.com/warp/public/707/which_vpn.html
- *Cisco IOS Wide-Area Networking Command Reference*, Release 12.2
- *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.2, the chapter "Configuring Broadband Access: PPP and Routed Bridge Encapsulation"

Supported Platforms

This feature is platform independent and was either developed for or tested on the following Cisco routers:

- Cisco 800 series
- Cisco 1600 series
- Cisco 1700 series
- Cisco 2600 series
- Cisco 3600 series
- Cisco 7100 series
- Cisco 7200 series

See the next section for information about Feature Navigator and how to use this tool to determine the platforms and software images in which this feature is available.

Determining Platform Support Through Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Feature Navigator. Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image.

To access Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Feature Navigator is updated when major Cisco IOS software releases and technology releases occur. As of May 2001, Feature Navigator supports M, T, E, S, and ST releases. You can access Feature Navigator at the following URL:

http://www.cisco.com/go/fn

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFC 2868, *RADIUS Attributes for Tunnel Protocol Support*

Prerequisites

Configuring VPDNs and HGW groups is beyond the scope of this document. Refer to the documentation listed in the section "Related Documents" for the tasks and commands to configure these types of networks.

Configuration Tasks

This feature has no new configuration commands; however, see the next section for an example of how to implement the RADIUS Tunnel Preference for Load Balancing and Fail-Over feature in a RADIUS tunnel profile.

Configuration Example

The following example shows how to create RADIUS tunnel profiles:

```
net3 Password = "cisco" Service-Type = Outbound
        Tunnel-Type = :0:L2TP,
        Tunnel-Medium-Type = :0:IP,
        Tunnel-Server-Endpoint = :0:"1.1.3.1",
        Tunnel-Assignment-Id = :0:"1",
        Tunnel-Preference = :0:1,
        Tunnel-Password = :0:"welcome"

        Tunnel-Type = :1:L2TP,
        Tunnel-Medium-Type = :1:IP,
        Tunnel-Server-Endpoint = :1:"1.1.5.1",
        Tunnel-Assignment-Id = :1:"1",
        Tunnel-Preference = :1:1,
        Tunnel-Password = :1:"welcome"

        Tunnel-Type = :2:L2TP,
        Tunnel-Medium-Type = :2:IP,
        Tunnel-Server-Endpoint = :2:"1.1.4.1",
        Tunnel-Assignment-Id = :2:"1",
        Tunnel-Preference = :2:1,
        Tunnel-Password = :2:"welcome"

        Tunnel-Type = :3:L2TP,
        Tunnel-Medium-Type = :3:IP,
        Tunnel-Server-Endpoint = :3:"1.1.6.1",
        Tunnel-Assignment-Id = :3:"1",
        Tunnel-Preference = :3:1,
        Tunnel-Password = :3:"welcome"
```

The section "Feature Overview" describes how fail-over addresses are selected in these profiles. The section "Related Documents" lists documentation that describes how to create RADIUS tunnel profiles.

Command Reference

This feature uses no new or modified commands. To see the command pages for the commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

Glossary

HGW—home gateway. A gateway that terminates Layer 2 tunneling protocols such as L2TP.

home gateway—See HGW.

L2TP—Layer 2 Tunnel Protocol. An Internet Engineering Task Force (IETF) standards track protocol defined in RFC 2661 that provides tunneling of PPP. Based upon the best features of L2F and PPTP, L2TP provides an industry-wide interoperable method of implementing VPDN.

L2TP network server—See LNS.

Layer 2 Tunnel Protocol—See L2TP.

LNS—L2TP network server. A node that acts as one side of an L2TP tunnel endpoint and is a peer to the NAS or L2TP access concentrator (LAC). The LNS is the logical termination point of a PPP session that is being tunneled from the remote system by the access server. Analogous to the Layer 2 Forwarding (L2F) HGW.

NAS—network access server. Cisco platform or collection of platforms that interfaces between the packet world (the Internet, for example) and the circuit world (the public switched telephone network, for example).

network access server—See NAS.

Request for Comments—See RFCs.

RFCs—Request for Comments. A series of notes about the Internet collected by the Internet Engineering Task Force (IETF). Started in 1969, the IETF is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture. RFCs define many aspects of computer communication, focusing on networking protocols, procedures, programs, and concepts.

virtual private dialup network—See VPDN.

VPDN—virtual private dialup network. Enables IP traffic to travel securely over a public TCP/IP network by encrypting all traffic from one network to another.

RADIUS Server Reorder on Failure

During periods of high load or when server failure occurs, the RADIUS Server Reorder on Failure feature provides for failover to another server in the server group. Subsequent to the failure, all RADIUS traffic is directed to the new server. Traffic is switched from the new server to another server in the server group only if the new server also fails. Traffic will not be automatically switched back to the first server.

By spreading the RADIUS transactions across multiple servers, authentication and accounting requests are serviced more quickly.

Feature Specifications for RADIUS Server Reorder on Failure

Feature History	
Release	**Modification**
12.3(1)	This feature was introduced.
Supported Platforms	
Cisco 7200, Cisco 7400, Cisco AS5300, Cisco AS5400, Cisco AS5800, Cisco AS5850	

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for RADIUS Server Reorder on Failure, page 334
- Restrictions for RADIUS Server Reorder on Failure, page 334
- Information About RADIUS Server Reorder on Failure, page 334
- How to Configure RADIUS Server Reorder on Failure, page 335
- Configuration Examples for RADIUS Server Reorder on Failure, page 339

- Additional References, page 341
- Command Reference, page 342

Prerequisites for RADIUS Server Reorder on Failure

- Before you can configure your RADIUS server to perform reorder on failure, you must enable authentication, authorization, and accounting (AAA) by using the **aaa new-model** command. (Refer to the chapter "AAA Overview" in the *Cisco IOS Security Configuration Guide*, Release 12.3.)
- You must also have RADIUS configured, for functions such as authentication, accounting, or static route download.

Restrictions for RADIUS Server Reorder on Failure

- An additional 4 bytes of memory is required per server group. However, because most server configurations have only a small number of server groups configured, the additional 4 bytes should have a minimal impact on performance.
- Some RADIUS features within the Cisco IOS software set may not be capable of using this feature. If a RADIUS feature cannot use the RADIUS Server Reorder on Failure feature, your server will behave as though the reorder feature is not configured.

Information About RADIUS Server Reorder on Failure

To configure the RADIUS Server Reorder on Failure feature, you must understand the following concepts:

- RADIUS Server Failure, page 334
- How the RADIUS Server Reorder on Failure Feature Works, page 335

RADIUS Server Failure

If the RADIUS Server Reorder on Failure feature is not configured and server failure occurs:

1. A new RADIUS transaction has to be performed.
2. A RADIUS packet for the transaction is sent to the first server in the group that is not marked dead (as per the configured deadtime) and is retransmitted for the configured number of retransmissions.
3. If all of those retransmits time out (as per the configured timeout), the router will transmit the packet to the next nondead server in the list for the configured number of retransmissions.
4. Step 3 is repeated until the specified maximum number of transmissions per transaction have been made. If the end of the list is reached before the maximum number of transmissions has been reached, the router will go back to the beginning of the list and continue from there.

If at any time during this process, a server meets the dead-server detection critiera (not configurable; it varies depending on the version of Cisco IOS software being used), the server is marked as dead for the configured deadtime.

How the RADIUS Server Reorder on Failure Feature Works

If you have configured the RADIUS Server Reorder on Failure feature, the decision about which RADIUS server to use as the initial server will be as follows:

- The network access server (NAS) maintains the status of "flagged" server, which is the first server to which a transmission is sent.
- After the transmission is sent to the flagged server, the transmission is sent to the flagged server again for the configured number of retransmissions.
- The NAS then sequentially sends failover "hello" messages through the list of nondead servers in the server group, starting with the one listed after the flagged server, until the configured transaction maximum tries is reached or until a response is received.
- At boot time, the flagged server is the first server in the server group list as was established using the **radius-server host** command.
- If the flagged server is marked as dead (even if the dead time is zero), the first nondead server listed after the flagged server becomes the flagged server.
- If the flagged server is the last server in the list, and it is marked as dead, the flagged server becomes the first server in the list that is not marked as dead.
- If all servers are marked as dead, the transaction fails, and no change is made to the flagged server.
- If the flagged server is marked as dead, and the dead timer expires, nothing happens.

When RADIUS Servers Are Dead

A server can be marked as dead if the criteria in 1 and 2 are met:

1. The server has not responded to at least the configured number of retransmissions as specified by the **radius-server transaction max-tries** command.
2. The server has not responded to any request for at least the configured timeout. The server is marked dead only if both criteria (this and the one listed above) are met. The marking of a server as dead, even if the dead time is zero, is significant for the RADIUS server retry method reorder system.

How to Configure RADIUS Server Reorder on Failure

This section contains the following procedures.

- Configuring a RADIUS Server to Reorder on Failure, page 335 (required)
- Monitoring RADIUS Server Reorder on Failure, page 337 (optional)

Configuring a RADIUS Server to Reorder on Failure

Perform this task to configure a server in a server group to direct traffic to another server in the server group when the first server fails.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **aaa new-model**
4. **radius-server retry method reorder**
5. **radius-server retransmit** {*retries*}
6. **radius-server transaction max-tries** {*number*}
7. **radius-server host** {*hostname* | *ip-address*} [*key string*]
8. **radius-server host** {*hostname* | *ip-address*} [*key string*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router (config)# aaa new-model`	Enables the AAA access control model.
Step 4	**radius-server retry method reorder** **Example:** `Router (config)# radius-server retry method reorder`	Specifies the reordering of RADIUS traffic retries among a server group.
Step 5	**radius-server retransmit** {*retries*} **Example:** `Router (config)# radius-server retransmit 1`	Specifies the number of times the Cisco IOS software searches the list of RADIUS server hosts before giving up. The *retries* argument is the maximum number of retransmission attempts. The default is 3 attempts.
Step 6	**radius-server transaction max-tries** {*number*} **Example:** `Router (config)# radius-server transaction max-tries 3`	Specifies the maximum number of transmissions per transaction that may be retried on a RADIUS server. The *number* argument is the total number of transmissions per transaction. If this command is not configured, the default is eight transmissions. **Note** This command is global across all RADIUS servers for a given transaction.

	Command or Action	Purpose
Step 7	**radius-server host** {*hostname* | *ip-address*} [**key** *string*] **Example:** `Router (config)# radius-server host 1.2.3.4 key radi23`	Specifies a RADIUS server host. **Note** You can also configure a global key for all RADIUS servers that do not have a per-server key configured by issuing the **radius-server key** command.
Step 8	**radius-server host** {*hostname* | *ip-address*} [**key** *string*] **Example:** `Router (config)# radius-server host 4.5.6.7 key rad234`	Specifies a RADIUS server host. **Note** At least two servers must be configured.

Monitoring RADIUS Server Reorder on Failure

To monitor the server-reorder-on-failure process on your router, use the following commands:

SUMMARY STEPS

1. **enable**
2. **debug aaa sg-server selection**
3. **debug radius**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug aaa sg-server selection** **Example:** `Router# debug aaa sg-server selection`	Displays information about why the RADIUS and TACAC+ server group system in the router is choosing a particular server.
Step 3	**debug radius** **Example:** `Router# debug radius`	Displays information about why the router is choosing a particular RADIUS server.

Examples

The following two debug outputs display the behavior of the RADIUS Server Reorder on Failure feature:

Debug 1

In the following sample output, the RADIUS Server Reorder on Failure feature is configured. The server retransmits are set to 0 (so each server is tried just one time before failover to the next configured server), and the transmissions per transaction are set to 4 (the transmissions will stop on the third failover). The third server in the server group (`10.107.164.118`) has accepted the transaction on the third transmission (second failover).

```
00:38:35: %SYS-5-CONFIG-I: Configured from console by console
00:38:53: RADIUS/ENCODE(OOOOOOOF) : ask "Username: "
00:38:53: RADIUS/ENCODE (0000000F) : send packet; GET-USER
00:38:58: RADIUS/ENCODE (0000000F) : ask "Password: "
00:38:58: RADIUS/ENCODE(0000000F) : send packet; GET-PASSWORD
00:38:59: RADIUS: AAA Unsupported [152] 4
00:38:59: RADIUS: 7474 [tt]
00:38:59: RADIUS (0000000F) : Storing nasport 2 in rad-db
00:38:59: RADIUS/ENCODE(0000000F) : dropping service type, "radius-server attribute 6
on-for-login-auth" is off
00:38:59: RADIUS (0000000F) : Config NAS IP: 0.0.0.0
00:38:59: RADIUS/ENCODE (0000000F) : acct-session-id: 15
00:38:59: RADIUS (0000000F) : sending
00:38:59: RADIUS/ENCODE: Best Local IP-Address 10.0.1.130 for Radius-Server 1.1.1.1
00:38:59: RAPIUS(0000000F) : Send Access-Request to 10.10.10.10:1645 id 21645/11, len 78
00:38:59: RADIUS:: authenticator 4481 E6 65 2D 5F 6F OA -lE F5 81 8F 4E 1478 9C
00:38:59: RADIUS: User-Name [1] 7 "david"
00:38:59: RADIUS: User-Password [2] 18 *
00:38:59: RADIUS: NAS-Port fSl 6 2
00:~8:59: RADIUS: NAS-Port-Type [61] 6 Virtual [5]
00:38:59: RADIUS: Calling-Station-Id [31] 15 "10.19.192.23"
00:39:00: RADIUS: NAS-IP-Address [4] 6 10.0.1.130
00:39:02: RADIUS: Fail-over to (10.2.2.2:1645,1646) for id 21645/11
00:39:02: RADIUS/ENCODE: Best Local IP-Address 10.0.1.130 for Radius-Server 2.2.2.2
00:39:04: RADIUS: Fail-over to (10.107.164.118:1645,1646) for id 21645/11
00:39:04: RADIUS/ENCODE: Best Local IP-Address 10.0.1.130 for Radius-Server
128.107.164.118
00:39:05: RADIUS: Received from id 21645/11 10.107.164.118:1645, Access-Accept, len 26
00:39:05: RADIUS: authenticator 5609 56 F9 64 4E DF 19- F3 A2 DD 73 EE 3F 9826
00:39:05: RADIUS: Service-Type [6] 6 Login [1]
```

Debug 2

In the following sample output, the RADIUS Server Reorder on Failure feature is configured. The server retransmits are set to 0, and the transmissions per transaction are set to 8. In this transaction, the transmission to server 10.10.10.0 has failed on the eighth transmission.

```
00:42:30: RADIUS(00000011): Received from id 21645/13
00:43:34: RADIUS/ENCODE(00000012) : ask "Username: "
00:43:34: RADIUS/ENCODE(00000012) : send packet; GET-USER
00:43:39: RADIUS/ENCODE(00000012) : ask "Password: "
00:43:39: RADIUS/ENCODE(00000012) : send packet; GET-PASSWORD
00:43:40: RADIUS: AAA Unsupported [152] 4
00:43:40: RADIUS: 7474 [tt]
00:43:40: RADIUS(00000012) : Storing nasport 2 in rad-db
00:43:40: RADIUS/ENCODE(00000012): dropping service type, "radius-server attribute 6
on-for-login-auth" is off
00:43:40: RADIUS(00000012) : Co~fig NAS IP: 0.0.0.0
00:43:40: RADIUS/ENCODE(00000012) : acct-session-id: 18
00:43:40: RADIUS(00000012) : sending
```

```
00:43:40: RADIUS/ENCODE: Best Local IP-Address 10.0.1.130 for Radius-Server 10.107.164.118
00:43:40: RADIUS(00000012) : Send Access-Request to 10.107.164.118:1645 id 21645/14, len
78 00:43:40: RADIUS: authenticator B8 OA 51 3A AF A6 0018 -B3 2E 94 5E 07 OB 2A IF
00:43:40: RADIUS: User-Name [1] 7 "david" 00:43:40: RADIUS: User-Password [2] 18 *
00:43:40: RADIUS: NAS-Port [5] 6 2
00:43:40: RADIUS: NAS-Port-Type [61] 6 Virtual [5] 00:43:40: RADIUS: Calling-Station-]d
[31] 15 "172.19.192.23" 00:43:40: RADIUS: NAS-IP-Address [4] 6 10.0.1.130
00:43:42: RADIUS: Fail-over to (10.10.10.10:1645,1646) for id 21645/14
00:43:42: RADIUS/ENCODE: Best Local IP-Address 10.0.1.130 for Radius-Server 1.1.1.1
00:43:44: RADius: Fail-over to (10.2.2.2:1645,1646) for id 21645/14
00:43:44: RADIUS/ENCODE: Best Local IP-Address 10.0.1.130 for Radius-Server 2.2.2.2
00:43:46: RADIUS: Fail-over to (10.107.164.118:1645,1646) for id 21645/14
00:43:46: RADIUS/ENCODE: Best Local IP-Address 10.0.1.130 for Radius-Server 10.107.164.118
00:43:48: RADIUS: Fail-over to (10.10.10.10:1645,1646) for id 21645/14
00:43:48: RADIUS/ENCODE: Best Local IP-Address 10.0.1.130 for Radius-Server 1.1.1.1
00:43:50: RADIUS: Fail-over to (10.2.2.2:1645,1646) for id 21645/14
00:43:50: RADIUS/ENCODE: Best Local IP-Address 10.0.1.130 for Radius-Server 2.2.2.2
00:43:52: RADIUS: Fail-over to (10.107.164.118:1645,1646) for id 21645/14
00:43:52: RADIUS/ENCODE: Best Local IP-Address 10.0.1.130 for Radius-Server 10.107.164.118
00:43:54: RADIUS: Fail-over to (10.10.10.10:1645,1646) for id 21645/14
00:43:54: RADIUS/ENCODE: Best Local IP-Address 10.0.1.130 for Radius-Server 1.1.1.1
00:43:56: RADIUS: No response from (10.10.10.10:1645,1646) for id 21645/14 00:43:56:
RADIUS/DECODE: parse response no app start; FAIL 00:43:56: RADIUS/DECODE: parse response;
FAIL
```

Configuration Examples for RADIUS Server Reorder on Failure

This section provides the following configuration examples:

- Configuring a RADIUS Server to Reorder on Failure Example, page 339
- Determining Transmission Order When RADIUS Servers Are Dead, page 339

Configuring a RADIUS Server to Reorder on Failure Example

The following configuration example shows that a RADIUS server is configured to reorder on failure. The maximum number of transmissions per transaction that may be retried on the RADIUS server is six.

```
aaa new-model
radius-server retry method reorder
radius-server retransmit 0
radius-server transaction max-tries 6
radius-server host 1.2.3.4 key rad123
radius-server host 4.5.6.7 key rad123
```

Determining Transmission Order When RADIUS Servers Are Dead

If at boot time you have configured the following:

```
aaa new-model
radius-server retry method reorder
radius-server retransmit 0
radius-server transaction max-tries 6
radius-server host 1.2.3.4
radius-server host 4.5.6.7
```

and both servers are down, but not yet marked dead, for the first transaction you would see the transmissions as follows:

```
1.2.3.4
4.5.6.7
1.2.3.4
4.5.6.7
1.2.3.4
4.5.6.7
```

If you configure the reorder as follows:

```
aaa new-model
radius-server retry method reorder
radius-server retransmit 1
radius-server transaction max-tries 3
radius-server host 1.2.3.4
radius-server host 3.4.5.6
```

and both RADIUS servers are not responding to RADIUS packets but are not yet marked dead (as after the NAS boots), the transmissions for the first transaction are as follows:

```
1.2.3.4
1.2.3.4
3.4.5.6
```

Subsequent transactions may be transmitted according to a different pattern. The transmissions will depend on whether the criteria for marking one (or both) servers as dead have been met, and as per the server flagging pattern already described.

If you configure the reorder as follows:

```
aaa new-model
radius-server retry method reorder
radius-server retransmit 1
radius-server max-tries-per-transaction 8
radius-server host 1.1.1.1
radius-server host 2.2.2.2
radius-server host 3.3.3.3
radius-server timeout 3
```

And the RADIUS server 1.1.1.1 is not responding to RADIUS packets but is not yet marked as dead, and the remaining two RADIUS servers are live, you will see the following:

For the first transaction:

```
1.1.1.1
1.1.1.1
2.2.2.2
```

For any additional transaction initiated for any transmissions before the server is marked as dead:

```
1.1.1.1
1.1.1.1
2.2.2.2
```

For transactions initiated thereafter:

```
2.2.2.2
```

If servers 2.2.2.2 and 3.3.3.3 then go down as well, you will see the following transmissions until servers 2.2.2.2 and 3.3.3.3 meet the criteria for being marked as dead:

```
2.2.2.2
2.2.2.2
```

```
3.3.3.3
3.3.3.3
1.1.1.1
1.1.1.1
2.2.2.2
2.2.2.2
```

The above is followed by the failure of the transmission and by the next method in the method list being used (if any).

If servers 2.2.2.2 and 3.3.3.3 go down but server 1.1.1.1 comes up at the same time, you see the following:

```
2.2.2.2
2.2.2.2
3.3.3.3
3.3.3.3
1.1.1.1
```

When servers 2.2.2.2 and 3.3.3.3 are then marked as dead, you see the following:

```
1.1.1.1
```

Additional References

For additional information related to RADIUS Server Reorder on Failure, refer to the following references:

Related Documents

Related Topic	Document Title
AAA and RADIUS	The section "Authentication, Authorization, and Accounting (AAA)" and the chapter "Configuring RADIUS," respectively, in the *Cisco IOS Security Configuration Guide*, Release 12.3
AAA and RADIUS commands	*Cisco IOS Security Command Reference,* Release 12.3
Enabling AAA	The chapter "AAA Overview" in the *Cisco IOS Security Configuration Guide*, Release 12.3

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified for this feature.	—

MIBs

MIBs[1]	MIBs Link
The CISCO-AAA-SERVER-MIB.my provides statistical information about how many transmissions go to which server.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

1. Not all supported MIBs are listed.

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified for this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **debug aaa sg-server selection**
- **radius-server retry method reorder**
- **radius-server transaction max-tries**

Subscriber Service Switch

The Subscriber Service Switch provides flexibility on where and how many subscribers are connected to available services and how those services are defined. The primary focus of Subscriber Service Switch is to direct PPP from one point to another using a Layer 2 subscriber policy. The policy will manage tunneling of PPP in a policy-based bridging fashion.

Feature Specifications for the Subscriber Service Switch Feature

Feature History	
Release	**Modification**
12.2(13)T	This feature was introduced.
12.2(27)SB	This feature was integrated into Cisco IOS Release 12.2(27)SB.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Restrictions for Subscriber Service Switch, page 344
- Information About Subscriber Service Switch, page 344
- How to Use Subscriber Service Switch, page 347
- Configuration Examples for Subscriber Service Switch, page 353
- Additional References, page 368
- Command Reference, page 370
- Glossary, page 372

Restrictions for Subscriber Service Switch

Subscriber Service Switch provides the framework for the management and scalability of PPP sessions that are switched from one virtual PPP link to another. Subscriber Service Switch will provide the infrastructure for any protocol to plug into, but the initial focus will be on switching PPP over Ethernet (PPPoE) and PPP over ATM (PPPoA) sessions to a Layer 2 Tunneling Protocol (L2TP) devices such as an L2TP access concentrator (LAC) switch, and switching L2TP sessions to an L2TP tunnel switch.

Information About Subscriber Service Switch

To configure Subscriber Service Switch, you need to understand the following concepts:

- Benefits of Subscriber Service Switch, page 344
- Backward Compatibility, page 345

Benefits of Subscriber Service Switch

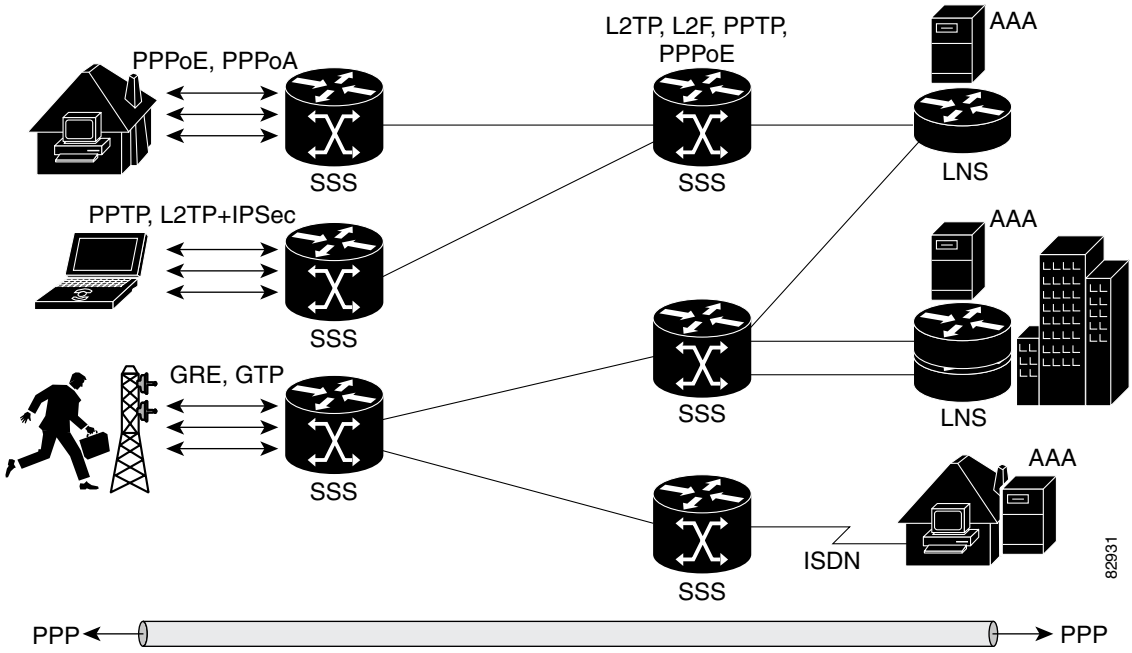
The Subscriber Service Switch was developed in response to a need by Internet service providers for increased scalability and extensibility for remote access service selection and Layer 2 subscriber policy management. This Layer 2 subscriber policy is needed to manage tunneling of PPP in a policy-based bridging fashion.

Subscriber Service Switch provides flexibility on where and how many subscribers are connected to available services and how those services are defined. In the past, remote access service selection was largely determined by the telephone number dialed or the PPP username and password entered during a PPP authentication cycle. However, emerging broadband, cable, Virtual Private Network (VPN), and wireless access methods have created an environment where PPP sessions may be tunneled over a variety of protocols and media. The multitude of protocols, management domains, network infrastructure, and variety of services has created a complex environment for directing a subscriber to a given service or application. The problem is further augmented by the much greater density of total PPP sessions that can be transported over shared media versus traditional point-to-point links. Subscriber Service Switch can provide a flexible and extensible decision point linking an incoming subscriber (typically a PPP session over some physical or virtual link) to another tunneled link or local termination for Layer 3 processing.

Subscriber Service Switch is also scalable in situations where a subscriber's Layer 2 service is switched across virtual links. Examples include switching between PPPoA, PPPoE, L2TP, Layer 2 Forwarding Protocol (L2F), Point-to-Point Tunneling Protocol (PPTP), generic routing encapsulation (GRE) and General Packet Radio Service (GPRS) Tunneling Protocol (GTP wireless data standard).

Figure 1 shows how Subscriber Service Switch provides its own centralized switching path that bypasses the virtual access-based switching available in software prior to Cisco IOS Release 12.2(13)T. In the figure, Subscriber Service Switch is switching data traffic from personal computers in a home and corporate office, and from a wireless user.

Figure 15 BASIC Subscriber Service Switch Operation



Protocols that register with the Subscriber Service Switch application programming interface (API) can take advantage of this new switching path. Bypassing the virtual access interface in this manner helps the Cisco IOS software to scale to the increased number of sessions that the market demands today. Subscriber Service Switch also markedly improves network performance, too. For example, benchmark testing indicates that performance of L2TP multihop tasks occurs twice as fast in networks with Subscriber Service Switch versus networks without it.

Backward Compatibility

All of the current virtual private dialup network (VPDN), Multichassis Multilink PPP (MMLP), and local termination policies and configurations will be maintained in this implementation of Subscriber Service Switch; however, default policies may be overridden by the following configurations or events:

- Resource Manager (RM) VPDN authorization is attempted before VPDN authorization.
- VPDN authorization is attempted before Stack Group Forwarding (SGF) MMLP.
- VPDN service authorization is attempted only when the **vpdn enable** command is configured.
- RM VPDN service authorization is attempted only if RM is enabled.
- SGF authorization is attempted only when the **sgbp member** command is configured and one or both of the following service keys are available from the subscriber: unauthenticated PPP name and endpoint discriminator.
- The **dnis** and **domain** service keys, in that order, are used to authorize VPDN service, provided that VPDN service is enabled. The order may be changed with the **vpdn search-order** global command, which may include **multihop-hostname** as a service key.
- An unauthenticated PPP name is always reduced to a domain name by taking all characters from the right of the PPP name up to a configurable delimiter character (default is the @ character). Only the domain portion is used to locate a service.

- If the **vpdn authen-before-forward** command is configured as a global configuration command, the authenticated PPP name is used to authorize VPDN service.
- The configuration defined by the **vpdn-group** command can specify four things:
  1. Authorization for VPDN call termination (using the **accept-dialin** and **accept-dialout** keywords).
  2. Authorization for VPDN subscriber service (using the **request-dialin** and **request-dialout** keywords).
  3. A directive to collect further service keys and reauthorize (using the **authen-before-forward** keyword).
  4. A tunnel configuration.

  Subscriber Service Switch adds a general configuration framework to replace the first three aspects of a VPDN group.
- If VPDN and SGF services either are not configured or cannot be authorized, local PPP termination service is selected. Further PPP authorization is still required to complete local termination.
- A two-phase authorization scheme is enabled by the **vpn domain authorization** command. An NAS-Port-ID (NAS port identifier) key is used to locate the first service record, which contains a restricted set of values for the domain substring of the unauthenticated PPP name. This filtered service key then locates the final service. Cisco refers to this scheme as *domain preauthorization*.
- Domain preauthorization will occur only when the NAS-Port-ID key is available.
- When domain preauthorization is enabled, both authenticated and unauthenticated domain names are checked for restrictions.
- It is possible to associate a fixed service with an ATM permanent virtual circuit (PVC), thus affecting any subscribers carried by the PVC. The **vpn service** command, in ATM VC or VC class configuration mode, and the associated key make up the generic service key.
- When the generic service key is available, it will be used for authorization instead of the unauthenticated domain name.
- If either the **vpdn authen-before-forward** or **per vpdn-group authen-before-forward** command is configured, the authenticated username is required and will be used to authorize VPDN service.
- To determine whether the **authen-before-forward** command is configured in a VPDN group (using the **vpdn-group** command), an unauthenticated username or the generic service key is required as the initial-want key set.
- When the global **vpdn authen-before-forward** command is not configured, the generic service key, if one is available, is used to determine whether the **authen-before-forward** function is configured in the VPDN group (using the **vpdn-group** command). If the generic service key is not available, the unauthenticated username will be used.
- If an accounting-enabled key is available, the unauthenticated username is required.
- VPDN multihop is allowed only when VPDN multihop is enabled.
- SGF on the L2TP network server (LNS) is allowed only when VPDN multihop is enabled on the LNS.
- Forwarding of SGF calls on the LAC is only allowed if VPDN multihop is enabled on the LAC.
- SGF-to-SGF multihop is not allowed.
- When PPP forwarding is configured, both MLP and non-MLP calls are forwarded to the winner of the Stack Group Bidding Protocol (SGBP) bid.
- Authentication is always required for forwarded Packet Data Serving Node (PDSN) calls.

- When the **directed-request** function is enabled and activated using the **ip host** command (legacy behavior), VPDN service authorization occurs only when the **vpdn authorize directed-request** function is enabled.
- Fixed legacy policy is still maintained for RM.

How to Use Subscriber Service Switch

The Subscriber Service Switch architecture is transparent, and existing PPP, VPDN, PPPoE, PPPoA, and authentication, authorization, and accounting (AAA) call configurations will continue to work in this new environment. You can, however, enable Subscriber Service Switch preauthorization and Subscriber Service Switch type authorization. You may also find it helpful to verify Subscriber Service Switch call operation.

This section contains the following optional procedures:

- Enabling Domain Preauthorization on a LAC, page 347 (optional)
- Enabling Subscriber Service Switch Preauthorization, page 348(optional)
- Verifying Subscriber Service Switch Call Operation, page 349 (optional)
- Troubleshooting the Subscriber Service Switch, page 350 (optional)

Enabling Domain Preauthorization on a LAC

To enable the LAC to perform domain authorization before tunneling, perform the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vpdn authorize domain**
4. **exit**
5. **show running-config**
6. Creating a RADIUS User Profile for Domain Preauthorization, page 348

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**vpdn authorize domain** **Example:** `Router(config)# vpdn authorize domain`	Enables domain preauthorization on a NAS.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 5	**show running-config** **Example:** `Router# show running-config`	Displays the configuration so you can check that you successfully enabled domain preauthorization.
Step 6	Create a RADIUS user profile for domain preauthorization.	See the next section and the "Domain Preauthorization RADIUS User Profile Example" on page 11.

Creating a RADIUS User Profile for Domain Preauthorization

Table 1 lists the attributes to enable domain preauthorization in a RADIUS user profile. Refer to the *Cisco IOS Security Configuration Guide*, Release 12.2, for information about creating a RADIUS user profile.

Table 18 Attributes for the RADIUS User Profile for Domain Preauthorization

RADIUS Entry	Purpose
nas-port:*ip-address:slot/subslot/port/vpi.vci*	Configures the NAS port username for domain preauthorization. • *ip-address*—Management IP address of the node switch processor (NSP). • *slot/subslot/port*—Specify ATM interface. • *vpi.vci*—Virtual path identifier (VPI) and virtual channel identifier (VCI) values for the PVC.
Password = "cisco"	Sets the fixed password.
User-Service-Type = Outbound-User	Configures the service-type as outbound.
Cisco-AVpair = "vpdn:vpn-domain-list=*domain1, domain2,...***"**	Specifies the domains accessible to the user. • *domain*—Domain to configure as accessible to the user.

Enabling Subscriber Service Switch Preauthorization

When Subscriber Service Switch preauthorization is enabled on a LAC, local configurations for session limit per VC and per VLAN are overwritten by the per-NAS-port session limit downloaded from the server. To enable this preauthorization, perform the following steps:

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **subscriber access** {**pppoe** | **pppoa**} **pre-authorize nas-port-id** [*aaa-method-list*]
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**subscriber access** {**pppoe** | **pppoa**} **pre-authorize nas-port-id** [*aaa-method-list*] **Example:** `Router(config)# subscriber access pppoe pre-authorize nas-port-id mlist_llid`	Enables Subscriber Service Switch preauthorization. **Note** The LACs maintains a current session number per NAS port. As a new session request comes in, the LAC makes a preauthorization request to AAA to get the session limit, and compares it with the number of sessions currently on that NAS port. This command ensures that session limit querying is only enabled for PPPoE-type calls, not for any other call types.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Verifying Subscriber Service Switch Call Operation

To verify that the Subscriber Service Switch is working, perform the following optional steps:

SUMMARY STEPS

1. **enable**
2. **show sss session** [**all**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**show sss session** [**all**] **Example:** `Router# show sss session all`	Displays Subscriber Service Switch session status. • Use the optional **all** keyword to display an extensive report about the Subscriber Service Switch sessions.

Information about troubleshooting a network running the Subscriber Service Switch can be found in the following "Troubleshooting the Subscriber Service Switch, page 350 section.

Troubleshooting the Subscriber Service Switch

This section provides troubleshooting tips for the Subscriber Service Switch. Examples of normal and failure operations can be found in Troubleshoot Subscriber Service Switch Examples on page 14. Reports from **debug** commands should be sent to technical personnel at Cisco Systems for evaluation

Debug Commands Available for Subscriber Service Switch

The Subscriber Service Switch feature introduces five new EXEC mode **debug** commands to enable diagnostic output about Subscriber Service Switch call operation, as follows:

- **debug sss event**—Displays diagnostic information about Subscriber Service Switch call setup events.
- **debug sss error**—Displays diagnostic information about errors that may occur during Subscriber Service Switch call setup.
- **debug sss fsm**—Displays diagnostic information about the Subscriber Service Switch call setup state.
- **debug sss aaa authorization event**—Displays messages about AAA authorization events that are part of normal call establishment.
- **debug sss aaa authorization fsm**—Displays messages about AAA authorization state changes.

These commands were designed to be used with Cisco IOS **debug** commands that exist for troubleshooting PPP and other Layer 2 call operations. Table 2 lists some of these **debug** commands.

Table 19 Additional Debugging Commands for Troubleshooting Subscriber Service Switch

Command	Purpose
debug pppoe events	Displays protocol event information.
debug pppoe errors	Displays PPPoE error messages.
debug ppp negotiation	Allows you to check that a client is passing PPP negotiation information.

Table 19 Additional Debugging Commands for Troubleshooting Subscriber Service Switch

Command	Purpose
debug vpdn l2x-events	Displays L2F and L2TP events that are part of tunnel establishment or shutdown.
debug vpdn l2x-errors	Displays L2F and L2TP protocol errors that prevent tunnel establishment or normal operation.
debug vpdn sss events	Displays diagnostic information about VPDN Subscriber Service Switch call setup events.
debug vpdn sss errors	Displays diagnostic information about errors that may occur during VPDN Subscriber Service Switch call setup.
debug vpdn call events	Enables VPDN call event debugging.
debug vpdn call fsm	Enables VPDN call setup state debugging.
debug vpdn events	Displays PPTP tunnel event change information.
debug vpdn errors	Displays PPTP protocol error messages.

Note The commands are intended only for troubleshooting purposes, because the volume of output generated by the software can result in severe performance degradation on the router.

Troubleshoot the Subscriber Service Switch

To troubleshoot a network running the Subscriber Service Switch, perform the following steps:

SUMMARY STEPS

1. Attach a console directly to a router running the Cisco IOS Release 12.2(13)T or a later release.
2. **enable**
3. **configure terminal**
4. **no logging console**
5. Use Telnet to access a router port and repeat Steps 2 and 3.
6. **terminal monitor**
7. **exit**
8. **debug** *command*
9. **configure {terminal | memory | network}**
10. **no terminal monitor**
11. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Attach a console directly to a router running the Cisco IOS Release 12.2(13)T or a later release.	—
Step 2	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**no logging console** **Example:** `Router(config)# no logging console`	Disables all logging to the console terminal. To reenable logging to the console, use the **logging console** command in global configuration mode.
Step 5	Use Telnet to access a router port and repeat Steps 2 and 3.	Enters global configuration mode in a recursive Telnet session, which allows the output to be redirected away from the console port.
Step 6	**terminal monitor** **Example:** `Router(config)# terminal monitor`	Enables logging output on the virtual terminal.
Step 7	**exit** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.
Step 8	**debug** *command* **Example:** `Router# debug sss error` `Router# debug sss event` `Router# debug sss fsm`	Enables the **debug** command. See "Debug Commands Available for Subscriber Service Switch" and Table 2 for commands that can be entered. **Note** You can enter more than one **debug** command.
Step 9	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 10	**no terminal monitor** **Example:** `Router(config)# no terminal monitor`	Disables logging on the virtual terminal.
Step 11	**exit** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.

Configuration Examples for Subscriber Service Switch

This section provides the following configuration examples:

- Enable LAC Domain Authorization Example, page 353
- Domain Preauthorization RADIUS User Profile Example, page 353
- Enable Subscriber Service Switch Preauthorization Example, page 354
- Verify Subscriber Service Switch Call Operation Example, page 354
- Troubleshoot Subscriber Service Switch Examples, page 356

Enable LAC Domain Authorization Example

The following example shows the configuration necessary for the LAC to participate in domain preauthorization:

```
!
aaa new-model
aaa authorization network default local group radius
!
vpdn authorize domain
!
radius-server host 10.9.9.9 auth-port 1645 acct-port 1646
radius-server attribute nas-port format d
radius-server key MyKey
radius-server vsa send authentication
!
```

Domain Preauthorization RADIUS User Profile Example

The following example shows a typical domain preauthorization RADIUS user profile:

```
user = nas-port:10.9.9.9:0/0/0/30.33{
 profile_id = 826
 profile_cycle = 1
 radius=Cisco {
 check_items= {
 2=cisco
 }
 reply_attributes= {
 9,1="vpdn:vpn-domain-list=net1.com,net2.com"
 6=5
```

```
 }
 }
 }
```

Enable Subscriber Service Switch Preauthorization Example

The following partial example signals Subscriber Service Switch to preauthorize the NAS-Port-ID string before authorizing the domain name. This policy applies only to all sessions with a PPPoE access type.

```
vpdn-group 3
 accept dialin
  protocol pppoe
  virtual-template 1
!
! Signals Subscriber Service Switch to preauthorize the NAS-Port-ID string before
! authorizing the domain name.
subscriber access pppoe pre-authorize nas-port-id mlist_llid
!
```

Verify Subscriber Service Switch Call Operation Example

The following example command output from the **show sss session all** command provides an extensive report of Subscriber Service Switch session activity. The text in bold shows the unique identifier for each session, which can be used to correlate that particular session with the session information retrieved from other **show** commands or **debug** command traces. See the following **show vpdn session** command output for an example of this unique ID correlation.

```
Router# show sss session all

Current SSS Information: Total sessions 9

SSS session handle is 40000013, state is connected, service is VPDN
Unique ID is 9
SIP subscriber access type(s) are PPPoE/PPP
Identifier is nobody3@xyz.com
Last Changed 00:02:49
Root SIP Handle is DF000010, PID is 49
AAA unique ID is 10
Current SIP options are Req Fwding/Req Fwded

SSS session handle is B0000017, state is connected, service is VPDN
Unique ID is 10
SIP subscriber access type(s) are PPPoE/PPP
Identifier is nobody3@xyz.com
Last Changed 00:02:05
Root SIP Handle is B9000015, PID is 49
AAA unique ID is 11
Current SIP options are Req Fwding/Req Fwded

SSS session handle is D6000019, state is connected, service is VPDN
Unique ID is 11
SIP subscriber access type(s) are PPPoE/PPP
Identifier is nobody3@xyz.com
Last Changed 00:02:13
Root SIP Handle is D0000016, PID is 49
AAA unique ID is 12
Current SIP options are Req Fwding/Req Fwded

SSS session handle is 8C000003, state is connected, service is VPDN
```

```
Unique ID is 3
SIP subscriber access type(s) are PPPoE/PPP
Identifier is user3@xyz.com
Last Changed 2d21h
Root SIP Handle is D3000002, PID is 49
AAA unique ID is 3
Current SIP options are Req Fwding/Req Fwded

SSS session handle is BE00000B, state is connected, service is Local Term
Unique ID is 6
SIP subscriber access type(s) are PPPoE/PPP
Identifier is user1
Last Changed 00:03:56
Root SIP Handle is A9000009, PID is 49
AAA unique ID is 7
Current SIP options are Req Fwding/Req Fwded

SSS session handle is DC00000D, state is connected, service is Local Term
Unique ID is 7
SIP subscriber access type(s) are PPPoE/PPP
Identifier is user2
Last Changed 00:03:57
Root SIP Handle is 2C00000A, PID is 49
AAA unique ID is 8
Current SIP options are Req Fwding/Req Fwded

SSS session handle is DB000011, state is connected, service is VPDN
Unique ID is 8
SIP subscriber access type(s) are PPPoE/PPP
Identifier is nobody3@xyz.com
Last Changed 00:02:58
Root SIP Handle is 1000000F, PID is 49
AAA unique ID is 9
Current SIP options are Req Fwding/Req Fwded

SSS session handle is 3F000007, state is connected, service is Local Term
Unique ID is 2
SIP subscriber access type(s) are PPP
Identifier is johndoe
Last Changed 00:05:30
Root SIP Handle is 8A000009, PID is 92
AAA unique ID is 1
Current SIP options are Req Fwding/Req Fwded

SSS session handle is 97000005, state is connected, service is VPDN
Unique ID is 4
SIP subscriber access type(s) are PPP
Identifier is nobody2@xyz.com
Last Changed 00:07:16
Root SIP Handle is 32000000, PID is 92
AAA unique ID is 5
Current SIP options are Req Fwding/Req Fwded
```

Correlating the Unique ID in show vpdn session all Command Output

The following partial sample output from the **show vpdn session all** command provides extensive reports on call activity for all L2TP, L2F, and PPPoE sessions, and identifies the unique ID for each session.

```
Router# show vpdn session all

L2TP Session Information Total tunnels 1 sessions 4

Session id 5 is up, tunnel id 13695
```

```
Call serial number is 3355500002
Remote tunnel name is User03
  Internet address is 10.0.0.63
  Session state is established, time since change 00:03:53
    52 Packets sent, 52 received
    2080 Bytes sent, 1316 received
  Last clearing of "show vpdn" counters never
  Session MTU is 1464 bytes
  Session username is nobody3@xyz.com
    Interface
    Remote session id is 692, remote tunnel id 58582
  UDP checksums are disabled
  SSS switching enabled
  No FS cached header information available
  Sequencing is off
  Unique ID is 8

Session id 6 is up, tunnel id 13695
Call serial number is 3355500003
Remote tunnel name is User03
  Internet address is 10.0.0.63
  Session state is established, time since change 00:04:22
    52 Packets sent, 52 received
    2080 Bytes sent, 1316 received
  Last clearing of "show vpdn" counters never
  Session MTU is 1464 bytes
  Session username is nobody3@xyz.com
    Interface
    Remote session id is 693, remote tunnel id 58582
  UDP checksums are disabled
  SSS switching enabled
  No FS cached header information available
  Sequencing is off
  Unique ID is 9
.
.
.
```

Troubleshoot Subscriber Service Switch Examples

This section provides the following debugging session examples for a network running the Subscriber Service Switch:

- Troubleshoot the Subscriber Service Switch Operation Example, page 357
- Troubleshoot the Subscriber Service Switch on the LAC—Normal Operation Example, page 358
- Troubleshoot the Subscriber Service Switch on the LAC—Authorization Failure Example, page 360
- Troubleshoot the Subscriber Service Switch on the LAC—Authentication Failure Example, page 362
- Troubleshoot the Subscriber Service Switch at the LNS—Normal Operation Example, page 365
- Troubleshoot the Subscriber Service Switch at the LNS—Tunnel Failure Example, page 367

Note Reports from **debug** commands should be sent to technical personnel at Cisco Systems for evaluation.

Troubleshoot the Subscriber Service Switch Operation Example

The following example shows the **debug** commands used and sample output for debugging Subscriber Service Switch operation:

```
Router# debug sss event
Router# debug sss error
Router# debug sss state
Router# debug sss aaa authorization event
Router# debug sss aaa authorization fsm

SSS:
  SSS events debugging is on
  SSS error debugging is on
  SSS fsm debugging is on
  SSS AAA authorization event debugging is on
  SSS AAA authorization FSM debugging is on

*Mar  4 21:33:18.248: SSS INFO: Element type is Access-Type, long value is 3
*Mar  4 21:33:18.248: SSS INFO: Element type is Switch-Id, long value is -1509949436
*Mar  4 21:33:18.248: SSS INFO: Element type is Nasport, ptr value is 6396882C
*Mar  4 21:33:18.248: SSS INFO: Element type is AAA-Id, long value is 7
*Mar  4 21:33:18.248: SSS INFO: Element type is AAA-ACCT_ENBL, long value is 1
*Mar  4 21:33:18.248: SSS INFO: Element type is AccIe-Hdl, ptr value is 78000006
*Mar  4 21:33:18.248: SSS MGR [uid:7]: Event service-request, state changed from
wait-for-req to wait-for-auth
*Mar  4 21:33:18.248: SSS MGR [uid:7]: Handling Policy Authorize (1 pending sessions)
*Mar  4 21:33:18.248: SSS PM [uid:7]: Need the following key: Unauth-User
*Mar  4 21:33:18.248: SSS PM [uid:7]: Received Service Request
*Mar  4 21:33:18.248: SSS PM [uid:7]: Event <need keys>, State: initial-req to
need-init-keys
*Mar  4 21:33:18.248: SSS PM [uid:7]: Policy reply - Need more keys
*Mar  4 21:33:18.248: SSS MGR [uid:7]: Got reply Need-More-Keys from PM
*Mar  4 21:33:18.248: SSS MGR [uid:7]: Event policy-or-mgr-more-keys, state changed from
wait-for-auth to wait-for-req
*Mar  4 21:33:18.248: SSS MGR [uid:7]: Handling More-Keys event
*Mar  4 21:33:20.256: SSS INFO: Element type is Unauth-User, string value is
nobody@xyz.com
*Mar  4 21:33:20.256: SSS INFO: Element type is AccIe-Hdl, ptr value is 78000006
*Mar  4 21:33:20.256: SSS INFO: Element type is AAA-Id, long value is 7
*Mar  4 21:33:20.256: SSS INFO: Element type is Access-Type, long value is 0
*Mar  4 21:33:20.256: SSS MGR [uid:7]: Event service-request, state changed from
wait-for-req to wait-for-auth
*Mar  4 21:33:20.256: SSS MGR [uid:7]: Handling Policy Authorize (1 pending sessions)
*Mar  4 21:33:20.256: SSS PM [uid:7]: Received More Initial Keys
*Mar  4 21:33:20.256: SSS PM [uid:7]: Event <rcvd keys>, State: need-init-keys to
check-auth-needed
*Mar  4 21:33:20.256: SSS PM [uid:7]: Handling Authorization Check
*Mar  4 21:33:20.256: SSS PM [uid:7]: Event <send auth>, State: check-auth-needed to
authorizing
*Mar  4 21:33:20.256: SSS PM [uid:7]: Handling AAA service Authorization
*Mar  4 21:33:20.256: SSS PM [uid:7]: Sending authorization request for 'xyz.com'
*Mar  4 21:33:20.256: SSS AAA AUTHOR [uid:7]:Event <make request>, state changed from idle
to authorizing
*Mar  4 21:33:20.256: SSS AAA AUTHOR [uid:7]:Authorizing key xyz.com
*Mar  4 21:33:20.260: SSS AAA AUTHOR [uid:7]:AAA request sent for key xyz.com
*Mar  4 21:33:20.260: SSS AAA AUTHOR [uid:7]:Received an AAA pass
*Mar  4 21:33:20.260: SSS AAA AUTHOR [uid:7]:Event <found service>, state changed from
authorizing to complete
*Mar  4 21:33:20.260: SSS AAA AUTHOR [uid:7]:Found service info for key xyz.com
*Mar  4 21:33:20.260: SSS AAA AUTHOR [uid:7]:Event <free request>, state changed from
complete to terminal
*Mar  4 21:33:20.260: SSS AAA AUTHOR [uid:7]:Free request
```

```
*Mar  4 21:33:20.264: SSS PM [uid:7]: Event <found>, State: authorizing to end
*Mar  4 21:33:20.264: SSS PM [uid:7]: Handling Service Direction
*Mar  4 21:33:20.264: SSS PM [uid:7]: Policy reply - Forwarding
*Mar  4 21:33:20.264: SSS MGR [uid:7]: Got reply Forwarding from PM
*Mar  4 21:33:20.264: SSS MGR [uid:7]: Event policy-start-service-fsp, state changed from
wait-for-auth to wait-for-service
*Mar  4 21:33:20.264: SSS MGR [uid:7]: Handling Connect-Forwarding-Service event
*Mar  4 21:33:20.272: SSS MGR [uid:7]: Event service-fsp-connected, state changed from
wait-for-service to connected
*Mar  4 21:33:20.272: SSS MGR [uid:7]: Handling Forwarding-Service-Connected event
```

Troubleshoot the Subscriber Service Switch on the LAC—Normal Operation Example

The following example shows the **debug** commands used and sample output indicating normal operation of the Subscriber Service Switch on the LAC:

```
Router# debug sss event
Router# debug sss error
Router# debug sss aaa authorization event
Router# debug sss aaa authorization fsm
Router# debug  pppoe events
Router# debug pppoe errors
Router# debug ppp negotiation
Router# debug vpdn l2x-events
Router# debug vpdn l2x-errors
Router# debug  vpdn sss events
Router# debug vpdn sss errors
Router# debug vpdn call events
Router# debug vpdn call fsm
Router# debug vpdn events
Router# debug vpdn errors

SSS:
  SSS events debugging is on
  SSS error debugging is on
  SSS AAA authorization event debugging is on
  SSS AAA authorization FSM debugging is on
PPPoE:
  PPPoE protocol events debugging is on
  PPPoE protocol errors debugging is on
PPP:
  PPP protocol negotiation debugging is on

VPN:
  L2X protocol events debugging is on
  L2X protocol errors debugging is on
  VPDN SSS events debugging is on
  VPDN SSS errors debugging is on
  VPDN call event debugging is on
  VPDN call FSM debugging is on
  VPDN events debugging is on
  VPDN errors debugging is on

*Nov 15 12:23:52.523: PPPoE 0: I PADI R:0000.0c14.71d0 L:ffff.ffff.ffff 1/32
ATM4/0.132
*Nov 15 12:23:52.523: PPPoE 0: O PADO R:0000.0c14.71d0 L:00b0.c2e9.c870 1/32
ATM4/0.132
*Nov 15 12:23:52.527: PPPoE 0: I PADR R:0000.0c14.71d0 L:00b0.c2e9.c870 1/32
ATM4/0.132
*Nov 15 12:23:52.527: PPPoE : encap string prepared
*Nov 15 12:23:52.527: [13]PPPoE 10: Access IE handle allocated
*Nov 15 12:23:52.527: [13]PPPoE 10: pppoe SSS switch updated
```

```
*Nov 15 12:23:52.527: [13]PPPoE 10: Service request sent to SSS
*Nov 15 12:23:52.527: [13]PPPoE 10: Created R:00b0.c2e9.c870 L:0000.0c14.71d0 1/32
ATM4/0.132
*Nov 15 12:23:52.547: SSS INFO: Element type is Access-Type, long value is 3
*Nov 15 12:23:52.547: SSS INFO: Element type is Switch-Id, long value is 2130706444
*Nov 15 12:23:52.547: SSS INFO: Element type is Nasport, ptr value is 63C07288
*Nov 15 12:23:52.547: SSS INFO: Element type is AAA-Id, long value is 14
*Nov 15 12:23:52.547: SSS INFO: Element type is AccIe-Hdl, ptr value is B200000C
*Nov 15 12:23:52.547: SSS MGR [uid:13]: Handling Policy Authorize (1 pending
sessions)
*Nov 15 12:23:52.547: SSS PM [uid:13]: RM/VPDN disabled: RM/VPDN author not needed
*Nov 15 12:23:52.547: SSS PM [uid:13]: Received Service Request
*Nov 15 12:23:52.547: SSS PM [uid:13]: Handling Authorization Check
*Nov 15 12:23:52.547: SSS PM [uid:13]: Policy requires 'Unauth-User' key
*Nov 15 12:23:52.547: SSS PM [uid:13]: Policy reply - Need more keys
*Nov 15 12:23:52.547: SSS MGR [uid:13]: Got reply Need-More-Keys from PM
*Nov 15 12:23:52.547: SSS MGR [uid:13]: Handling More-Keys event
*Nov 15 12:23:52.547: [13]PPPoE 10: State REQ_NASPORT    Event MORE_KEYS
*Nov 15 12:23:52.547: [13]PPPoE 10: O PADS R:0000.0c14.71d0 L:00b0.c2e9.c870 1/32
ATM4/0.132
*Nov 15 12:23:52.547: ppp13 PPP: Using default call direction
*Nov 15 12:23:52.547: ppp13 PPP: Treating connection as a dedicated line
*Nov 15 12:23:52.547: ppp13 PPP: Phase is ESTABLISHING, Active Open
*Nov 15 12:23:52.547: ppp13 LCP: O CONFREQ [Closed] id 1 len 19
*Nov 15 12:23:52.547: ppp13 LCP:    MRU 1492 (0x010405D4)
*Nov 15 12:23:52.547: ppp13 LCP:    AuthProto CHAP (0x0305C22305)
*Nov 15 12:23:52.547: ppp13 LCP:    MagicNumber 0xB0EC4557 (0x0506B0EC4557)
*Nov 15 12:23:52.547: [13]PPPoE 10: State START_PPP    Event DYN_BIND
*Nov 15 12:23:52.547: [13]PPPoE 10: data path set to PPP
*Nov 15 12:23:52.571: ppp13 LCP: I CONFREQ [REQsent] id 1 len 14
*Nov 15 12:23:52.571: ppp13 LCP:    MRU 1492 (0x010405D4)
*Nov 15 12:23:52.571: ppp13 LCP:    MagicNumber 0x0017455D (0x05060017455D)
*Nov 15 12:23:52.571: ppp13 LCP: O CONFACK [REQsent] id 1 len 14
*Nov 15 12:23:52.571: ppp13 LCP:    MRU 1492 (0x010405D4)
*Nov 15 12:23:52.571: ppp13 LCP:    MagicNumber 0x0017455D (0x05060017455D)
*Nov 15 12:23:54.543: ppp13 LCP: TIMEout: State ACKsent
*Nov 15 12:23:54.543: ppp13 LCP: O CONFREQ [ACKsent] id 2 len 19
*Nov 15 12:23:54.543: ppp13 LCP:    MRU 1492 (0x010405D4)
*Nov 15 12:23:54.543: ppp13 LCP:    AuthProto CHAP (0x0305C22305)
*Nov 15 12:23:54.543: ppp13 LCP:    MagicNumber 0xB0EC4557 (0x0506B0EC4557)
*Nov 15 12:23:54.543: ppp13 LCP: I CONFACK [ACKsent] id 2 len 19
*Nov 15 12:23:54.543: ppp13 LCP:    MRU 1492 (0x010405D4)
*Nov 15 12:23:54.543: ppp13 LCP:    AuthProto CHAP (0x0305C22305)
*Nov 15 12:23:54.543: ppp13 LCP:    MagicNumber 0xB0EC4557 (0x0506B0EC4557)
*Nov 15 12:23:54.543: ppp13 LCP: State is Open
*Nov 15 12:23:54.543: ppp13 PPP: Phase is AUTHENTICATING, by this end
*Nov 15 12:23:54.543: ppp13 CHAP: O CHALLENGE id 1 len 25 from "7200"
*Nov 15 12:23:54.547: ppp13 CHAP: I RESPONSE id 1 len 38 from "nobody@xyz.com"
*Nov 15 12:23:54.547: ppp13 PPP: Phase is FORWARDING, Attempting Forward
*Nov 15 12:23:54.547: SSS INFO: Element type is Unauth-User, string value is
nobody@xyz.com
*Nov 15 12:23:54.547: SSS INFO: Element type is AccIe-Hdl, ptr value is B200000C
*Nov 15 12:23:54.547: SSS INFO: Element type is AAA-Id, long value is 14
*Nov 15 12:23:54.547: SSS INFO: Element type is Access-Type, long value is 0
*Nov 15 12:23:54.547: SSS MGR [uid:13]: Handling Policy Authorize (1 pending
sessions)
*Nov 15 12:23:54.547: SSS PM [uid:13]: Received More Keys
*Nov 15 12:23:54.547: SSS PM [uid:13]: Handling Authorization Check
*Nov 15 12:23:54.547: SSS PM [uid:13]: Handling AAA service Authorization
*Nov 15 12:23:54.547: SSS PM [uid:13]: Sending authorization request for 'xyz.com'

*Nov 15 12:23:54.547: SSS AAA AUTHOR [uid:13]:Event <make request>, state changed
from idle to authorizing
*Nov 15 12:23:54.547: SSS AAA AUTHOR [uid:13]:Authorizing key xyz.com
```

```
*Nov 15 12:23:54.547: SSS AAA AUTHOR [uid:13]:AAA request sent for key xyz.com
*Nov 15 12:23:54.551: SSS AAA AUTHOR [uid:13]:Received an AAA pass
*Nov 15 12:23:54.551: SSS AAA AUTHOR [uid:13]:Event <found service>, state changed
from authorizing to complete
*Nov 15 12:23:54.551: SSS AAA AUTHOR [uid:13]:Found service info for key xyz.com
*Nov 15 12:23:54.551: SSS AAA AUTHOR [uid:13]:Event <free request>, state changed
from complete to terminal
*Nov 15 12:23:54.551: SSS AAA AUTHOR [uid:13]:Free request
*Nov 15 12:23:54.551: SSS PM [uid:13]: Handling Service Direction
*Nov 15 12:23:54.551: SSS PM [uid:13]: Policy reply - Forwarding
*Nov 15 12:23:54.551: SSS MGR [uid:13]: Got reply Forwarding from PM
*Nov 15 12:23:54.551: SSS MGR [uid:13]: Handling Connect-Service event
*Nov 15 12:23:54.551: VPDN CALL [uid:13]: Event connect req, state changed from idle
to connecting
*Nov 15 12:23:54.551: VPDN CALL [uid:13]: Requesting connection
*Nov 15 12:23:54.551: VPDN CALL [uid:13]: Call request sent
*Nov 15 12:23:54.551: VPDN MGR [uid:13]: Event client connect, state changed from
idle to connecting
*Nov 15 12:23:54.551: VPDN MGR [uid:13]: Initiating compulsory connection to
199.11.8.2
*Nov 15 12:23:54.551:   Tnl/Sn61510/7 L2TP: Session FS enabled
*Nov 15 12:23:54.551:   Tnl/Sn61510/7 L2TP: Session state change from idle to
wait-for-tunnel
*Nov 15 12:23:54.551: uid:13  Tnl/Sn61510/7 L2TP: Create session
*Nov 15 12:23:54.551: uid:13  Tnl/Sn61510/7 L2TP: O ICRQ to rp1 9264/0
*Nov 15 12:23:54.551: [13]PPPoE 10: Access IE nas port called
*Nov 15 12:23:54.555:   Tnl61510 L2TP: Control channel retransmit delay set to 1
seconds
*Nov 15 12:23:54.555: uid:13  Tnl/Sn61510/7 L2TP: Session state change from
wait-for-tunnel to wait-reply
*Nov 15 12:23:54.555: [13]PPPoE 10: State LCP_NEGO    Event PPP_FWDING
*Nov 15 12:23:54.559: uid:13  Tnl/Sn61510/7 L2TP: O ICCN to rp1 9264/13586
*Nov 15 12:23:54.559:   Tnl61510 L2TP: Control channel retransmit delay set to 1
seconds
*Nov 15 12:23:54.559: uid:13  Tnl/Sn61510/7 L2TP: Session state change from
wait-reply to established
*Nov 15 12:23:54.559: uid:13  Tnl/Sn61510/7 L2TP: VPDN session up
*Nov 15 12:23:54.559: VPDN MGR [uid:13]: Event peer connected, state changed from
connecting to connected
*Nov 15 12:23:54.559: VPDN MGR [uid:13]: Succeed to forward nobody@xyz.com
*Nov 15 12:23:54.559: VPDN MGR [uid:13]: accounting start sent
*Nov 15 12:23:54.559: VPDN CALL [uid:13]: Event connect ok, state changed from
connecting to connected
*Nov 15 12:23:54.559: VPDN CALL [uid:13]: Connection succeeded
*Nov 15 12:23:54.559: SSS MGR [uid:13]: Handling Service-Connected event
*Nov 15 12:23:54.559: ppp13 PPP: Phase is FORWARDED, Session Forwarded
*Nov 15 12:23:54.559: [13]PPPoE 10: State LCP_NEGO    Event PPP_FWDED
*Nov 15 12:23:54.563: [13]PPPoE 10: data path set to SSS Switch
*Nov 15 12:23:54.563: [13]PPPoE 10: Connected Forwarded
```

Troubleshoot the Subscriber Service Switch on the LAC—Authorization Failure Example

The following is sample output indicating call failure due to authorization failure:

```
*Nov 15 12:37:24.535: PPPoE 0: I PADI R:0000.0c14.71d0 L:ffff.ffff.ffff 1/32
ATM4/0.132
*Nov 15 12:37:24.535: PPPoE 0: O PADO R:0000.0c14.71d0 L:00b0.c2e9.c870 1/32
ATM4/0.132
*Nov 15 12:37:24.539: PPPoE 0: I PADR R:0000.0c14.71d0 L:00b0.c2e9.c870 1/32
ATM4/0.132
*Nov 15 12:37:24.539: PPPoE : encap string prepared
*Nov 15 12:37:24.539: [18]PPPoE 15: Access IE handle allocated
*Nov 15 12:37:24.539: [18]PPPoE 15: pppoe SSS switch updated
```

```
*Nov 15 12:37:24.539: PPPoE 15: AAA pppoe_aaa_acct_get_retrieved_attrs
*Nov 15 12:37:24.539: [18]PPPoE 15: AAA pppoe_aaa_acct_get_nas_port_details
*Nov 15 12:37:24.539: [18]PPPoE 15: AAA pppoe_aaa_acct_get_dynamic_attrs
*Nov 15 12:37:24.539: [18]PPPoE 15: AAA pppoe_aaa_acct_get_dynamic_attrs
*Nov 15 12:37:24.539: [18]PPPoE 15: AAA unique ID allocated
*Nov 15 12:37:24.539: [18]PPPoE 15: No AAA accounting method list
*Nov 15 12:37:24.539: [18]PPPoE 15: Service request sent to SSS
*Nov 15 12:37:24.539: [18]PPPoE 15: Created R:00b0.c2e9.c870 L:0000.0c14.71d0 1/32
ATM4/0.132
*Nov 15 12:37:24.559: SSS INFO: Element type is Access-Type, long value is 3
*Nov 15 12:37:24.559: SSS INFO: Element type is Switch-Id, long value is -738197487
*Nov 15 12:37:24.559: SSS INFO: Element type is Nasport, ptr value is 63C0E590
*Nov 15 12:37:24.559: SSS INFO: Element type is AAA-Id, long value is 19
*Nov 15 12:37:24.559: SSS INFO: Element type is AccIe-Hdl, ptr value is 5B000011
*Nov 15 12:37:24.559: SSS MGR [uid:18]: Handling Policy Authorize (1 pending
sessions)
*Nov 15 12:37:24.559: SSS PM [uid:18]: RM/VPDN disabled: RM/VPDN author not needed
*Nov 15 12:37:24.559: SSS PM [uid:18]: Received Service Request
*Nov 15 12:37:24.559: SSS PM [uid:18]: Handling Authorization Check
*Nov 15 12:37:24.559: SSS PM [uid:18]: Policy requires 'Unauth-User' key
*Nov 15 12:37:24.559: SSS PM [uid:18]: Policy reply - Need more keys
*Nov 15 12:37:24.559: SSS MGR [uid:18]: Got reply Need-More-Keys from PM
*Nov 15 12:37:24.559: SSS MGR [uid:18]: Handling More-Keys event
*Nov 15 12:37:24.559: [18]PPPoE 15: State REQ_NASPORT    Event MORE_KEYS
*Nov 15 12:37:24.559: [18]PPPoE 15: O PADS R:0000.0c14.71d0 L:00b0.c2e9.c870 1/32
ATM4/0.132
*Nov 15 12:37:24.559: ppp18 PPP: Using default call direction
*Nov 15 12:37:24.559: ppp18 PPP: Treating connection as a dedicated line
*Nov 15 12:37:24.559: ppp18 PPP: Phase is ESTABLISHING, Active Open
*Nov 15 12:37:24.559: ppp18 LCP: O CONFREQ [Closed] id 1 len 19
*Nov 15 12:37:24.559: ppp18 LCP:    MRU 1492 (0x010405D4)
*Nov 15 12:37:24.559: ppp18 LCP:    AuthProto CHAP (0x0305C22305)
*Nov 15 12:37:24.559: ppp18 LCP:    MagicNumber 0xB0F8A971 (0x0506B0F8A971)
*Nov 15 12:37:24.559: [18]PPPoE 15: State START_PPP    Event DYN_BIND
*Nov 15 12:37:24.559: [18]PPPoE 15: data path set to PPP
*Nov 15 12:37:24.563: ppp18 LCP: I CONFREQ [REQsent] id 1 len 14
*Nov 15 12:37:24.563: ppp18 LCP:    MRU 1492 (0x010405D4)
*Nov 15 12:37:24.563: ppp18 LCP:    MagicNumber 0x0023A93E (0x05060023A93E)
*Nov 15 12:37:24.563: ppp18 LCP: O CONFACK [REQsent] id 1 len 14
*Nov 15 12:37:24.563: ppp18 LCP:    MRU 1492 (0x010405D4)
*Nov 15 12:37:24.563: ppp18 LCP:    MagicNumber 0x0023A93E (0x05060023A93E)
*Nov 15 12:37:26.523: ppp18 LCP: I CONFREQ [ACKsent] id 2 len 14
*Nov 15 12:37:26.523: ppp18 LCP:    MRU 1492 (0x010405D4)
*Nov 15 12:37:26.523: ppp18 LCP:    MagicNumber 0x0023A93E (0x05060023A93E)
*Nov 15 12:37:26.523: ppp18 LCP: O CONFACK [ACKsent] id 2 len 14
*Nov 15 12:37:26.527: ppp18 LCP:    MRU 1492 (0x010405D4)
*Nov 15 12:37:26.527: ppp18 LCP:    MagicNumber 0x0023A93E (0x05060023A93E)
*Nov 15 12:37:26.575: ppp18 LCP: TIMEout: State ACKsent
*Nov 15 12:37:26.575: ppp18 LCP: O CONFREQ [ACKsent] id 2 len 19
*Nov 15 12:37:26.575: ppp18 LCP:    MRU 1492 (0x010405D4)
*Nov 15 12:37:26.575: ppp18 LCP:    AuthProto CHAP (0x0305C22305)
*Nov 15 12:37:26.575: ppp18 LCP:    MagicNumber 0xB0F8A971 (0x0506B0F8A971)
*Nov 15 12:37:26.575: ppp18 LCP: I CONFACK [ACKsent] id 2 len 19
*Nov 15 12:37:26.575: ppp18 LCP:    MRU 1492 (0x010405D4)
*Nov 15 12:37:26.575: ppp18 LCP:    AuthProto CHAP (0x0305C22305)
*Nov 15 12:37:26.575: ppp18 LCP:    MagicNumber 0xB0F8A971 (0x0506B0F8A971)
*Nov 15 12:37:26.575: ppp18 LCP: State is Open
*Nov 15 12:37:26.575: ppp18 PPP: Phase is AUTHENTICATING, by this end
*Nov 15 12:37:26.575: ppp18 CHAP: O CHALLENGE id 1 len 25 from "7200"
*Nov 15 12:37:26.579: ppp18 CHAP: I RESPONSE id 1 len 38 from "nobody@xyz.com"
*Nov 15 12:37:26.579: ppp18 PPP: Phase is FORWARDING, Attempting Forward
*Nov 15 12:37:26.579: SSS INFO: Element type is Unauth-User, string value is
nobody@xyz.com
*Nov 15 12:37:26.579: SSS INFO: Element type is AccIe-Hdl, ptr value is 5B000011
```

```
*Nov 15 12:37:26.579: SSS INFO: Element type is AAA-Id, long value is 19
*Nov 15 12:37:26.579: SSS INFO: Element type is Access-Type, long value is 0
*Nov 15 12:37:26.579: SSS MGR [uid:18]: Handling Policy Authorize (1 pending
sessions)
*Nov 15 12:37:26.579: SSS PM [uid:18]: Received More Keys
*Nov 15 12:37:26.579: SSS PM [uid:18]: Handling Authorization Check
*Nov 15 12:37:26.579: SSS PM [uid:18]: Handling AAA service Authorization
*Nov 15 12:37:26.579: SSS PM [uid:18]: Sending authorization request for 'xyz.com'

*Nov 15 12:37:26.579: SSS AAA AUTHOR [uid:18]:Event <make request>, state changed
from idle to authorizing
*Nov 15 12:37:26.579: SSS AAA AUTHOR [uid:18]:Authorizing key xyz.com
*Nov 15 12:37:26.579: SSS AAA AUTHOR [uid:18]:AAA request sent for key xyz.com
*Nov 15 12:37:26.587: SSS AAA AUTHOR [uid:18]:Received an AAA failure
*Nov 15 12:37:26.587: SSS AAA AUTHOR [uid:18]:Event <service not found>, state
changed from authorizing to complete
*Nov 15 12:37:26.587: SSS AAA AUTHOR [uid:18]:No service authorization info found
*Nov 15 12:37:26.587: SSS AAA AUTHOR [uid:18]:Event <free request>, state changed
from complete to terminal
*Nov 15 12:37:26.587: SSS AAA AUTHOR [uid:18]:Free request
*Nov 15 12:37:26.587: SSS PM [uid:18]: Handling Next Authorization Check
*Nov 15 12:37:26.587: SSS PM [uid:18]: Default policy: SGF author not needed
*Nov 15 12:37:26.587: SSS PM [uid:18]: Handling Default Service
*Nov 15 12:37:26.587: SSS PM [uid:18]: Policy reply - Local terminate
*Nov 15 12:37:26.591: SSS MGR [uid:18]: Got reply Local-Term from PM
*Nov 15 12:37:26.591: SSS MGR [uid:18]: Handling Send-Client-Local-Term event
*Nov 15 12:37:26.591: ppp18 PPP: Phase is AUTHENTICATING, Unauthenticated User
*Nov 15 12:37:26.595: ppp18 CHAP: O FAILURE id 1 len 25 msg is "Authentication
failed"
*Nov 15 12:37:26.599: ppp18 PPP: Sending Acct Event[Down] id[13]
*Nov 15 12:37:26.599: ppp18 PPP: Phase is TERMINATING
*Nov 15 12:37:26.599: ppp18 LCP: O TERMREQ [Open] id 3 len 4
*Nov 15 12:37:26.599: ppp18 LCP: State is Closed
*Nov 15 12:37:26.599: ppp18 PPP: Phase is DOWN
*Nov 15 12:37:26.599: ppp18 PPP: Phase is TERMINATING
*Nov 15 12:37:26.599: [18]PPPoE 15: State LCP_NEGO    Event PPP_DISCNCT
*Nov 15 12:37:26.599: [18]PPPoE 15: O PADT R:0000.0c14.71d0 L:00b0.c2e9.c870 1/32
ATM4/0.132
*Nov 15 12:37:26.599: [18]PPPoE 15: Destroying R:0000.0c14.71d0 L:00b0.c2e9.c870
1/32 ATM4/0.132
*Nov 15 12:37:26.599: [18]PPPoE 15: AAA account stopped
*Nov 15 12:37:26.599: SSS MGR [uid:18]: Processing a client disconnect
*Nov 15 12:37:26.599: SSS MGR [uid:18]: Handling Send-Service-Disconnect event
```

Troubleshoot the Subscriber Service Switch on the LAC—Authentication Failure Example

The following is sample output indicating call failure due to authentication failure at the LNS:

```
*Nov 15 12:45:02.067: PPPoE 0: I PADI R:0000.0c14.71d0 L:ffff.ffff.ffff 1/32
ATM4/0.132
*Nov 15 12:45:02.071: PPPoE 0: O PADO R:0000.0c14.71d0 L:00b0.c2e9.c870 1/32
ATM4/0.132
*Nov 15 12:45:02.071: PPPoE 0: I PADR R:0000.0c14.71d0 L:00b0.c2e9.c870 1/32
ATM4/0.132
*Nov 15 12:45:02.071: PPPoE : encap string prepared
*Nov 15 12:45:02.071: [21]PPPoE 18: Access IE handle allocated
*Nov 15 12:45:02.071: [21]PPPoE 18: pppoe SSS switch updated
*Nov 15 12:45:02.071: PPPoE 18: AAA pppoe_aaa_acct_get_retrieved_attrs
*Nov 15 12:45:02.071: [21]PPPoE 18: AAA pppoe_aaa_acct_get_nas_port_details
*Nov 15 12:45:02.071: [21]PPPoE 18: AAA pppoe_aaa_acct_get_dynamic_attrs
*Nov 15 12:45:02.071: [21]PPPoE 18: AAA pppoe_aaa_acct_get_dynamic_attrs
*Nov 15 12:45:02.071: [21]PPPoE 18: AAA unique ID allocated
*Nov 15 12:45:02.071: [21]PPPoE 18: No AAA accounting method list
```

```
*Nov 15 12:45:02.071: [21]PPPoE 18: Service request sent to SSS
*Nov 15 12:45:02.071: [21]PPPoE 18: Created R:00b0.c2e9.c870 L:0000.0c14.71d0 1/32
ATM4/0.132
*Nov 15 12:45:02.091: SSS INFO: Element type is Access-Type, long value is 3
*Nov 15 12:45:02.091: SSS INFO: Element type is Switch-Id, long value is 1946157076
*Nov 15 12:45:02.091: SSS INFO: Element type is Nasport, ptr value is 63B34170
*Nov 15 12:45:02.091: SSS INFO: Element type is AAA-Id, long value is 22
*Nov 15 12:45:02.091: SSS INFO: Element type is AccIe-Hdl, ptr value is 71000014
*Nov 15 12:45:02.091: SSS MGR [uid:21]: Handling Policy Authorize (1 pending
sessions)
*Nov 15 12:45:02.091: SSS PM [uid:21]: RM/VPDN disabled: RM/VPDN author not needed
*Nov 15 12:45:02.091: SSS PM [uid:21]: Received Service Request
*Nov 15 12:45:02.091: SSS PM [uid:21]: Handling Authorization Check
*Nov 15 12:45:02.091: SSS PM [uid:21]: Policy requires 'Unauth-User' key
*Nov 15 12:45:02.091: SSS PM [uid:21]: Policy reply - Need more keys
*Nov 15 12:45:02.091: SSS MGR [uid:21]: Got reply Need-More-Keys from PM
*Nov 15 12:45:02.091: SSS MGR [uid:21]: Handling More-Keys event
*Nov 15 12:45:02.091: [21]PPPoE 18: State REQ_NASPORT    Event MORE_KEYS
*Nov 15 12:45:02.091: [21]PPPoE 18: O PADS R:0000.0c14.71d0 L:00b0.c2e9.c870 1/32
ATM4/0.132
*Nov 15 12:45:02.091: ppp21 PPP: Using default call direction
*Nov 15 12:45:02.091: ppp21 PPP: Treating connection as a dedicated line
*Nov 15 12:45:02.091: ppp21 PPP: Phase is ESTABLISHING, Active Open
*Nov 15 12:45:02.091: ppp21 LCP: O CONFREQ [Closed] id 1 len 19
*Nov 15 12:45:02.091: ppp21 LCP:    MRU 1492 (0x010405D4)
*Nov 15 12:45:02.091: ppp21 LCP:    AuthProto CHAP (0x0305C22305)
*Nov 15 12:45:02.091: ppp21 LCP:    MagicNumber 0xB0FFA4D8 (0x0506B0FFA4D8)
*Nov 15 12:45:02.091: [21]PPPoE 18: State START_PPP    Event DYN_BIND
*Nov 15 12:45:02.091: [21]PPPoE 18: data path set to PPP
*Nov 15 12:45:02.095: ppp21 LCP: I CONFREQ [REQsent] id 1 len 14
*Nov 15 12:45:02.095: ppp21 LCP:    MRU 1492 (0x010405D4)
*Nov 15 12:45:02.095: ppp21 LCP:    MagicNumber 0x002AA481 (0x0506002AA481)
*Nov 15 12:45:02.095: ppp21 LCP: O CONFACK [REQsent] id 1 len 14
*Nov 15 12:45:02.095: ppp21 LCP:    MRU 1492 (0x010405D4)
*Nov 15 12:45:02.095: ppp21 LCP:    MagicNumber 0x002AA481 (0x0506002AA481)
*Nov 15 12:45:02.315:   Tnl41436 L2TP: I StopCCN from rp1 tnl 31166
*Nov 15 12:45:02.315:   Tnl41436 L2TP: Shutdown tunnel
*Nov 15 12:45:02.315:   Tnl41436 L2TP: Tunnel state change from no-sessions-left to
idle
*Nov 15 12:45:04.055: ppp21 LCP: I CONFREQ [ACKsent] id 2 len 14
*Nov 15 12:45:04.055: ppp21 LCP:    MRU 1492 (0x010405D4)
*Nov 15 12:45:04.059: ppp21 LCP:    MagicNumber 0x002AA481 (0x0506002AA481)
*Nov 15 12:45:04.059: ppp21 LCP: O CONFACK [ACKsent] id 2 len 14
*Nov 15 12:45:04.059: ppp21 LCP:    MRU 1492 (0x010405D4)
*Nov 15 12:45:04.059: ppp21 LCP:    MagicNumber 0x002AA481 (0x0506002AA481)
*Nov 15 12:45:04.079: ppp21 LCP: TIMEout: State ACKsent
*Nov 15 12:45:04.079: ppp21 LCP: O CONFREQ [ACKsent] id 2 len 19
*Nov 15 12:45:04.079: ppp21 LCP:    MRU 1492 (0x010405D4)
*Nov 15 12:45:04.079: ppp21 LCP:    AuthProto CHAP (0x0305C22305)
*Nov 15 12:45:04.079: ppp21 LCP:    MagicNumber 0xB0FFA4D8 (0x0506B0FFA4D8)
*Nov 15 12:45:04.079: ppp21 LCP: I CONFACK [ACKsent] id 2 len 19
*Nov 15 12:45:04.079: ppp21 LCP:    MRU 1492 (0x010405D4)
*Nov 15 12:45:04.079: ppp21 LCP:    AuthProto CHAP (0x0305C22305)
*Nov 15 12:45:04.079: ppp21 LCP:    MagicNumber 0xB0FFA4D8 (0x0506B0FFA4D8)
*Nov 15 12:45:04.079: ppp21 LCP: State is Open
*Nov 15 12:45:04.079: ppp21 PPP: Phase is AUTHENTICATING, by this end
*Nov 15 12:45:04.079: ppp21 CHAP: O CHALLENGE id 1 len 25 from "7200"
*Nov 15 12:45:04.083: ppp21 CHAP: I RESPONSE id 1 len 38 from "nobody@xyz.com"
*Nov 15 12:45:04.083: ppp21 PPP: Phase is FORWARDING, Attempting Forward
*Nov 15 12:45:04.083: SSS INFO: Element type is Unauth-User, string value is
nobody@xyz.com
*Nov 15 12:45:04.083: SSS INFO: Element type is AccIe-Hdl, ptr value is 71000014
*Nov 15 12:45:04.083: SSS INFO: Element type is AAA-Id, long value is 22
*Nov 15 12:45:04.083: SSS INFO: Element type is Access-Type, long value is 0
```

```
*Nov 15 12:45:04.083: SSS MGR [uid:21]: Handling Policy Authorize (1 pending
sessions)
*Nov 15 12:45:04.083: SSS PM [uid:21]: Received More Keys
*Nov 15 12:45:04.083: SSS PM [uid:21]: Handling Authorization Check
*Nov 15 12:45:04.083: SSS PM [uid:21]: Handling AAA service Authorization
*Nov 15 12:45:04.083: SSS PM [uid:21]: Sending authorization request for 'xyz.com'

*Nov 15 12:45:04.083: SSS AAA AUTHOR [uid:21]:Event <make request>, state changed
from idle to authorizing
*Nov 15 12:45:04.083: SSS AAA AUTHOR [uid:21]:Authorizing key xyz.com
*Nov 15 12:45:04.083: SSS AAA AUTHOR [uid:21]:AAA request sent for key xyz.com
*Nov 15 12:45:04.095: SSS AAA AUTHOR [uid:21]:Received an AAA pass
*Nov 15 12:45:04.095: SSS AAA AUTHOR [uid:21]:Event <found service>, state changed
from authorizing to complete
*Nov 15 12:45:04.095: SSS AAA AUTHOR [uid:21]:Found service info for key xyz.com
*Nov 15 12:45:04.095: SSS AAA AUTHOR [uid:21]:Event <free request>, state changed
from complete to terminal
*Nov 15 12:45:04.095: SSS AAA AUTHOR [uid:21]:Free request
*Nov 15 12:45:04.095: SSS PM [uid:21]: Handling Service Direction
*Nov 15 12:45:04.095: SSS PM [uid:21]: Policy reply - Forwarding
*Nov 15 12:45:04.095: SSS MGR [uid:21]: Got reply Forwarding from PM
*Nov 15 12:45:04.099: SSS MGR [uid:21]: Handling Connect-Service event
*Nov 15 12:45:04.099: VPDN CALL [uid:21]: Event connect req, state changed from idle
to connecting
*Nov 15 12:45:04.099: VPDN CALL [uid:21]: Requesting connection
*Nov 15 12:45:04.099: VPDN CALL [uid:21]: Call request sent
*Nov 15 12:45:04.099: VPDN MGR [uid:21]: Event client connect, state changed from
idle to connecting
*Nov 15 12:45:04.099: VPDN MGR [uid:21]: Initiating compulsory connection to
199.11.8.2
*Nov 15 12:45:04.099:   Tnl/Sn31399/10 L2TP: Session FS enabled
*Nov 15 12:45:04.099:   Tnl/Sn31399/10 L2TP: Session state change from idle to
wait-for-tunnel
*Nov 15 12:45:04.099: uid:21  Tnl/Sn31399/10 L2TP: Create session
*Nov 15 12:45:04.099:   Tnl31399 L2TP: SM State idle
*Nov 15 12:45:04.099:   Tnl31399 L2TP: O SCCRQ
*Nov 15 12:45:04.099:   Tnl31399 L2TP: Control channel retransmit delay set to 1
seconds
*Nov 15 12:45:04.099:   Tnl31399 L2TP: Tunnel state change from idle to
wait-ctl-reply
*Nov 15 12:45:04.099:   Tnl31399 L2TP: SM State wait-ctl-reply
*Nov 15 12:45:04.099: [21]PPPoE 18: State LCP_NEGO    Event PPP_FWDING
*Nov 15 12:45:04.107:   Tnl31399 L2TP: I SCCRP from rp1
*Nov 15 12:45:04.107:   Tnl31399 L2TP: Got a challenge from remote peer, rp1
*Nov 15 12:45:04.107:   Tnl31399 L2TP: Got a response from remote peer, rp1
*Nov 15 12:45:04.107:   Tnl31399 L2TP: Tunnel Authentication success
*Nov 15 12:45:04.107:   Tnl31399 L2TP: Tunnel state change from wait-ctl-reply to
established
*Nov 15 12:45:04.107:   Tnl31399 L2TP: O SCCCN  to rp1 tnlid 9349
*Nov 15 12:45:04.107:   Tnl31399 L2TP: Control channel retransmit delay set to 1
seconds
*Nov 15 12:45:04.107:   Tnl31399 L2TP: SM State established
*Nov 15 12:45:04.107: uid:21  Tnl/Sn31399/10 L2TP: O ICRQ to rp1 9349/0
*Nov 15 12:45:04.107: [21]PPPoE 18: Access IE nas port called
*Nov 15 12:45:04.107: uid:21  Tnl/Sn31399/10 L2TP: Session state change from
wait-for-tunnel to wait-reply
*Nov 15 12:45:04.115: uid:21  Tnl/Sn31399/10 L2TP: O ICCN to rp1 9349/13589
*Nov 15 12:45:04.115:   Tnl31399 L2TP: Control channel retransmit delay set to 1
seconds
*Nov 15 12:45:04.115: uid:21  Tnl/Sn31399/10 L2TP: Session state change from
wait-reply to established
*Nov 15 12:45:04.115: uid:21  Tnl/Sn31399/10 L2TP: VPDN session up
*Nov 15 12:45:04.115: VPDN MGR [uid:21]: Event peer connected, state changed from
connecting to connected
```

```
*Nov 15 12:45:04.115: VPDN MGR [uid:21]: Succeed to forward nobody@xyz.com
*Nov 15 12:45:04.115: VPDN MGR [uid:21]: accounting start sent
*Nov 15 12:45:04.115: [21]PPPoE 18: AAA pppoe_aaa_acct_get_dynamic_attrs
*Nov 15 12:45:04.115: [21]PPPoE 18: AAA pppoe_aaa_acct_get_dynamic_attrs
*Nov 15 12:45:04.115: VPDN CALL [uid:21]: Event connect ok, state changed from
connecting to connected
*Nov 15 12:45:04.115: VPDN CALL [uid:21]: Connection succeeded
*Nov 15 12:45:04.115: SSS MGR [uid:21]: Handling Service-Connected event
*Nov 15 12:45:04.115: ppp21 PPP: Phase is FORWARDED, Session Forwarded
*Nov 15 12:45:04.115: [21]PPPoE 18: State LCP_NEGO    Event PPP_FWDED
*Nov 15 12:45:04.115: [21]PPPoE 18: data path set to SSS Switch
*Nov 15 12:45:04.119: [21]PPPoE 18: Connected Forwarded
*Nov 15 12:45:04.119: ppp21 PPP: Process pending packets
*Nov 15 12:45:04.139: uid:21  Tnl/Sn31399/10 L2TP: Result code(2): 2: Call
disconnected, refer to error msg
*Nov 15 12:45:04.139:      Error code(6): Vendor specific
*Nov 15 12:45:04.139:      Optional msg: Locally generated disconnect
*Nov 15 12:45:04.139: uid:21  Tnl/Sn31399/10 L2TP: I CDN from rp1 tnl 9349, cl
13589
01:06:21: %VPDN-6-CLOSED: L2TP LNS 199.11.8.2 closed  user nobody@xyz.com; Result
2, Error 6, Locally generated disconnect
*Nov 15 12:45:04.139: uid:21  Tnl/Sn31399/10 L2TP: disconnect (L2X) IETF:
18/host-request Ascend: 66/VPDN Local PPP Disconnect
*Nov 15 12:45:04.139: uid:21  Tnl/Sn31399/10 L2TP: Destroying session
*Nov 15 12:45:04.139: uid:21  Tnl/Sn31399/10 L2TP: Session state change from
established to idle
*Nov 15 12:45:04.139: VPDN MGR [uid:21]: Event peer disconnect, state changed from
connected to disconnected
*Nov 15 12:45:04.139: VPDN MGR [uid:21]: Remote disconnected nobody@xyz.com
*Nov 15 12:45:04.139: VPDN MGR [uid:21]: accounting stop sent
*Nov 15 12:45:04.139:   Tnl31399 L2TP: Tunnel state change from established to
no-sessions-left
*Nov 15 12:45:04.143:   Tnl31399 L2TP: No more sessions in tunnel, shutdown (likely)
in 15 seconds
*Nov 15 12:45:04.143: VPDN CALL [uid:21]: Event server disc, state changed from
connected to disconnected
*Nov 15 12:45:04.143: VPDN CALL [uid:21]: Server disconnected call
*Nov 15 12:45:04.143: VPDN CALL [uid:21]: Event free req, state changed from
disconnected to terminal
*Nov 15 12:45:04.143: VPDN CALL [uid:21]: Free request
*Nov 15 12:45:04.143: SSS MGR [uid:21]: Handling Send Client Disconnect
*Nov 15 12:45:04.143: [21]PPPoE 18: State CNCT_FWDED    Event SSS_DISCNCT
*Nov 15 12:45:04.143: ppp21 PPP: Sending Acct Event[Down] id[16]
*Nov 15 12:45:04.143: ppp21 PPP: Phase is TERMINATING
*Nov 15 12:45:04.143: ppp21 LCP: State is Closed
*Nov 15 12:45:04.143: ppp21 PPP: Phase is DOWN
*Nov 15 12:45:04.143: [21]PPPoE 18: O PADT R:0000.0c14.71d0 L:00b0.c2e9.c870 1/32
ATM4/0.132
*Nov 15 12:45:04.143: [21]PPPoE 18: Destroying R:0000.0c14.71d0 L:00b0.c2e9.c870
1/32 ATM4/0.132
*Nov 15 12:45:04.143: [21]PPPoE 18: AAA pppoe_aaa_acct_get_dynamic_attrs
*Nov 15 12:45:04.143: [21]PPPoE 18: AAA pppoe_aaa_acct_get_dynamic_attrs
*Nov 15 12:45:04.143: [21]PPPoE 18: AAA account stopped
*Nov 15 12:45:14.139:   Tnl31399 L2TP: I StopCCN from rp1 tnl 9349
*Nov 15 12:45:14.139:   Tnl31399 L2TP: Shutdown tunnel
*Nov 15 12:45:14.139:   Tnl31399 L2TP: Tunnel state change from no-sessions-left
```

Troubleshoot the Subscriber Service Switch at the LNS—Normal Operation Example

The following example shows the **debug** commands used and sample output indicating normal operation of the Subscriber Service Switch on the LNS:

```
Router# debug sss event
Router# debug sss error
Router# debug sss fsm
Router# debug ppp negotiation
Router# debug vpdn l2x-events
Router# debug vpdn l2x-errors
Router# debug vpdn sss events
Router# debug vpdn sss errors
Router# debug vpdn sss fsm

SSS:
  SSS events debugging is on
  SSS error debugging is on
  SSS fsm debugging is on
PPP:
  PPP protocol negotiation debugging is on

VPN:
  L2X protocol events debugging is on
  L2X protocol errors debugging is on
  VPDN SSS events debugging is on
  VPDN SSS errors debugging is on
  VPDN SSS FSM debugging is on

3d17h:   Tnl9264 L2TP: I ICRQ from server1 tnl 61510
3d17h:   Tnl/Sn9264/13586 L2TP: Session FS enabled
3d17h:   Tnl/Sn9264/13586 L2TP: Session state change from idle to wait-connect
3d17h:   Tnl/Sn9264/13586 L2TP: New session created
3d17h:   Tnl/Sn9264/13586 L2TP: O ICRP to server1 61510/7
3d17h:   Tnl9264 L2TP: Control channel retransmit delay set to 1 seconds
3d17h:   Tnl/Sn9264/13586 L2TP: I ICCN from server1 tnl 61510, cl 7
3d17h: nobody@xyz.com  Tnl/Sn9264/13586 L2TP: Session state change from
wait-connect to wait-for-service-selection
3d17h: VPDN SSS []: Event start sss, state changed from IDLE to SSS
3d17h: VPDN SSS [uid:707]: Service request sent to SSS
3d17h: SSS INFO: Element type is Access-Type, long value is 4
3d17h: SSS INFO: Element type is Switch-Id, long value is 1493172561
3d17h: SSS INFO: Element type is Tunnel-Name, string value is server1
3d17h: SSS INFO: Element type is Can-SIP-Redirect, long value is 1
3d17h: SSS INFO: Element type is AAA-Id, long value is 16726
3d17h: SSS INFO: Element type is AccIe-Hdl, ptr value is D1000167
3d17h: SSS MGR [uid:707]: Event service-request, state changed from wait-for-req to
wait-for-auth
3d17h: SSS MGR [uid:707]: Handling Policy Authorize (1 pending sessions)
3d17h: SSS PM [uid:707]: RM/VPDN disabled: RM/VPDN author not needed
3d17h: SSS PM [uid:707]: Multihop disabled: AAA author not needed
3d17h: SSS PM [uid:707]: Multihop disabled: SGF author not needed
3d17h: SSS PM [uid:707]: No more authorization methods left to try, providing
default service
3d17h: SSS PM [uid:707]: Received Service Request
3d17h: SSS PM [uid:707]: Event <found>, State: initial-req to end
3d17h: SSS PM [uid:707]: Handling Service Direction
3d17h: SSS PM [uid:707]: Policy reply - Local terminate
3d17h: SSS MGR [uid:707]: Got reply Local-Term from PM
3d17h: SSS MGR [uid:707]: Event policy-connect local, state changed from
wait-for-auth to connected
3d17h: SSS MGR [uid:707]: Handling Send-Client-Local-Term event
3d17h: VPDN SSS [uid:707]: Event connect local, state changed from SSS to PPP
3d17h: ppp707 PPP: Phase is ESTABLISHING
3d17h: ppp707 LCP: I FORCED rcvd CONFACK len 15
3d17h: ppp707 LCP:    MRU 1492 (0x010405D4)
3d17h: ppp707 LCP:    AuthProto CHAP (0x0305C22305)
3d17h: ppp707 LCP:    MagicNumber 0xB0EC4557 (0x0506B0EC4557)
3d17h: ppp707 LCP: I FORCED sent CONFACK len 10
```

```
3d17h: ppp707 LCP:    MRU 1492 (0x010405D4)
3d17h: ppp707 LCP:    MagicNumber 0x0017455D (0x05060017455D)
3d17h: ppp707 PPP: Phase is FORWARDING, Attempting Forward
3d17h: VPDN SSS [uid:707]: Event dyn bind resp, state changed from PPP to PPP
3d17h: ppp707 PPP: Phase is AUTHENTICATING, Unauthenticated User
3d17h: ppp707 PPP: Phase is FORWARDING, Attempting Forward
3d17h: VPDN SSS [uid:707]: Event connect local, state changed from PPP to PPP
3d17h: VPDN SSS [Vi4.2]: Event vaccess resp, state changed from PPP to PPP
3d17h: VPDN SSS [Vi4.2]: Event stat bind resp, state changed from PPP to CNCT
3d17h: Vi4.2  Tnl/Sn9264/13586 L2TP: Session state change from
wait-for-service-selection to established
3d17h: Vi4.2 PPP: Phase is AUTHENTICATING, Authenticated User
3d17h: Vi4.2 CHAP: O SUCCESS id 1 len 4
3d17h: Vi4.2 PPP: Phase is UP
3d17h: Vi4.2 IPCP: O CONFREQ [Closed] id 1 len 10
3d17h: Vi4.2 IPCP:    Address 172.18.0.0 (0x030681010000)
3d17h: Vi4.2 PPP: Process pending packets
3d17h: Vi4.2 IPCP: I CONFREQ [REQsent] id 1 len 10
3d17h: Vi4.2 IPCP:    Address 0.0.0.0 (0x030600000000)
3d17h: Vi4.2 AAA/AUTHOR/IPCP: Start.  Her address 0.0.0.0, we want 0.0.0.0
3d17h: Vi4.2 AAA/AUTHOR/IPCP: Done.  Her address 0.0.0.0, we want 0.0.0.0
3d17h: Vi4.2 IPCP: Pool returned 10.1.1.3
3d17h: Vi4.2 IPCP: O CONFNAK [REQsent] id 1 len 10
3d17h: Vi4.2 IPCP:    Address 10.1.1.3 (0x03065B010103)
3d17h: Vi4.2 IPCP: I CONFACK [REQsent] id 1 len 10
3d17h: Vi4.2 IPCP:    Address 172.18.0.0 (0x030681010000)
3d17h: Vi4.2 IPCP: I CONFREQ [ACKrcvd] id 2 len 10
3d17h: Vi4.2 IPCP:    Address 10.1.1.3 (0x03065B010103)
3d17h: Vi4.2 IPCP: O CONFACK [ACKrcvd] id 2 len 10
3d17h: Vi4.2 IPCP:    Address 10.1.1.3 (0x03065B010103)
3d17h: Vi4.2 IPCP: State is Open
3d17h: Vi4.2 IPCP: Install route to 10.1.1.3
```

Troubleshoot the Subscriber Service Switch at the LNS—Tunnel Failure Example

The following is sample output indicating tunnel failure on the LNS:

```
3d17h: L2TP: I SCCRQ from server1 tnl 31399
3d17h:   Tnl9349 L2TP: Got a challenge in SCCRQ, server1
3d17h:   Tnl9349 L2TP: New tunnel created for remote server1, address 199.11.8.1
3d17h:   Tnl9349 L2TP: O SCCRP  to server1 tnlid 31399
3d17h:   Tnl9349 L2TP: Control channel retransmit delay set to 1 seconds
3d17h:   Tnl9349 L2TP: Tunnel state change from idle to wait-ctl-reply
3d17h:   Tnl9349 L2TP: I SCCCN from server1 tnl 31399
3d17h:   Tnl9349 L2TP: Got a Challenge Response in SCCCN from server1
3d17h:   Tnl9349 L2TP: Tunnel Authentication success
3d17h:   Tnl9349 L2TP: Tunnel state change from wait-ctl-reply to established
3d17h:   Tnl9349 L2TP: SM State established
3d17h:   Tnl9349 L2TP: I ICRQ from server1 tnl 31399
3d17h:   Tnl/Sn9349/13589 L2TP: Session FS enabled
3d17h:   Tnl/Sn9349/13589 L2TP: Session state change from idle to wait-connect
3d17h:   Tnl/Sn9349/13589 L2TP: New session created
3d17h:   Tnl/Sn9349/13589 L2TP: O ICRP to server1 31399/10
3d17h:   Tnl9349 L2TP: Control channel retransmit delay set to 1 seconds
3d17h:   Tnl/Sn9349/13589 L2TP: I ICCN from server1 tnl 31399, cl 10
3d17h: nobody@xyz.com  Tnl/Sn9349/13589 L2TP: Session state change from
wait-connect to wait-for-service-selection
3d17h: VPDN SSS []: Event start sss, state changed from IDLE to SSS
3d17h: VPDN SSS [uid:709]: Service request sent to SSS
3d17h: SSS INFO: Element type is Access-Type, long value is 4
3d17h: SSS INFO: Element type is Switch-Id, long value is -1912602284
3d17h: SSS INFO: Element type is Tunnel-Name, string value is server1
3d17h: SSS INFO: Element type is Can-SIP-Redirect, long value is 1
```

```
3d17h: SSS INFO: Element type is AAA-Id, long value is 16729
3d17h: SSS INFO: Element type is AccIe-Hdl, ptr value is 8D00016A
3d17h: SSS MGR [uid:709]: Event service-request, state changed from wait-for-req to
wait-for-auth
3d17h: SSS MGR [uid:709]: Handling Policy Authorize (1 pending sessions)
3d17h: SSS PM [uid:709]: RM/VPDN disabled: RM/VPDN author not needed
3d17h: SSS PM [uid:709]: Multihop disabled: AAA author not needed
3d17h: SSS PM [uid:709]: Multihop disabled: SGF author not needed
3d17h: SSS PM [uid:709]: No more authorization methods left to try, providing default
service
3d17h: SSS PM [uid:709]: Received Service Request
3d17h: SSS PM [uid:709]: Event <found>, State: initial-req to end
3d17h: SSS PM [uid:709]: Handling Service Direction
3d17h: SSS PM [uid:709]: Policy reply - Local terminate
3d17h: SSS MGR [uid:709]: Got reply Local-Term from PM
3d17h: SSS MGR [uid:709]: Event policy-connect local, state changed from
wait-for-auth to connected
3d17h: SSS MGR [uid:709]: Handling Send-Client-Local-Term event
3d17h: VPDN SSS [uid:709]: Event connect local, state changed from SSS to PPP
3d17h: ppp709 PPP: Phase is ESTABLISHING
3d17h: ppp709 LCP: I FORCED rcvd CONFACK len 15
3d17h: ppp709 LCP:    MRU 1492 (0x010405D4)
3d17h: ppp709 LCP:    AuthProto CHAP (0x0305C22305)
3d17h: ppp709 LCP:    MagicNumber 0xB0FFA4D8 (0x0506B0FFA4D8)
3d17h: ppp709 LCP: I FORCED sent CONFACK len 10
3d17h: ppp709 LCP:    MRU 1492 (0x010405D4)
3d17h: ppp709 LCP:    MagicNumber 0x002AA481 (0x0506002AA481)
3d17h: ppp709 PPP: Phase is FORWARDING, Attempting Forward
3d17h: VPDN SSS [uid:709]: Event dyn bind resp, state changed from PPP to PPP
3d17h: ppp709 PPP: Phase is AUTHENTICATING, Unauthenticated User
3d17h: ppp709 CHAP: O FAILURE id 1 len 25 msg is "Authentication failed"
3d17h: ppp709 PPP: Sending Acct Event[Down] id[4159]
3d17h: ppp709 PPP: Phase is TERMINATING
3d17h: ppp709 LCP: O TERMREQ [Open] id 1 len 4
3d17h: ppp709 LCP: State is Closed
3d17h: ppp709 PPP: Phase is DOWN
3d17h: ppp709 PPP: Phase is TERMINATING
3d17h: VPDN SSS [uid:709]: Event peer disc, state changed from PPP to DSC
3d17h: nobody@xyz.com  Tnl/Sn9349/13589 L2TP: disconnect (AAA) IETF:
17/user-error Ascend: 26/PPP CHAP Fail
3d17h: nobody@xyz.com  Tnl/Sn9349/13589 L2TP: O CDN to server1 31399/10
3d17h:   Tnl9349 L2TP: Control channel retransmit delay set to 1 seconds
3d17h: nobody@xyz.com  Tnl/Sn9349/13589 L2TP: Destroying session
3d17h: nobody@xyz.com  Tnl/Sn9349/13589 L2TP: Session state change from
wait-for-service-selection to idle
3d17h: VPDN SSS [uid:709]: Event vpdn disc, state changed from DSC to END
3d17h:   Tnl9349 L2TP: Tunnel state change from established to no-sessions-left
3d17h:   Tnl9349 L2TP: No more sessions in tunnel, shutdown (likely) in 10 seconds
3d17h: SSS MGR [uid:709]: Processing a client disconnect
3d17h: SSS MGR [uid:709]: Event client-disconnect, state changed from connected to
end
3d17h: SSS MGR [uid:709]: Handling Send-Service-Disconnect event
3d17h:   Tnl9349 L2TP: O StopCCN  to server1 tnlid 31399
3d17h:   Tnl9349 L2TP: Control channel retransmit delay set to 1 seconds
3d17h:   Tnl9349 L2TP: Tunnel state change from no-sessions-left to shutting-down
3d17h:   Tnl9349 L2TP: Shutdown tunnel
```

Additional References

For additional information related to Subscriber Service Switch, refer to the following references:

Related Documents

Related Topic	Document Title
"Virtual Templates, Profiles, and Networks" chapter "PPP Configuration" chapter	• *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2
VPDN and PPP commands: complete command syntax, command mode, defaults, usage guidelines, and examples	• *Cisco IOS Dial Technologies Command Reference*, Release 12.2
"Authentication, Authorization, and Accounting (AAA)" chapter	• *Cisco IOS Security Configuration Guide*, Release 12.2
AAA commands: complete command syntax, command mode, defaults, usage guidelines, and examples	• *Cisco IOS Security Command Reference*, Release 12.2
"Configuring Broadband Access: PPP and Routed Bridging Encapsulations" chapter	• *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.2
PPPoE and PPPoA commands: complete command syntax, command mode, defaults, usage guidelines, and examples	• *Cisco IOS Wide-Area Networking Command Reference*, Release 12.2
L2TP tunnel service authorization	• *L2TP Tunnel Service Authorization Enhancements*
LLID feature	• *RADIUS Logical Line ID*

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

RFCs[1]	Title
RFC 2341	*Cisco Layer Two Forwarding (Protocol) L2F*
RFC2661	*Layer Two Tunneling Protocol L2TP*
RFC 2516	*A Method for Transmitting PPP Over Ethernet (PPPoE)* (PPPoE Discovery)

1. Not all supported RFCs are listed.

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, tools, and lots more. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **atm pppatm passive**
- **clear pppatm interface atm**
- **clear pppoe**
- **debug pppatm**
- **debug sss aaa authorization event**
- **debug sss aaa authorization fsm**
- **debug sss error**
- **debug sss event**
- **debug sss fsm**
- **multihop hostname**
- **show pppatm summary**

- **show pppatm trace**
- **show sss session**
- **show vpdn session**
- **subscriber access**
- **subscriber authorization enable**
- **vpdn authorize domain**
- **vpn service**

Glossary

Note Refer to the *Internetworking Terms and Acronyms* for terms not included in this glossary.

L2TP access concentrator—See LAC**.**

L2TP network server—See LNS**.**

Layer 2 Forwarding Protocol—See L2F.

Layer 2 Tunneling Protocol—See L2TP.

L2F—Layer 2 Forwarding Protocol, as described in RFC 2341.

L2TP—Layer 2 Tunneling Protocol, as described in RFC2661.

LAC—L2TP access concentrator. The peer of the LNS that serves as one endpoint of an L2TP tunnel. The client connects to the LAC directly and PPP frames are tunneled over L2TP to the LNS.

LLID—logical line identification. An alphanumeric string that is a minimum of one character and a maximum of 253 characters in length, which is the logical identification of a subscriber line. LLID is maintained in a RADIUS server customer profile database. This customer profile database is connected to a LAC and is separate from the RADIUS server that the LAC and LNS use for the authentication and authorization of incoming users. When the customer profile database receives a preauthorization request from the LAC, the server sends the LLID to the LAC as the Calling-Station-ID attribute (attribute 31).

LNS—L2TP network server. A node that acts as one side of an L2TP tunnel endpoint. A peer to the LAC. The logical termination point of a PPP session that is being tunneled by the LAC.

locally terminated—Refers to a PPP session that will no longer be forwarded. The PPP framing and negotiation ends at this point to allow Layer 3 processing of the framed packet.

logical line identification—See LLID.

MLP—Multilink PPP. Method of splitting, recombining, and sequencing datagrams across multiple logical data links.

multihop—Traditional term for accepting a PPP session from a virtual private dial-up network (VPDN) protocol such as L2TP or L2F and tunneling it back out via L2TP or L2F.

Multilink PPP—See MLP.

NAS—network access server, the L2F equivalent of a LAC.

network access server—See NAS.

Packet Data Serving Node—See PDSN.

PDSN—Packet Data Serving Node. Provides access to the Internet, intranets and applications servers for mobile stations utilizing a cdma2000 Radio Access Network (RAN). Acting as an access gateway, PDSN provides simple IP and mobile IP access, foreign agent support, and packet transport for virtual private networking. It acts as a client for authentication, authorization, and accounting (AAA) servers and provides mobile stations with a gateway to the IP network. Mobility differentiates Cisco's PDSN from the traditional routed network. With PDSN, the host can move and, therefore, there must be a way to forward packets to it. Cisco's PDSN solution offers a secure way to provide packet data services to mobile stations.

PPPoA—PPP over ATM.

PPPoE—PPP over Ethernet (RFC 2516). Refers to the signaling protocol defined within PPPoE and as the encapsulation method. Sometimes ambiguous as to whether this term refers to PPPoE over ATM or PPPoE directly over Ethernet.

PPPoEoA—PPP over Ethernet over ATM. The most common form of PPPoE into LAC.

PPPoEoE—This acronym is often used to differentiate actual PPP over Ethernet from PPPoEoA, but does not imply that PPP is being encapsulated in two levels of Ethernet.

PPPoX—Either PPPoEoA, PPPoEoE, or PPPoA.

service—What a subscriber is provided. Example of a service may be tunneling an incoming PPP session from PPPoE to another location via L2TP, or local termination of the PPPoE session. In addition, QoS parameters or other specifics may be bundled as part of an identified service.

service key—An elemental piece of information about a subscriber that is used to determine a service for that subscriber. Examples include the PPP authenticated name, ATM VPI/VCI, the PPPoE service name, and so on.

SGF—stack group forwarding. Process used by the Stack Group Bidding Protocol (SGBP) to authorize a session for forwarding.

stack group forwarding—See SGF.

subscriber—An access user that typically uses a PPP-based service. This service may be PPPoE, PPPoA, L2TP, PPP over a dial infrastructure such as ISDN or analog, and so on.

Tunnel Authentication via RADIUS on Tunnel Terminator

Feature History

Release	Modification
12.2(15)B	This feature was introduced on the Cisco 6400 series, Cisco 7200 series, and Cisco 7400 series.

- Feature Overview, page 375
- Supported Platforms, page 377
- Supported Standards, MIBs, and RFCs, page 378
- Prerequisites, page 378
- Configuration Tasks, page 378
- Configuration Examples, page 380
- Command Reference, page 381
- Glossary, page 382

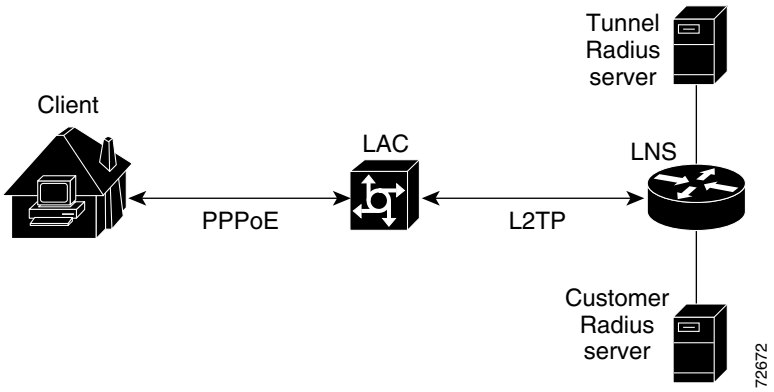
Feature Overview

The Tunnel Authentication via RADIUS on Tunnel Terminator feature allows the Layer 2 Tunneling Protocol (L2TP) network server (LNS) to perform remote authentication and authorization with RADIUS on incoming L2TP access concentrator (LAC) dialin connection requests. This feature also allows the L2TP LAC to perform remote authentication and authorization with RADIUS on incoming L2TP LNS dialout connection requests.

Before the introduction of this feature, the LNS could only perform L2TP tunnel authentication and authorization locally. These processes can be difficult to maintain across numerous LNSs, especially if the number of virtual private dialup network (VPDN) groups is large, because the LAC information must be configured under the VPDN group configurations of the LNS. Remote RADIUS authentication and authorization allows users to store the LAC configurations on the RADIUS server, thereby avoiding the need to store information locally. Thus, the new LAC information can be added to the RADIUS server as necessary, and the group of LNSs can authenticate and authorize by using a common user database on RADIUS.

Figure 16 and the corresponding steps explain how this feature works.

Figure 16 LNS Remote RADIUS Tunnel Authentication and Authorization for L2TP Dialin Calls Topology



- After the LNS receives a start-control-connection request (SCCRQ), it starts tunnel authentication and submits a request to RADIUS with the LAC hostname and the dummy password "cisco." (If the LNS determines that authorization should be performed locally, it will search the VPDN group configurations.)

Note To change the dummy password, use the **vpdn tunnel authorization password** command.

- If the password sent by the LNS matches the password that is configured in the RADIUS server, the server will return attribute 90 (Tunnel-Client-Auth-ID) and attribute 69 (Tunnel-Password) after the LAC information is located. Otherwise, the RADIUS server replies back with an access-reject, and the LNS drops the tunnel.
- The LNS will check for the following attribute information from the RADIUS reply:
  - Attribute 90 (Tunnel-Client-Auth-ID), which is used as the LAC hostname. If this attribute does not match the LAC hostname, the tunnel will be dropped.
  - Attribute 69 (Tunnel-Password), which is used for the L2TP CHAP-like authentication shared secret. This attribute is compared against the LAC challenge attribute-value pair (AVP) that was received in the SCCRQ. If this attribute does not match the AVP, the tunnel will be dropped.
- If both attributes match, the L2TP tunnel will be established. Thereafter, you can proceed with PPP negotiation and authentication with the remote client.

Note PPP remote authentication is done to a potential different customer RADIUS server by a separate access-request/access-accept sequence. The tunnel authorization may be done by a different tunnel RADIUS server.

New RADIUS Attributes

To help implement this feature, the following two new Cisco-specific RADIUS attributes have been introduced:

- Cisco:Cisco-Avpair = “vpdn:dout-dialer = <LAC dialer number>”—Specifies which LAC dialer to use on the LAC for a dialout configuration.
- Cisco:Cisco-Avpair = “vpdn:vpdn-vtemplate = <vtemplate number>”—Specifies the virtual template number that will be used for cloning on the LNS for a dialin configuration. (This attribute is the RADIUS counterpart for the virtual-template under the vpdn-group configuration.)

Note The service-type in the RADIUS user’s profile for the tunnel initiator should always be set to “Outbound.”

Benefits

This feature allows tunnel authentication and authorization to occur via a remote RADIUS server instead of local configuration on the tunnel terminator. Thus, users no longer have to configure LAC or LNS data in a VPDN group when an LNS or LAC is configured for incoming dialin or dialout L2TP tunnel termination; this information can now be added to a remote RADIUS server, providing a more manageable and scalable solution for L2TP tunnel authentication and authorization on the tunnel terminator.

Restrictions

This is applicable only to L2TP; that is, protocols such as (Layer 2 Forwarding) L2F and Point-to-Point Tunneling Protocol (PPTP) are not supported.

Related Documents

- The chapter “Configuring Virtual Private Networks” in the *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2
- The appendix “RADIUS Attributes” in the *Cisco IOS Security Configuration Guide*, Release 12.2
- *Cisco IOS Dial Technologies Command Reference*, Release 12.2
- *Cisco IOS Security Command Reference*, Release 12.2

Supported Platforms

- Cisco 6400 series
- Cisco 7200 series
- Cisco 7400 series

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFC 2868, *RADIUS Attributes for Tunnel Protocol Support*

Prerequisites

Before configuring this feature, you should define a RADIUS server group. For information on completing this task, refer to the chapter "Configuring RADIUS" in the *Cisco IOS Security Configuration Guide*, Release 12.2.

Configuration Tasks

See the following sections for configuration tasks for the Tunnel Authentication via RADIUS on Tunnel Terminator feature. Each task in the list is identified as either required or optional.

- Configuring the LNS or LAC for Remote RADIUS Tunnel Authentication and Authorization (required)
- Verifying Remote RADIUS Tunnel Authentication and Authorization Configurations (optional)

Configuring the LNS or LAC for Remote RADIUS Tunnel Authentication and Authorization

To configure an LNS or LAC for incoming dialin or dialout L2TP tunnel termination, use the following commands in global configuration:

	Command	Purpose
Step 1	`Router(config)# aaa authorization network {default | list-name} method1 [method2...]`	Defines an AAA authorization method list for network services.
Step 2	`Router(config)# vpdn tunnel authorization network {method-list-name | default}`	Specifies the AAA authorization method list that will be used for remote tunnel hostname-based authorization. • If the *list-name* argument was specified in the **aaa authorization** command, you use that list name here. • If the default keyword was specified in the **aaa authorization** command, you must choose that keyword, which specifies the default authorization methods that are listed with the **aaa authorization** command here.
Step 3	`Router(config)# vpdn tunnel authorization virtual-template vtemplate-number`	(Optional) Selects the default virtual template from which to clone virtual access interfaces.
Step 4	`Router(config)# vpdn tunnel authorization password password`	(Optional) Configures a "dummy" password for the RADIUS authorization request to retrieve the tunnel configuration that is based on the remote tunnel hostname. **Note** If this command is not enabled, the password will always be "cisco."

Verifying Remote RADIUS Tunnel Authentication and Authorization Configurations

To verify that the L2TP tunnel is up, use the **show vpdn tunnel** command in EXEC mode. One tunnel and one session must be set up.

```
Router# show vpdn tunnel

L2TP Tunnel and Session Information Total tunnels 1 sessions 1
LocID RemID Remote Name   State  Remote Address  Port  Sessions VPDN Group
4571  61568 csidtw13 est    10.0.195.4      1701  1        ?

LocID RemID TunID Intf          Username               State  Last Chg
4     11    4571  Vi4.1         csidtw9@cisco.com       est    00:02:29

%No active L2F tunnels
%No active PPTP tunnels
%No active PPPoE tunnels
```

To verify that the AAA authorization RADIUS server is configured on the LNS and that the LNS can receive attributes 90 and 69 from the RADIUS server, perform the following steps:

Step 1 Enable the **debug radius** command on the LNS.

Step 2 Enable the **show logging** command on the LNS and ensure that "access-accept" is in the output and that attributes 90 and 69 can be seen in the RADIUS reply.

```
00:32:56: RADIUS: Received from id 21645/5 172.19.192.50:1645, Access-Accept, len 81
00:32:56: RADIUS:  authenticator 73 2B 1B C2 33 71 93 19 - 62 AC 3E BE 0D 13 14 85
00:32:56: RADIUS:  Service-Type        [6]   6   Outbound                  [5]
00:32:56: RADIUS:  Tunnel-Type         [64]  6   00:L2TP                   [3]
00:32:56: RADIUS:  Tunnel-Medium-Type  [65]  6   00:IPv4                   [1]
00:32:56: RADIUS:  Tunnel-Client-Auth-I[90]  6   00:"csidtw13"
00:32:56: RADIUS:  Tunnel-Password     [69]  8   *
00:32:56: RADIUS:  Vendor, Cisco       [26]  29
00:32:56: RADIUS:   Cisco AVpair       [1]   23  "vpdn:vpdn-vtemplate=1"
```

To verify that the L2TP tunnel has been established and that the LNS can perform PPP negotiation and authentication with the remote client, perform the following steps:

Step 1 Enable the **debug ppp negotiation** and **debug ppp authentication** commands on LNS.

Step 2 Enable the **show logging** command on LNS and observe that LNS receives a PPP CHAP challenge and then sends a PPP CHAP "SUCCESS" to the client.

```
00:38:50: ppp3 PPP: Received LOGIN Response from AAA = PASS
00:38:50: ppp3 PPP: Phase is FORWARDING, Attempting Forward
00:38:50: Vi4.1  Tnl/Sn4571/4 L2TP: Session state change from wait-for-service-selection
to established
00:38:50: Vi4.1 PPP: Phase is AUTHENTICATING, Authenticated User
00:38:50: Vi4.1 CHAP: O SUCCESS id 1 len 4
```

Step 3 After PPP authentication is successful, observe from the debug output that PPP negotiation has started, that the LNS has received LCP (IPCP) packets, and that negotiation is successful.

```
00:38:50: Vi4.1 IPCP: State is Open
00:38:50: Vi4.1 IPCP: Install route to 200.1.1.4
```

Configuration Examples

This section provides the following configuration examples:

- L2TP Network Server (LNS) Configuration Example
- RADIUS User Profile for Remote RADIUS Tunnel Authentication Example

L2TP Network Server (LNS) Configuration Example

The following example shows how to configure the LNS to enable remote RADIUS tunnel authentication and authorization:

```
! Define a RADIUS server group
aaa group server radius VPDN-group
 server 64.102.48.91 auth-port 1645 acct-port 1646
!
! RADIUS configurations only
aaa authorization network mymethodlist group VPDN-Group
vpdn tunnel authorization network mymethodlist
vpdn tunnel authorization virtual-template 10
```

RADIUS User Profile for Remote RADIUS Tunnel Authentication Example

The following are examples of RADIUS user profiles for the LNS to terminate L2TP tunnels from a LAC. In the first user profile, the final line is optional if the **vpdn tunnel authorization virtual-template** command is used. Also, the first RADIUS user profile is for L2TP dialin, and the second RADIUS user profile is for L2TP dialout.

The service-type in the RADIUS user's profile for the tunnel initiator should always be set to "Outbound."

```
csidtw13  Password = "cisco"
        Service-Type = Outbound,
        Tunnel-Type = :0:L2TP,
        Tunnel-Medium-Type = :0:IP,
        Tunnel-Client-Auth-ID = :0:"csidtw13",
        Tunnel-Password = :0:"cisco"
        Cisco:Cisco-Avpair = "vpdn:vpdn-vtemplate=1"

csidtw1 Password = "cisco"
        Service-Type = Outbound,
        Tunnel-Type = :0:L2TP,
        Tunnel-Medium-Type = :0:IP,
        Tunnel-Client-Auth-ID = :0:"csidtw1",
        Tunnel-Password = :0:"cisco"
        Cisco:Cisco-Avpair = "vpdn:dout-dialer=2"
```

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **vpdn tunnel authorization network**
- **vpdn tunnel authorization password**
- **vpdn tunnel authorization virtual-template**

Glossary

L2TP—Layer 2 Tunnel Protocol. A Layer 2 tunneling protocol that enables an ISP or other access service to create a virtual tunnel to link customer remote sites or remote users with corporate home networks. In particular, a network access server (NAS) at the ISP point of presence (POP) exchanges PPP messages with the remote users and communicates by L2F or L2TP requests and responses with the customer tunnel server to set up tunnels.

LAC—L2TP access concentrator. A network access server (NAS) to which the client directly connects and through which PPP frames are tunneled to the L2TP network server (LNS). The LAC need only implement the media over which L2TP is to operate to pass traffic to one or more LNSs. The LAC may tunnel any protocol carried within PPP. The LAC initiates incoming calls and receives outgoing calls. A LAC is analogous to an L2F network access server.

LNS—L2TP network server. A termination point for L2TP tunnels and an access point where PPP frames are processed and passed to higher layer protocols. An LNS can operate on any platform that terminates PPP. The LNS handles the server side of the L2TP protocol. L2TP relies only on the single medium over which L2TP tunnels arrive. The LNS initiates outgoing calls and receives incoming calls. An LNS is analogous to a home gateway in L2F technology.

TACACS+

This part consists of the following:

- Configuring TACACS+
- Per VRF for TACACS+ Servers

Configuring TACACS+

This chapter discusses how to enable and configure TACACS+, which provides detailed accounting information and flexible administrative control over authentication and authorization processes. TACACS+ is facilitated through AAA and can be enabled only through AAA commands.

For a complete description of the TACACS+ commands used in this chapter, refer to the chapter "TACACS+ Commands" in the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature, or refer to the software release notes for a specific release. For more information, see the"Finding Additional Feature Support Information" section on page cxvii in the chapter Using Cisco IOS Software for Release 12.4.

In This Chapter

This chapter includes the following sections:

- About TACACS+
- TACACS+ Operation
- TACACS+ Configuration Task List
- TACACS+ AV Pairs
- TACACS+ Configuration Examples

About TACACS+

TACACS+ is a security application that provides centralized validation of users attempting to gain access to a router or network access server. TACACS+ services are maintained in a database on a TACACS+ daemon running, typically, on a UNIX or Windows NT workstation. You must have access to and must configure a TACACS+ server before the configured TACACS+ features on your network access server are available.

TACACS+ provides for separate and modular authentication, authorization, and accounting facilities. TACACS+ allows for a single access control server (the TACACS+ daemon) to provide each service—authentication, authorization, and accounting—independently. Each service can be tied into its own database to take advantage of other services available on that server or on the network, depending on the capabilities of the daemon.

The goal of TACACS+ is to provide a methodology for managing multiple network access points from a single management service. The Cisco family of access servers and routers and the Cisco IOS user interface (for both routers and access servers) can be network access servers.

Network access points enable traditional “dumb” terminals, terminal emulators, workstations, personal computers (PCs), and routers in conjunction with suitable adapters (for example, modems or ISDN adapters) to communicate using protocols such as Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), Compressed SLIP (CSLIP), or AppleTalk Remote Access (ARA) protocol. In other words, a network access server provides connections to a single user, to a network or subnetwork, and to interconnected networks. The entities connected to the network through a network access server are called *network access clients*; for example, a PC running PPP over a voice-grade circuit is a network access client. TACACS+, administered through the AAA security services, can provide the following services:

- Authentication—Provides complete control of authentication through login and password dialog, challenge and response, messaging support.

  The authentication facility provides the ability to conduct an arbitrary dialog with the user (for example, after a login and password are provided, to challenge a user with a number of questions, like home address, mother’s maiden name, service type, and social security number). In addition, the TACACS+ authentication service supports sending messages to user screens. For example, a message could notify users that their passwords must be changed because of the company’s password aging policy.

- Authorization—Provides fine-grained control over user capabilities for the duration of the user’s session, including but not limited to setting autocommands, access control, session duration, or protocol support. You can also enforce restrictions on what commands a user may execute with the TACACS+ authorization feature.

- Accounting—Collects and sends information used for billing, auditing, and reporting to the TACACS+ daemon. Network managers can use the accounting facility to track user activity for a security audit or to provide information for user billing. Accounting records include user identities, start and stop times, executed commands (such as PPP), number of packets, and number of bytes.

The TACACS+ protocol provides authentication between the network access server and the TACACS+ daemon, and it ensures confidentiality because all protocol exchanges between a network access server and a TACACS+ daemon are encrypted.

You need a system running TACACS+ daemon software to use the TACACS+ functionality on your network access server.

Cisco makes the TACACS+ protocol specification available as a draft RFC for those customers interested in developing their own TACACS+ software.

TACACS+ Operation

When a user attempts a simple ASCII login by authenticating to a network access server using TACACS+, the following process typically occurs:

1. When the connection is established, the network access server will contact the TACACS+ daemon to obtain a username prompt, which is then displayed to the user. The user enters a username and the network access server then contacts the TACACS+ daemon to obtain a password prompt. The network access server displays the password prompt to the user, the user enters a password, and the password is then sent to the TACACS+ daemon.

Note TACACS+ allows an arbitrary conversation to be held between the daemon and the user until the daemon receives enough information to authenticate the user. This is usually done by prompting for a username and password combination, but may include other items, such as mother's maiden name, all under the control of the TACACS+ daemon.

2. The network access server will eventually receive one of the following responses from the TACACS+ daemon:
   a. ACCEPT—The user is authenticated and service may begin. If the network access server is configured to requite authorization, authorization will begin at this time.
   b. REJECT—The user has failed to authenticate. The user may be denied further access, or will be prompted to retry the login sequence depending on the TACACS+ daemon.
   c. ERROR—An error occurred at some time during authentication. This can be either at the daemon or in the network connection between the daemon and the network access server. If an ERROR response is received, the network access server will typically try to use an alternative method for authenticating the user.
   d. CONTINUE—The user is prompted for additional authentication information.
3. A PAP login is similar to an ASCII login, except that the username and password arrive at the network access server in a PAP protocol packet instead of being typed in by the user, so the user is not prompted. PPP CHAP logins are also similar in principle.

Following authentication, the user will also be required to undergo an additional authorization phase, if authorization has been enabled on the network access server. Users must first successfully complete TACACS+ authentication before proceeding to TACACS+ authorization.

4. If TACACS+ authorization is required, the TACACS+ daemon is again contacted and it returns an ACCEPT or REJECT authorization response. If an ACCEPT response is returned, the response will contain data in the form of attributes that are used to direct the EXEC or NETWORK session for that user, determining services that the user can access.

   Services include the following:
   a. Telnet, rlogin, Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), or EXEC services
   b. Connection parameters, including the host or client IP address, access list, and user timeouts

TACACS+ Configuration Task List

To configure your router to support TACACS+, you must perform the following tasks:

- Use the **aaa new-model** global configuration command to enable AAA. AAA must be configured if you plan to use TACACS+. For more information about using the **aaa new-model** command, refer to the chapter "AAA Overview".
- Use the **tacacs-server host** command to specify the IP address of one or more TACACS+ daemons. Use the **tacacs-server key** command to specify an encryption key that will be used to encrypt all exchanges between the network access server and the TACACS+ daemon. This same key must also be configured on the TACACS+ daemon.
- Use the **aaa authentication** global configuration command to define method lists that use TACACS+ for authentication. For more information about using the **aaa authentication** command, refer to the chapter "Configuring Authentication".

- Use **line** and **interface** commands to apply the defined method lists to various interfaces. For more information, refer to the chapter "Configuring Authentication".
- If needed, use the **aaa authorization** global command to configure authorization for the network access server. Unlike authentication, which can be configured per line or per interface, authorization is configured globally for the entire network access server. For more information about using the **aaa authorization** command, refer to the "Configuring Authorization" chapter.
- If needed, use the **aaa accounting** command to enable accounting for TACACS+ connections. For more information about using the **aaa accounting** command, refer to the "Configuring Accounting" chapter.

To configure TACACS+, perform the tasks in the following sections:

- Identifying the TACACS+ Server Host (Required)
- Setting the TACACS+ Authentication Key (Optional)
- Configuring AAA Server Groups (Optional)
- Configuring AAA Server Group Selection Based on DNIS (Optional)
- Specifying TACACS+ Authentication (Required)
- Specifying TACACS+ Authorization (Optional)
- Specifying TACACS+ Accounting (Optional)

For TACACS+ configuration examples using the commands in this chapter, refer to the "TACACS+ Configuration Examples" section at the end of the this chapter.

Identifying the TACACS+ Server Host

The **tacacs-server host** command enables you to specify the names of the IP host or hosts maintaining a TACACS+ server. Because the TACACS+ software searches for the hosts in the order specified, this feature can be useful for setting up a list of preferred daemons.

To specify a TACACS+ host, use the following command in global configuration mode:

Command	Purpose
`Router(config)# tacacs-server host hostname [single-connection] [port integer] [timeout integer] [key string]`	Specifies a TACACS+ host.

Using the **tacacs-server host** command, you can also configure the following options:

- Use the **single-connection** keyword to specify single-connection (only valid with CiscoSecure Release 1.0.1 or later). Rather than have the router open and close a TCP connection to the daemon each time it must communicate, the single-connection option maintains a single open connection between the router and the daemon. This is more efficient because it allows the daemon to handle a higher number of TACACS operations.

Note The daemon must support single-connection mode for this to be effective, otherwise the connection between the network access server and the daemon will lock up or you will receive spurious errors.

- Use the **port** *integer* argument to specify the TCP port number to be used when making connections to the TACACS+ daemon. The default port number is 49.
- Use the **timeout** *integer* argument to specify the period of time (in seconds) the router will wait for a response from the daemon before it times out and declares an error.

Note Specifying the timeout value with the **tacacs-server host** command overrides the default timeout value set with the **tacacs-server timeout** command for this server only.

- Use the **key** *string* argument to specify an encryption key for encrypting and decrypting all traffic between the network access server and the TACACS+ daemon.

Note Specifying the encryption key with the **tacacs-server host** command overrides the default key set by the global configuration **tacacs-server key** command for this server only.

Because some of the parameters of the **tacacs-server host** command override global settings made by the **tacacs-server timeout** and **tacacs-server key** commands, you can use this command to enhance security on your network by uniquely configuring individual TACACS+ connections.

Setting the TACACS+ Authentication Key

To set the global TACACS+ authentication key and encryption key, use the following command in global configuration mode:

Command	Purpose
`Router(config)# tacacs-server key key`	Sets the encryption key to match that used on the TACACS+ daemon.

Note You must configure the same key on the TACACS+ daemon for encryption to be successful.

Configuring AAA Server Groups

Configuring the router to use AAA server groups provides a way to group existing server hosts. This allows you to select a subset of the configured server hosts and use them for a particular service. A server group is used in conjunction with a global server-host list. The server group lists the IP addresses of the selected server hosts.

Server groups can include multiple host entries as long as each entry has a unique IP address. If two different host entries in the server group are configured for the same service—for example, accounting—the second host entry configured acts as fail-over backup to the first one. Using this example, if the first host entry fails to provide accounting services, the network access server will try the second host entry for accounting services. (The TACACS+ host entries will be tried in the order in which they are configured.)

To define a server host with a server group name, enter the following commands starting in global configuration mode. The listed server must exist in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# tacacs-server host name [single-connection] [port integer] [timeout integer] [key string]`	Specifies and defines the IP address of the server host before configuring the AAA server-group. Refer to the "Identifying the TACACS+ Server Host" section of this chapter for more information on the **tacacs-server host** command.
Step 2	`Router(config-if)# aaa group server {radius | tacacs+} group-name`	Defines the AAA server-group with a group name. All members of a group must be the same type; that is, RADIUS or TACACS+. This command puts the router in server group subconfiguration mode.
Step 3	`Router(config-sg)# server ip-address [auth-port port-number] [acct-port port-number]`	Associates a particular TACACS+ server with the defined server group. Use the **auth-port** *port-number* option to configure a specific UDP port solely for authentication. Use the **acct-port** *port-number* option to configure a specific UDP port solely for accounting. Repeat this step for each TACACS+ server in the AAA server group. **Note** Each server in the group must be defined previously using the **tacacs-server host** command.

Configuring AAA Server Group Selection Based on DNIS

Cisco IOS software allows you to authenticate users to a particular AAA server group based on the Dialed Number Identification Service (DNIS) number of the session. Any phone line (a regular home phone or a commercial T1/PRI line) can be associated with several phone numbers. The DNIS number identifies the number that was called to reach you.

For example, suppose you want to share the same phone number with several customers, but you want to know which customer is calling before you pick up the phone. You can customize how you answer the phone because DNIS allows you to know which customer is calling when you answer.

Cisco routers with either ISDN or internal modems can receive the DNIS number. This functionality allows users to assign different TACACS+ server groups for different customers (that is, different TACACS+ servers for different DNIS numbers). Additionally, using server groups you can specify the same server group for AAA services or a separate server group for each AAA service.

Cisco IOS software provides the flexibility to implement authentication and accounting services in several ways:

- Globally—AAA services are defined using global configuration access list commands and applied in general to all interfaces on a specific network access server.
- Per interface—AAA services are defined using interface configuration commands and applied specifically to the interface being configured on a specific network access server.
- DNIS mapping—You can use DNIS to specify an AAA server to supply AAA services.

Because AAA configuration methods can be configured simultaneously, Cisco has established an order of precedence to determine which server or groups of servers provide AAA services. The order of precedence is as follows:

- Per DNIS—If you configure the network access server to use DNIS to identify which server group provides AAA services, then this method takes precedence over any additional AAA selection method.
- Per interface—If you configure the network access server per interface to use access lists to determine how a server provides AAA services, this method takes precedence over any global configuration AAA access lists.
- Globally—If you configure the network access server by using global AAA access lists to determine how the security server provides AAA services, this method has the lowest precedence.

Note Prior to configuring AAA Server Group Selection Based on DNIS, you must configure the remote security servers associated with each AAA server group. See the sections "Identifying the TACACS+ Server Host" and "Configuring AAA Server Groups" in this chapter.

To configure the router to select a particular AAA server group based on the DNIS of the server group, configure DNIS mapping. To map a server group with a group name with DNIS number, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa dnis map enable`	Enables DNIS mapping.
Step 2	`Router(config)# aaa dnis map` *dnis-number* `authentication ppp group` *server-group-name*	Maps a DNIS number to a defined AAA server group; the servers in this server group are being used for authentication.
Step 3	`Router(config)# aaa dnis map` *dnis-number* `accounting network [none | start-stop | stop-only] group` *server-group-name*	Maps a DNIS number to a defined AAA server group; the servers in this server group are being used for accounting.

Specifying TACACS+ Authentication

After you have identified the TACACS+ daemon and defined an associated TACACS+ encryption key, you must define method lists for TACACS+ authentication. Because TACACS+ authentication is operated via AAA, you need to issue the **aaa authentication** command, specifying TACACS+ as the authentication method. For more information, refer to the chapter "Configuring Authentication."

Specifying TACACS+ Authorization

AAA authorization enables you to set parameters that restrict a user's access to the network. Authorization via TACACS+ may be applied to commands, network connections, and EXEC sessions. Because TACACS+ authorization is facilitated through AAA, you must issue the **aaa authorization** command, specifying TACACS+ as the authorization method. For more information, refer to the chapter "Configuring Authorization."

Specifying TACACS+ Accounting

AAA accounting enables you to track the services users are accessing as well as the amount of network resources they are consuming. Because TACACS+ accounting is facilitated through AAA, you must issue the **aaa accounting** command, specifying TACACS+ as the accounting method. For more information, refer to the chapter "Configuring Accounting."

TACACS+ AV Pairs

The network access server implements TACACS+ authorization and accounting functions by transmitting and receiving TACACS+ attribute-value (AV) pairs for each user session. For a list of supported TACACS+ AV pairs, refer to the appendix "TACACS+ Attribute-Value Pairs."

TACACS+ Configuration Examples

The following sections provide TACACS+ configuration examples:

- TACACS+ Authentication Examples
- TACACS+ Authorization Example
- TACACS+ Accounting Example
- TACACS+ Server Group Example
- AAA Server Group Selection Based on DNIS Example
- TACACS+ Daemon Configuration Example

TACACS+ Authentication Examples

The following example shows how to configure TACACS+ as the security protocol for PPP authentication:

```
aaa new-model
aaa authentication ppp test group tacacs+ local
tacacs-server host 10.1.2.3
tacacs-server key goaway
interface serial 0
 ppp authentication chap pap test
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, "test," to be used on serial interfaces running PPP. The keyword **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.
- The **tacacs-server host** command identifies the TACACS+ daemon as having an IP address of 10.1.2.3. The **tacacs-server key** command defines the shared encryption key to be "goaway."
- The **interface** command selects the line, and the **ppp authentication** command applies the test method list to this line.

The following example shows how to configure TACACS+ as the security protocol for PPP authentication, but instead of the "test" method list, the "default" method list is used.

```
aaa new-model
aaa authentication ppp default if-needed group tacacs+ local
tacacs-server host 10.1.2.3
tacacs-server key goaway
interface serial 0
 ppp authentication default
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, "default," to be used on serial interfaces running PPP. The keyword **default** means that PPP authentication is applied by default to all interfaces. The **if-needed** keyword means that if the user has already authenticated by going through the ASCII login procedure, then PPP authentication is not necessary and can be skipped. If authentication is needed, the keyword **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.
- The **tacacs-server host** command identifies the TACACS+ daemon as having an IP address of 10.1.2.3. The **tacacs-server key** command defines the shared encryption key to be "goaway."
- The **interface** command selects the line, and the **ppp authentication** command applies the default method list to this line.

The following example shows how to create the same authentication algorithm for PAP, but it calls the method list "MIS-access" instead of "default":

```
aaa new-model
aaa authentication pap MIS-access if-needed group tacacs+ local
tacacs-server host 10.1.2.3
tacacs-server key goaway
interface serial 0
 ppp authentication pap MIS-access
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, "MIS-access," to be used on serial interfaces running PPP. The method list, "MIS-access," means that PPP authentication is applied to all interfaces. The **if-needed** keyword means that if the user has already authenticated by going through the ASCII login procedure, then PPP authentication is not necessary and can be skipped. If authentication is needed, the keyword **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.
- The **tacacs-server host** command identifies the TACACS+ daemon as having an IP address of 10.1.2.3. The **tacacs-server key** command defines the shared encryption key to be "goaway."
- The **interface** command selects the line, and the **ppp authentication** command applies the default method list to this line.

The following example shows the configuration for a TACACS+ daemon with an IP address of 10.2.3.4 and an encryption key of “apple”:

```
aaa new-model
aaa authentication login default group tacacs+ local
tacacs-server host 10.2.3.4
tacacs-server key apple
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines the default method list. Incoming ASCII logins on all interfaces (by default) will use TACACS+ for authentication. If no TACACS+ server responds, then the network access server will use the information contained in the local username database for authentication.
- The **tacacs-server host** command identifies the TACACS+ daemon as having an IP address of 10.2.3.4. The **tacacs-server key** command defines the shared encryption key to be “apple.”

TACACS+ Authorization Example

The following example shows how to configure TACACS+ as the security protocol for PPP authentication using the default method list; it also shows how to configure network authorization via TACACS+:

```
aaa new-model
aaa authentication ppp default if-needed group tacacs+ local
aaa authorization network default group tacacs+
tacacs-server host 10.1.2.3
tacacs-server key goaway
interface serial 0
 ppp authentication default
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, “default,” to be used on serial interfaces running PPP. The keyword **default** means that PPP authentication is applied by default to all interfaces. The **if-needed** keyword means that if the user has already authenticated by going through the ASCII login procedure, then PPP authentication is not necessary and can be skipped. If authentication is needed, the keyword **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.
- The **aaa authorization** command configures network authorization via TACACS+. Unlike authentication lists, this authorization list always applies to all incoming network connections made to the network access server.
- The **tacacs-server host** command identifies the TACACS+ daemon as having an IP address of 10.1.2.3. The **tacacs-server key** command defines the shared encryption key to be “goaway.”
- The **interface** command selects the line, and the **ppp authentication** command applies the default method list to this line.

TACACS+ Accounting Example

The following example shows how to configure TACACS+ as the security protocol for PPP authentication using the default method list; it also shows how to configure accounting via TACACS+:

```
aaa new-model
aaa authentication ppp default if-needed group tacacs+ local
aaa accounting network default stop-only group tacacs+
tacacs-server host 10.1.2.3
tacacs-server key goaway
interface serial 0
 ppp authentication default
```

The lines in the preceding sample configuration are defined as follows:

- The **aaa new-model** command enables the AAA security services.
- The **aaa authentication** command defines a method list, "default," to be used on serial interfaces running PPP. The keyword **default** means that PPP authentication is applied by default to all interfaces. The **if-needed** keyword means that if the user has already authenticated by going through the ASCII login procedure, then PPP authentication is not necessary and can be skipped. If authentication is needed, the keyword **group tacacs+** means that authentication will be done through TACACS+. If TACACS+ returns an ERROR of some sort during authentication, the keyword **local** indicates that authentication will be attempted using the local database on the network access server.
- The **aaa accounting** command configures network accounting via TACACS+. In this example, accounting records describing the session that just terminated will be sent to the TACACS+ daemon whenever a network connection terminates.
- The **tacacs-server host** command identifies the TACACS+ daemon as having an IP address of 10.1.2.3. The **tacacs-server key** command defines the shared encryption key to be "goaway."
- The **interface** command selects the line, and the **ppp authentication** command applies the default method list to this line.

TACACS+ Server Group Example

The following example shows how to create a server group with three different TACACS+ servers members:

```
aaa group server tacacs tacgroup1
    server 172.16.1.1
    server 172.16.1.21
    server 172.16.1.31
```

AAA Server Group Selection Based on DNIS Example

The following example shows how to select TACAC+ server groups based on DNIS to provide specific AAA services:

```
! This command enables AAA.
aaa new-model
!
! The following set of commands configures the TACACS+ servers that will be associated
! with one of the defined server groups.
tacacs-server host 172.16.0.1
tacacs-server host 172.17.0.1
```

```
tacacs-server host 172.18.0.1
tacacs-server host 172.19.0.1
tacacs-server host 172.20.0.1
tacacs-server key abcdefg

! The following commands define the sg1 TACACS+ server group and associate servers
! with it.
aaa group server tacacs sg1
  server 172.16.0.1
  server 172.17.0.1
! The following commands define the sg2 TACACS+ server group and associate a server
! with it.
aaa group server tacacs sg2
  server 172.18.0.1
! The following commands define the sg3 TACACS+ server group and associate a server
! with it.
aaa group server tacacs sg3
  server 172.19.0.1
! The following commands define the default-group TACACS+ server group and associate
! a server with it.
aaa group server tacacs default-group
  server 172.20.0.1
!
! The next set of commands configures default-group tacacs server group parameters.
aaa authentication ppp default group default-group
aaa accounting network default start-stop group default-group
!
! The next set of commands enables DNIS mapping and maps DNIS numbers to the defined
! RADIUS server groups. In this configuration, all PPP connection requests using DNIS
! 7777 are sent to the sg1 server group. The accounting records for these connections
! (specifically, start-stop records) are handled by the sg2 server group. Calls with a
! DNIS of 8888 use server group sg3 for authentication and server group default-group
! for accounting. Calls with a DNIS of 9999 use server group default-group for
! authentication and server group sg3 for accounting records (stop records only). All
! other calls with DNIS other than the ones defined use the server group default-group
! for both authentication and stop-start accounting records.
aaa dnis map enable
aaa dnis map 7777 authentication ppp group sg1
aaa dnis map 7777 accounting network start-stop group sg2
aaa dnis map 8888 authentication ppp group sg3
aaa dnis map 9999 accounting network stop-only group sg3
```

TACACS+ Daemon Configuration Example

The following example shows a sample configuration of the TACACS+ daemon. The precise syntax used by your TACACS+ daemon may be different from what is included in this example.

```
user = mci_customer1 {
 chap = cleartext "some chap password"
 service = ppp protocol = ip {
inacl#1="permit ip any any precedence immediate"
inacl#2="deny igrp 0.0.1.2 255.255.0.0 any"
 }
}
```

Per VRF for TACACS+ Servers

The Per VRF for TACACS+ Servers feature allows you to configure per virtual route forwarding (per VRF) authentication, authorization, and accounting (AAA) on TACACS+ servers.

Feature History for Per VRF for TACACS+ Servers

Release	Modification
12.3(7)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for Per VRF for TACACS+ Servers, page 397
- Restrictions for Per VRF for TACACS+ Servers, page 398
- Information About Per VRF for TACACS+ Servers, page 398
- How to Configure Per VRF for TACACS+ Servers, page 398
- Configuration Examples for Per VRF for TACACS+ Servers, page 401
- Additional References, page 402
- Command Reference, page 403

Prerequisites for Per VRF for TACACS+ Servers

- You must have access to a TACACS+ server.
- You should be familiar with configuring TACACS+.
- You should be familiar with configuring AAA and per VRF AAA.
- You should be familiar with configuring group servers.

Restrictions for Per VRF for TACACS+ Servers

- You must define the VRF instance before you can configure per VRF for a TACACS+ server.

Information About Per VRF for TACACS+ Servers

To configure the Per VRF for TACACS+ Servers feature, you should understand the following concept:

- Per VRF for TACACS+ Servers Overview, page 398

Per VRF for TACACS+ Servers Overview

The Per VRF for TACACS+ Servers feature allows you to configure per VRF AAA on TACACS+ servers. Prior to Cisco IOS Release 12.3(7)T, this functionality was available only on RADIUS servers.

How to Configure Per VRF for TACACS+ Servers

This section contains the following procedures:

- Configuring Per VRF on a TACACS+ Server, page 398 (required)
- Verifying Per VRF for TACACS+ Servers, page 400 (optional)

Configuring Per VRF on a TACACS+ Server

Before configuring per VRF on a TACACS+ server, you must have configured AAA and a server group. Then you are ready to create the VRF routing table, as shown in Steps 3 and 4 of the DETAILED STEPS table below. At that point, you need to configure the interface, which is shown in Steps 6, 7, and 8 of the table. The actual configuration of per VRF on a TACACS+ server is configured in Steps 10 through 13 of the table.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **exit**
6. **interface** *interface-name*
7. **ip vrf forwarding** *vrf-name*
8. **ip address** *ip-address mask* [**secondary**]
9. **exit**
10. **aaa group server tacacs+** *group-name*

11. **server-private** {*ip-address* | *name*} [**nat**] [**single-connection**] [**port** *port-number*] [**timeout** *seconds*] [**key** [**0** | **7**] *string*]
12. **ip vrf forwarding** *vrf-name*
13. **ip tacacs source-interface** *subinterface-name*
14. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Router (config)# ip vrf cisco`	Configures a VRF table and enters VRF configuration mode.
Step 4	**rd** *route-distinguisher* **Example:** `Router (config-vrf)# rd 100:1`	Creates routing and forwarding tables for a VRF instance.
Step 5	**exit** **Example:** `Router (config-vrf)# exit`	Exits VRF configuration mode.
Step 6	**interface** *interface-name* **Example:** `Router (config)# interface Loopback0`	Configures an interface and enters interface configuration mode.
Step 7	**ip vrf forwarding** *vrf-name* **Example:** `Router (config-if)# ip vrf forwarding cisco`	Configures a VRF for the interface.
Step 8	**ip address** *ip-address mask* [**secondary**] **Example:** `Router (config-if)# ip address 10.0.0.2 255.0.0.0`	Sets a primary or secondary IP address for an interface.
Step 9	**exit** **Example:** `Router (config-if)# exit`	Exits interface configuration mode.

	Command or Action	Purpose
Step 10	**aaa group server tacacs+** *group-name* **Example:** `Router (config)# aaa group server tacacs+ tacacs1`	Groups different TACACS+ server hosts into distinct lists and distinct methods and enters server-group configuration mode.
Step 11	**server-private** {*ip-address* | *name*} [**nat**] [**single-connection**] [**port** *port-number*] [**timeout** *seconds*] [**key** [**0** | **7**] *string*] **Example:** `Router (config-sg-tacacs+)# server-private 10.1.1.1 port 19 key cisco`	Configures the IP address of the private TACACS+ server for the group server.
Step 12	**ip vrf forwarding** *vrf-name* **Example:** `Router (config-sg-tacacs+)# ip vrf forwarding cisco`	Configures the VRF reference of a AAA TACACS+ server group.
Step 13	**ip tacacs source-interface** *subinterface-name* **Example:** `Router (config-sg-tacacs+)# ip tacacs source-interface Loopback0`	Uses the IP address of a specified interface for all outgoing TACACS+ packets.
Step 14	**exit** **Example:** `Router (config-sg-tacacs)# exit`	Exits server-group configuration mode.

Verifying Per VRF for TACACS+ Servers

To verify your per VRF TACACS+ configuration, you can perform the following steps. The **debug** commands may be used in any order.

SUMMARY STEPS

1. **enable**
2. **debug tacacs authentication**
3. **debug tacacs authorization**
4. **debug tacacs accounting**
5. **debug tacacs packets**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug tacacs authentication** **Example:** `Router# debug tacacs authentication`	Displays information about AAA/TACACS+ authentication.
Step 3	**debug tacacs authorization** **Example:** `Router# debug tacacs authorization`	Displays information about AAA/TACACS+ authorization.
Step 4	**debug tacacs accounting** **Example:** `Router# debug tacacs accounting`	Displays information about accountable events as they occur.
Step 5	**debug tacacs packets** **Example:** `Router# debug tacacs packets`	Displays information about TACACS+ packets.

Configuration Examples for Per VRF for TACACS+ Servers

This section includes the following configuration example:

• Configuring Per VRF for TACACS+ Servers: Example, page 401

Configuring Per VRF for TACACS+ Servers: Example

The following output example shows that the group server tacacs1 has been configured for per VRF AAA services:

```
aaa group server tacacs+ tacacs1
    server-private 10.1.1.1 port 19 key cisco
    ip vrf forwarding cisco
    ip tacacs source-interface Loopback0

  ip vrf cisco
   rd 100:1

  interface Loopback0
   ip address 4.0.0.2 255.0.0.0
   ip vrf forwarding cisco
```

Additional References

The following sections provide references related to Per VRF for TACACS+ Servers.

Related Documents

Related Topic	Document Title
Configuring TACACS+	"Configuring TACACS+" chapter of the "Security Server Protocols" section of the *Cisco IOS Security Configuration Guide*
Per VRF AAA	*Per VRF AAA*
Cisco IOS commands	Cisco Master Commands list, Release 12.3 T
Security commands	*Cisco IOS Security Commands*, Release 12.3 T

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **ip tacacs source-interface**
- **ip vrf forwarding (server-group)**
- **server-private (TACACS+)**

Configuring Kerberos

This chapter describes the Kerberos security system. For a complete description of the Kerberos commands used in this chapter, refer to the "Kerberos Commands" chapter in the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature, or refer to the software release notes for a specific release. For more information, see the "Finding Additional Feature Support Information" section on page cxvii in the chapter Using Cisco IOS Software for Release 12.4.

In This Chapter

This chapter includes the following topics and tasks:

- About Kerberos
- Kerberos Client Support Operation
- Kerberos Configuration Task List
- Kerberos Configuration Examples

About Kerberos

Kerberos is a secret-key network authentication protocol, developed at the Massachusetts Institute of Technology (MIT), that uses the Data Encryption Standard (DES) cryptographic algorithm for encryption and authentication. Kerberos was designed to authenticate requests for network resources. Kerberos, like other secret-key systems, is based on the concept of a trusted third party that performs secure verification of users and services. In the Kerberos protocol, this trusted third party is called the key distribution center (KDC).

The primary use of Kerberos is to verify that users and the network services they use are really who and what they claim to be. To accomplish this, a trusted Kerberos server issues tickets to users. These tickets, which have a limited lifespan, are stored in a user's credential cache and can be used in place of the standard username-and-password authentication mechanism.

The Kerberos credential scheme embodies a concept called "single logon." This process requires authenticating a user once, and then allows secure authentication (without encrypting another password) wherever that user's credential is accepted.

Starting with Cisco IOS Release 11.2, Cisco IOS software includes Kerberos 5 support, which allows organizations already deploying Kerberos 5 to use the same Kerberos authentication database on their routers that they are already using on their other network hosts (such as UNIX servers and PCs).

The following network services are supported by the Kerberos authentication capabilities in Cisco IOS software:

- Telnet
- rlogin
- rsh
- rcp

Note Cisco Systems' implementation of Kerberos client support is based on code developed by CyberSafe, which was derived from the MIT code. As a result, the Cisco Kerberos implementation has successfully undergone full compatibility testing with the CyberSafe Challenger commercial Kerberos server and MIT's server code, which is freely distributed.

Table 20 lists common Kerberos-related terms and their definitions.

Table 20 Kerberos Terminology

Term	Definition
authentication	A process by which a user or service identifies itself to another service. For example, a client can authenticate to a router or a router can authenticate to another router.
authorization	A means by which the router determines what privileges you have in a network or on the router and what actions you can perform.
credential	A general term that refers to authentication tickets, such as ticket granting tickets (TGTs) and service credentials. Kerberos credentials verify the identity of a user or service. If a network service decides to trust the Kerberos server that issued a ticket, it can be used in place of retyping in a username and password. Credentials have a default lifespan of eight hours.
instance	An authorization level label for Kerberos principals. Most Kerberos principals are of the form user@REALM (for example, smith@EXAMPLE.COM). A Kerberos principal with a Kerberos instance has the form user/instance@REALM (for example, smith/admin@EXAMPLE.COM). The Kerberos instance can be used to specify the authorization level for the user if authentication is successful. It is up to the server of each network service to implement and enforce the authorization mappings of Kerberos instances. Note that the Kerberos realm name must be in uppercase characters.
Kerberized	Applications and services that have been modified to support the Kerberos credential infrastructure.
Kerberos realm	A domain consisting of users, hosts, and network services that are registered to a Kerberos server. The Kerberos server is trusted to verify the identity of a user or network service to another user or network service. Kerberos realms must always be in uppercase characters.
Kerberos server	A daemon running on a network host. Users and network services register their identity with the Kerberos server. Network services query the Kerberos server to authenticate to other network services.

Table 20 Kerberos Terminology (continued)

Term	Definition
key distribution center (KDC)	A Kerberos server and database program running on a network host.
principal	Also known as a Kerberos identity, this is who you are or what a service is according to the Kerberos server.
service credential	A credential for a network service. When issued from the KDC, this credential is encrypted with the password shared by the network service and the KDC, and with the user's TGT.
SRVTAB	A password that a network service shares with the KDC. The network service authenticates an encrypted service credential by using the SRVTAB (also known as a KEYTAB) to decrypt it.
ticket granting ticket (TGT)	A credential that the key distribution center (KDC) issues to authenticated users. When users receive a TGT, they can authenticate to network services within the Kerberos realm represented by the KDC.

Kerberos Client Support Operation

This section describes how the Kerberos security system works with a Cisco router functioning as the security server. Although (for convenience or technical reasons) you can customize Kerberos in a number of ways, remote users attempting to access network services must pass through three layers of security before they can access network services.

This section includes the following sections:

- Authenticating to the Boundary Router
- Obtaining a TGT from a KDC
- Authenticating to Network Services

Authenticating to the Boundary Router

This section describes the first layer of security that remote users must pass through when they attempt to access a network. The first step in the Kerberos authentication process is for users to authenticate themselves to the boundary router. The following process describes how users authenticate to a boundary router:

1. The remote user opens a PPP connection to the corporate site router.
2. The router prompts the user for a username and password.
3. The router requests a TGT from the KDC for this particular user.
4. The KDC sends an encrypted TGT to the router that includes (among other things) the user's identity.
5. The router attempts to decrypt the TGT using the password the user entered. If the decryption is successful, the remote user is authenticated to the router.

A remote user who successfully initiates a PPP session and authenticates to the boundary router is inside the firewall but still must authenticate to the KDC directly before being allowed to access network services. This is because the TGT issued by the KDC is stored on the router and is not useful for additional authentication unless the user physically logs on to the router.

Obtaining a TGT from a KDC

This section describes how remote users who are authenticated to the boundary router authenticate themselves to a KDC.

When a remote user authenticates to a boundary router, that user technically becomes part of the network; that is, the network is extended to include the remote user and the user's machine or network. To gain access to network services, however, the remote user must obtain a TGT from the KDC. The following process describes how remote users authenticate to the KDC:

1. The remote user, at a workstation on a remote site, launches the KINIT program (part of the client software provided with the Kerberos protocol).
2. The KINIT program finds the user's identity and requests a TGT from the KDC.
3. The KDC creates a TGT, which contains the identity of the user, the identity of the KDC, and the expiration time of the TGT.
4. Using the user's password as a key, the KDC encrypts the TGT and sends the TGT to the workstation.
5. When the KINIT program receives the encrypted TGT, it prompts the user for a password (this is the password that is defined for the user in the KDC).
6. If the KINIT program can decrypt the TGT with the password the user enters, the user is authenticated to the KDC, and the KINIT program stores the TGT in the user's credential cache.

At this point, the user has a TGT and can communicate securely with the KDC. In turn, the TGT allows the user to authenticate to other network services.

Authenticating to Network Services

The following process describes how a remote user with a TGT authenticates to network services within a given Kerberos realm. Assume the user is on a remote workstation (Host A) and wants to log in to Host B.

1. The user on Host A initiates a Kerberized application (such as Telnet) to Host B.
2. The Kerberized application builds a service credential request and sends it to the KDC. The service credential request includes (among other things) the user's identity and the identity of the desired network service. The TGT is used to encrypt the service credential request.
3. The KDC tries to decrypt the service credential request with the TGT it issued to the user on Host A. If the KDC can decrypt the packet, it is assured that the authenticated user on Host A sent the request.
4. The KDC notes the network service identity in the service credential request.
5. The KDC builds a service credential for the appropriate network service on Host B on behalf of the user on Host A. The service credential contains the client's identity and the desired network service's identity.

6. The KDC then encrypts the service credential twice. It first encrypts the credential with the SRVTAB that it shares with the network service identified in the credential. It then encrypts the resulting packet with the TGT of the user (who, in this case, is on Host A).
7. The KDC sends the twice-encrypted credential to Host A.
8. Host A attempts to decrypt the service credential with the user's TGT. If Host A can decrypt the service credential, it is assured the credential came from the real KDC.
9. Host A sends the service credential to the desired network service. Note that the credential is still encrypted with the SRVTAB shared by the KDC and the network service.
10. The network service attempts to decrypt the service credential using its SRVTAB.
11. If the network service can decrypt the credential, it is assured the credential was in fact issued from the KDC. Note that the network service trusts anything it can decrypt from the KDC, even if it receives it indirectly from a user. This is because the user first authenticated with the KDC.

At this point, the user is authenticated to the network service on Host B. This process is repeated each time a user wants to access a network service in the Kerberos realm.

Kerberos Configuration Task List

For hosts and the KDC in your Kerberos realm to communicate and mutually authenticate, you must identify them to each other. To do this, you add entries for the hosts to the Kerberos database on the KDC and add SRVTAB files generated by the KDC to all hosts in the Kerberos realm. You also make entries for users in the KDC database.

This section describes how to set up a Kerberos-authenticated server-client system and contains the following topics:

- Configuring the KDC Using Kerberos Commands
- Configuring the Router to Use the Kerberos Protocol

This section assumes that you have installed the Kerberos administrative programs on a UNIX host, known as the KDC, initialized the database, and selected a Kerberos realm name and password. For instructions about completing these tasks, refer to documentation that came with your Kerberos software.

Note Write down the host name or IP address of the KDC, the port number you want the KDC to monitor for queries, and the name of the Kerberos realm it will serve. You need this information to configure the router.

Configuring the KDC Using Kerberos Commands

After you set up a host to function as the KDC in your Kerberos realm, you must make entries to the KDC database for all principals in the realm. Principals can be network services on Cisco routers and hosts or they can be users.

To use Kerberos commands to add services to the KDC database (and to modify existing database information), complete the tasks in the following sections:

- Adding Users to the KDC Database
- Creating SRVTABs on the KDC
- Extracting SRVTABs

Note All Kerberos command examples are based on Kerberos 5 Beta 5 of the original MIT implementation. Later versions use a slightly different interface.

Adding Users to the KDC Database

To add users to the KDC and create privileged instances of those users, use the **su** command to become root on the host running the KDC and use the kdb5_edit program to use the following commands in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# ank username@REALM`	Use the **ank** (add new key) command to add a user to the KDC. This command prompts for a password, which the user must enter to authenticate to the router.
Step 2	`Router# ank username/instance@REALM`	Use the **ank** command to add a privileged instance of a user.

For example, to add user *loki* of Kerberos realm CISCO.COM, enter the following Kerberos command:

```
ank loki@CISCO.COM
```

Note The Kerberos realm name must be in uppercase characters.

You might want to create privileged instances to allow network administrators to connect to the router at the enable level, for example, so that they need not enter a clear text password (and compromise security) to enter enable mode.

To add an instance of *loki* with additional privileges (in this case, *enable*, although it could be anything) enter the following Kerberos command:

```
ank loki/enable@CISCO.COM
```

In each of these examples, you are prompted to enter a password, which you must give to user *loki* to use at login.

The "Enabling Kerberos Instance Mapping" section describes how to map Kerberos instances to various Cisco IOS privilege levels.

Creating SRVTABs on the KDC

All routers that you want to authenticate to use the Kerberos protocol must have an SRVTAB. This section and the "Extracting SRVTABs" section describe how to create and extract SRVTABs for a router called *router1*. The section "Copying SRVTAB Files" describes how to copy SRVTAB files to the router.

To make SRVTAB entries on the KDC, use the following command in privileged EXEC mode:

Command	Purpose
`Router# ark SERVICE/HOSTNAME@REALM`	Use the **ark** (add random key) command to add a network service supported by a host or router to the KDC.

For example, to add a Kerberized authentication service for a Cisco router called *router1* to the Kerberos realm CISCO.COM, enter the following Kerberos command:

```
ark host/router1.cisco.com@CISCO.COM
```

Make entries for all network services on all Kerberized hosts that use this KDC for authentication.

Extracting SRVTABs

SRVTABs contain (among other things) the passwords or randomly generated keys for the service principals you entered into the KDC database. Service principal keys must be shared with the host running that service. To do this, you must save the SRVTAB entries to a file, then copy the file to the router and all hosts in the Kerberos realm. Saving SRVTAB entries to a file is called *extracting* SRVTABs. To extract SRVTABs, use the following command in privileged EXEC mode:

Command	Purpose
`Router# xst router-name host`	Use the kdb5_edit command **xst** to write an SRVTAB entry to a file.

For example, to write the host/router1.cisco.com@CISCO.COM SRVTAB to a file, enter the following Kerberos command:

```
xst router1.cisco.com@CISCO.COM host
```

Use the **quit** command to exit the kdb5_edit program.

Configuring the Router to Use the Kerberos Protocol

To configure a Cisco router to function as a network security server and authenticate users using the Kerberos protocol, complete the tasks in the following sections:

- Defining a Kerberos Realm
- Copying SRVTAB Files
- Specifying Kerberos Authentication
- Enabling Credentials Forwarding
- Opening a Telnet Session to the Router
- Establishing an Encrypted Kerberized Telnet Session
- Enabling Mandatory Kerberos Authentication
- Enabling Kerberos Instance Mapping
- Monitoring and Maintaining Kerberos

Defining a Kerberos Realm

For a router to authenticate a user defined in the Kerberos database, it must know the host name or IP address of the host running the KDC, the name of the Kerberos realm and, optionally, be able to map the host name or Domain Name System (DNS) domain to the Kerberos realm.

To configure the router to authenticate to a specified KDC in a specified Kerberos realm, use the following commands in global configuration mode. Note that DNS domain names must begin with a leading dot (.):

	Command	Purpose
Step 1	`Router(config)# kerberos local-realm kerberos-realm`	Defines the default realm for the router.
Step 2	`Router(config)# kerberos server kerberos-realm {hostname | ip-address} [port-number]`	Specifies to the router which KDC to use in a given Kerberos realm and, optionally, the port number that the KDC is monitoring. (The default is 88.)
Step 3	`Router(config)# kerberos realm {dns-domain | host} kerberos-realm`	(Optional) Maps a host name or DNS domain to a Kerberos realm.

Note Because the machine running the KDC and all Kerberized hosts must interact within a 5-minute window or authentication fails, all Kerberized machines, and especially the KDC, should be running the Network Time Protocol (NTP).

The **kerberos local-realm**, **kerberos realm**, and **kerberos server** commands are equivalent to the UNIX *krb.conf* file. Table 21 identifies mappings from the Cisco IOS configuration commands to a Kerberos 5 configuration file (krb5.conf).

Table 21 Kerberos 5 Configuration File and Commands

krb5.conf File	Cisco IOS Configuration Command
[libdefaults] `default_realm = DOMAIN.COM`	(in configuration mode) **kerberos local-realm** *DOMAIN.COM*
[domain_realm] `.domain.com = DOMAIN.COM` `domain.com = DOMAIN.COM`	(in configuration mode) **kerberos realm**.*domain.com DOMAIN.COM* **kerberos realm** *domain.com DOMAIN.COM*
[realms] `kdc = DOMAIN.PIL.COM:750` `admin_server = DOMAIN.PIL.COM` `default_domain = DOMAIN.COM`	(in configuration mode) **kerberos server** *DOMAIN.COM 172.65.44.2* (*172.65.44.2* is the example IP address for *DOMAIN.PIL.COM*)

For an example of defining a Kerberos realm, see the section "Defining a Kerberos Realm" later in this chapter.

Copying SRVTAB Files

To make it possible for remote users to authenticate to the router using Kerberos credentials, the router must share a secret key with the KDC. To do this, you must give the router a copy of the SRVTAB you extracted on the KDC.

The most secure method to copy SRVTAB files to the hosts in your Kerberos realm is to copy them onto physical media and go to each host in turn and manually copy the files onto the system. To copy SRVTAB files to the router, which does not have a physical media drive, you must transfer them via the network using TFTP.

To remotely copy SRVTAB files to the router from the KDC, use the following command in global configuration mode:

Command	Purpose
`Router(config)# kerberos srvtab remote {hostname | ip-address} {filename}`	Retrieves an SRVTAB file from the KDC.

When you copy the SRVTAB file from the router to the KDC, the **kerberos srvtab remote** command parses the information in this file and stores it in the router's running configuration in the **kerberos srvtab entry** format. To ensure that the SRVTAB is available (does not need to be acquired from the KDC) when you reboot the router, use the **write memory** configuration command to write your running configuration (which contains the parsed SRVTAB file) to NVRAM.

For an example of copying SRVTAB files, see the section "SRVTAB File Copying Example" later in this chapter.

Specifying Kerberos Authentication

You have now configured Kerberos on your router. This makes it possible for the router to authenticate using Kerberos. The next step is to tell it to do so. Because Kerberos authentication is facilitated through AAA, you need to enter the **aaa authentication** command, specifying Kerberos as the authentication method. For more information, refer to the chapter "Configuring Authentication".

Enabling Credentials Forwarding

With Kerberos configured thus far, a user authenticated to a Kerberized router has a TGT and can use it to authenticate to a host on the network. However, if the user tries to list credentials after authenticating to a host, the output will show no Kerberos credentials present.

You can optionally configure the router to forward users' TGTs with them as they authenticate from the router to Kerberized remote hosts on the network when using Kerberized Telnet, rcp, rsh, and rlogin (with the appropriate flags).

To force all clients to forward users' credentials as they connect to other hosts in the Kerberos realm, use the following command in global configuration mode:

Command	Purpose
`Router(config)# kerberos credentials forward`	Forces all clients to forward user credentials upon successful Kerberos authentication.

With credentials forwarding enabled, users' TGTs are automatically forwarded to the next host they authenticate to. In this way, users can connect to multiple hosts in the Kerberos realm without running the KINIT program each time to get a new TGT.

Opening a Telnet Session to the Router

To use Kerberos to authenticate users opening a Telnet session to the router from within the network, use the following command in global configuration mode:

Command	Purpose
`Router(config)# aaa authentication login {default | list-name} krb5_telnet`	Sets login authentication to use the Kerberos 5 Telnet authentication protocol when using Telnet to connect to the router.

Although Telnet sessions to the router are authenticated, users must still enter a clear text password if they want to enter enable mode. The **kerberos instance map** command, discussed in a later section, allows them to authenticate to the router at a predefined privilege level.

Establishing an Encrypted Kerberized Telnet Session

Another way for users to open a secure Telnet session is to use Encrypted Kerberized Telnet. With Encrypted Kerberized Telnet, users are authenticated by their Kerberos credentials before a Telnet session is established. The Telnet session is encrypted using 56-bit Data Encryption Standard (DES) encryption with 64-bit Cipher Feedback (CFB). Because data sent or received is encrypted, not clear text, the integrity of the dialed router or access server can be more easily controlled.

Note This feature is available only if you have the 56-bit encryption image. 56-bit DES encryption is subject to U.S. Government export control regulations.

To establish an encrypted Kerberized Telnet session from a router to a remote host, use either of the following commands in EXEC command mode:

Command	Purpose
`Router(config)# connect host [port] /encrypt kerberos` or `Router(config)# telnet host [port] /encrypt kerberos`	Establishes an encrypted Telnet session.

When a user opens a Telnet session from a Cisco router to a remote host, the router and remote host negotiate to authenticate the user using Kerberos credentials. If this authentication is successful, the router and remote host then negotiate whether or not to use encryption. If this negotiation is successful, both inbound and outbound traffic is encrypted using 56-bit DES encryption with 64-bit CFB.

When a user dials in from a remote host to a Cisco router configured for Kerberos authentication, the host and router will attempt to negotiate whether or not to use encryption for the Telnet session. If this negotiation is successful, the router will encrypt all outbound data during the Telnet session.

If encryption is not successfully negotiated, the session will be terminated and the user will receive a message stating that the encrypted Telnet session was not successfully established.

For information about enabling bidirectional encryption from a remote host, refer to the documentation specific to the remote host device.

For an example of using encrypted Kerberized Telnet to open a secure Telnet session, see the section "Encrypted Telnet Session Example" later in this chapter.

Enabling Mandatory Kerberos Authentication

As an added layer of security, you can optionally configure the router so that, after remote users authenticate to it, these users can authenticate to other services on the network only with Kerberized Telnet, rlogin, rsh, and rcp. If you do not make Kerberos authentication mandatory and Kerberos authentication fails, the application attempts to authenticate users using the default method of authentication for that network service; for example, Telnet and rlogin prompt for a password, and rsh attempts to authenticate using the local rhost file.

To make Kerberos authentication mandatory, use the following command in global configuration mode:

Command	Purpose
`Router(config)# kerberos clients mandatory`	Sets Telnet, rlogin, rsh, and rcp to fail if they cannot negotiate the Kerberos protocol with the remote server.

Enabling Kerberos Instance Mapping

As mentioned in the section "Creating SRVTABs on the KDC," you can create administrative instances of users in the KDC database. The **kerberos instance map** command allows you to map those instances to Cisco IOS privilege levels so that users can open secure Telnet sessions to the router at a predefined privilege level, obviating the need to enter a clear text password to enter enable mode.

To map a Kerberos instance to a Cisco IOS privilege level, use the following command in global configuration mode:

Command	Purpose
`Router(config)# kerberos instance map` *`instance privilege-level`*	Maps a Kerberos instance to a Cisco IOS privilege level.

If there is a Kerberos instance for user *loki* in the KDC database (for example, *loki/admin*), user *loki* can now open a Telnet session to the router as loki/admin and authenticate automatically at privilege level 15, assuming instance "admin" is mapped to privilege level 15. (See the section "Adding Users to the KDC Database" earlier in this chapter.)

Cisco IOS commands can be set to various privilege levels using the **privilege level** command.

After you map a Kerberos instance to a Cisco IOS privilege level, you must configure the router to check for Kerberos instances each time a user logs in. To run authorization to determine if a user is allowed to run an EXEC shell based on a mapped Kerberos instance, use the **aaa authorization** command with the **krb5-instance** keyword. For more information, refer to the chapter "Configuring Authorization."

Monitoring and Maintaining Kerberos

To display or remove a current user's credentials, use the following commands in EXEC mode:

	Command	Purpose
Step 1	`Router# show kerberos creds`	Lists the credentials in a current user's credentials cache.
Step 2	`Router# clear kerberos creds`	Destroys all credentials in a current user's credentials cache, including those forwarded.

For an example of Kerberos configuration, see the section "Kerberos Configuration Examples".

Kerberos Configuration Examples

The following sections provide Kerberos configuration examples:

- Kerberos Realm Definition Examples
- SRVTAB File Copying Example
- Kerberos Configuration Examples
- Encrypted Telnet Session Example

Kerberos Realm Definition Examples

To define CISCO.COM as the default Kerberos realm, use the following command:

```
kerberos local-realm CISCO.COM
```

To tell the router that the CISCO.COM KDC is running on host 10.2.3.4 at port number 170, use the following Kerberos command:

```
kerberos server CISCO.COM 10.2.3.4 170
```

To map the DNS domain cisco.com to the Kerberos realm CISCO.COM, use the following command:

```
kerberos realm.cisco.com CISCO.COM
```

SRVTAB File Copying Example

To copy over the SRVTAB file on a host named host123.cisco.com for a router named router1.cisco.com, the command would look like this:

```
kerberos srvtab remote host123.cisco.com router1.cisco.com-new-srvtab
```

Kerberos Configuration Examples

This section provides a typical non-Kerberos router configuration and shows output for this configuration from the **write term** command, then builds on this configuration by adding optional Kerberos functionality. Output for each configuration is presented for comparison against the previous configuration.

This example shows how to use the kdb5_edit program to perform the following configuration tasks:

- Adding user chet to the Kerberos database
- Adding a privileged Kerberos instance of user chet (chet/admin) to the Kerberos database
- Adding a restricted instance of chet (chet/restricted) to the Kerberos database
- Adding workstation chet-ss20.cisco.com
- Adding router chet-2500.cisco.com to the Kerberos database
- Adding workstation chet-ss20.cisco.com to the Kerberos database
- Extracting SRVTABs for the router and workstations
- Listing the contents of the KDC database (with the **ldb** command)

Note that, in this sample configuration, host chet-ss20 is also the KDC:

```
chet-ss20# sbin/kdb5_edit
kdb5_edit:  ank chet
Enter password:
Re-enter password for verification:
kdb5_edit:  ank chet/admin
Enter password:
Re-enter password for verification:
kdb5_edit:  ank chet/restricted
Enter password:
Re-enter password for verification:
kdb5_edit:  ark host/chet-ss20.cisco.com
kdb5_edit:  ark host/chet-2500.cisco.com
kdb5_edit:  xst chet-ss20.cisco.com host
'host/chet-ss20.cisco.com@CISCO.COM' added to keytab
'WRFILE:chet-ss20.cisco.com-new-srvtab'
kdb5_edit:  xst chet-2500.cisco.com host
'host/chet-2500.cisco.com@CISCO.COM' added to keytab
'WRFILE:chet-2500.cisco.com-new-srvtab'
kdb5_edit:  ldb
entry: host/chet-2500.cisco.com@CISCO.COM
entry: chet/restricted@CISCO.COM
entry: chet@CISCO.COM
entry: K/M@CISCO.COM
entry: host/chet-ss20.cisco.com@CISCO.COM
entry: krbtgt/CISCO.COM@CISCO.COM
entry: chet/admin@CISCO.COM
kdb5_edit:  q
chet-ss20#
```

The following example shows output from a **write term** command, which displays the configuration of router chet-2500. This is a typical configuration with no Kerberos authentication.

```
chet-2500# write term
Building configuration...

Current configuration:
!
! Last configuration
change at 14:03:55 PDT Mon May 13 1996
!
version 11.2
service udp-small-servers
service tcp-small-servers
!
hostname chet-2500
!
clock timezone PST -8
```

```
clock summer-time PDT recurring
aaa new-model
aaa authentication login console none
aaa authentication ppp local local
enable password sMudgKin
!
username chet-2500 password 7 sMudgkin
username chet-3000 password 7 sMudgkin
username chetin password 7 sMudgkin
!
interface Ethernet0
 ip address 172.16.0.0 255.255.255.0
!
interface Serial0
 no ip address
 shutdown
 no fair-queue
!
interface Serial1
 no ip address
 shutdown
 no fair-queue
!
interface Async2
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic routing
 async mode dedicated
 no cdp enable
 ppp authentication pap local
 no tarp propagate
!
interface Async3
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic address
 async dynamic routing
 async mode dedicated
 no cdp enable
 ppp authentication pap local
 no tarp propagate
!
router eigrp 109
 network 172.17.0.0
 no auto-summary
!
ip default-gateway 172.30.55.64
ip domain-name cisco.com
ip name-server 192.168.0.0
ip classless
!
!

line con 0
 exec-timeout 0 0
 login authentication console
line 1 16
 transport input all
line aux 0
 transport input all
line vty 0 4
 password sMudgKin
```

```
!
ntp clock-period 17179703
ntp peer 172.19.10.0
ntp peer 172.19.0.0
end
```

The following example shows how to enable user authentication on the router via the Kerberos database. To enable user authentication via the Kerberos database, you would perform the following tasks:

- Entering configuration mode
- Defining the Kerberos local realm
- Identifying the machine hosting the KDC
- Enabling credentials forwarding
- Specifying Kerberos as the method of authentication for login
- Exiting configuration mode (CTL-Z)
- Writing the new configuration to the terminal

```
chet-2500# configure term
Enter configuration commands, one per line. End with CNTL/Z.
chet-2500(config)# kerberos local-realm CISCO.COM
chet-2500(config)# kerberos server CISCO.COM chet-ss20
Translating "chet-ss20"...domain server (192.168.0.0) [OK]

chet-2500(config)# kerberos credentials forward
chet-2500(config)# aaa authentication login default krb5
chet-2500(config)#
chet-2500#
%SYS-5-CONFIG_I: Configured from console by console
chet-2500# write term
```

Compare the following configuration with the previous one. In particular, look at the lines beginning with the words “aaa,” “username,” and “kerberos” (lines 10 through 20) in this new configuration.

```
Building configuration...

Current configuration:
!
! Last configuration change at 14:05:54 PDT Mon May 13 1996
!
version 11.2
service udp-small-servers
service tcp-small-servers
!
hostname chet-2500
!
clock timezone PST -8
clock summer-time PDT recurring
aaa new-model
aaa authentication login default krb5
aaa authentication login console none
aaa authentication ppp local local
enable password sMudgKin
!
username chet-2500 password 7 sMudgkin
username chet-3000 password 7 sMudgkin
username chetin password 7 sMudgkin
kerberos local-realm CISCO.COM
kerberos server CISCO.COM 172.71.54.14
kerberos credentials forward
!
```

```
interface Ethernet0
 ip address 172.16.0.0 255.255.255.0
!
interface Serial0
 no ip address
 shutdown
 no fair-queue
!
interface Serial1
 no ip address
 shutdown
 no fair-queue
!
interface Async2
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic routing
 async mode dedicated
 no cdp enable
 ppp authentication pap local
 no tarp propagate
!
interface Async3
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic address
 async dynamic routing
 async mode dedicated
 no cdp enable
 ppp authentication pap local
 no tarp propagate
!
router eigrp 109
 network 172.17.0.0
 no auto-summary
!
ip default-gateway 172.30.55.64
ip domain-name cisco.com
ip name-server 192.168.0.0
ip classless
!
!
line con 0
 exec-timeout 0 0
 login authentication console
line 1 16
 transport input all
line aux 0
 transport input all
line vty 0 4
 password sMudgKin
!
ntp clock-period 17179703
ntp peer 172.19.10.0
ntp peer 172.19.0.0
end
```

With the router configured thus far, user chet can log in to the router with a username and password and automatically obtain a TGT, as illustrated in the next example. With possession of a credential, user chet successfully authenticates to host chet-ss20 without entering a username/password.

```
chet-ss20% telnet chet-2500
Trying 172.16.0.0 ...
Connected to chet-2500.cisco.com.
Escape character is '^]'.


User Access Verification

Username: chet
Password:

chet-2500> show kerberos creds
Default Principal:  chet@CISCO.COM
Valid Starting          Expires                 Service Principal
13-May-1996 14:05:39    13-May-1996 22:06:40    krbtgt/CISCO.COM@CISCO.COM

chet-2500> telnet chet-ss20
Trying chet-ss20.cisco.com (172.71.54.14)... Open
Kerberos:       Successfully forwarded credentials


SunOS UNIX (chet-ss20) (pts/7)

Last login: Mon May 13 13:47:35 from chet-ss20.cisco.c
Sun Microsystems Inc.   SunOS 5.4       Generic July 1994
unknown mode: new
chet-ss20%
```

The following example shows how to authenticate to the router using Kerberos credentials. To authenticate using Kerberos credentials, you would perform the following tasks:

- Entering configuration mode
- Remotely copying over the SRVTAB file from the KDC
- Setting authentication at login to use the Kerberos 5 Telnet authentication protocol when using Telnet to connect to the router
- Writing the configuration to the terminal

Note that the new configuration contains a **kerberos srvtab entry** line. This line is created by the **kerberos srvtab remote** command.

```
chet-2500# configure term
Enter configuration commands, one per line.  End with CNTL/Z.
chet-2500(config)# kerberos srvtab remote earth chet/chet-2500.cisco.com-new-srvtab
Translating "earth"...domain server (192.168.0.0) [OK]

Loading chet/chet-2500.cisco.com-new-srvtab from 172.68.1.123 (via Ethernet0): !
[OK - 66/1000 bytes]

chet-2500(config)# aaa authentication login default krb5-telnet krb5
chet-2500(config)#
chet-2500#
%SYS-5-CONFIG_I: Configured from console by console
chet-2500# write term
Building configuration...

Current configuration:
!
```

```
! Last configuration change at 14:08:32 PDT Mon May 13 1996
!
version 11.2
service udp-small-servers
service tcp-small-servers
!
hostname chet-2500
!
clock timezone PST -8
clock summer-time PDT recurring
aaa new-model
aaa authentication login default krb5-telnet krb5
aaa authentication login console none
aaa authentication ppp local local
enable password sMudgKin
!
username chet-2500 password 7 sMudgkin
username chet-3000 password 7 sMudgkin
username chetin password 7 sMudgkin
kerberos local-realm CISCO.COM
kerberos srvtab entry host/chet-2500.cisco.com@CISCO.COM 0 832015393 1 1 8 7 sMudgkin
kerberos server CISCO.COM 172.71.54.14
kerberos credentials forward
!
interface Ethernet0
 ip address 172.16.0.0 255.255.255.0
!
interface Serial0
 no ip address
 shutdown
 no fair-queue
!

interface Serial1
 no ip address
 shutdown
 no fair-queue
!
interface Async2
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic routing
 async mode dedicated
 no cdp enable
 ppp authentication pap local
 no tarp propagate
!
interface Async3
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic address
 async dynamic routing
 async mode dedicated
 no cdp enable
 ppp authentication pap local
 no tarp propagate
!
router eigrp 109
 network 172.17.0.0
 no auto-summary
!
ip default-gateway 172.30.55.64
```

```
ip domain-name cisco.com
ip name-server 192.168.0.0
ip classless
!
!
line con 0
 exec-timeout 0 0
 login authentication console
line 1 16
 transport input all
line aux 0
 transport input all
line vty 0 4
 password sMudgKin
!
ntp clock-period 17179703
ntp peer 172.19.10.0
ntp peer 172.19.0.0
end

chet-2500#
```

With this configuration, the user can Telnet in to the router using Kerberos credentials, as illustrated in the next example:

```
chet-ss20% bin/telnet -a -F chet-2500
Trying 172.16.0.0...
Connected to chet-2500.cisco.com.
Escape character is '^]'.
[ Kerberos V5 accepts you as "chet@CISCO.COM" ]

User Access Verification

chet-2500>[ Kerberos V5 accepted forwarded credentials ]

chet-2500> show kerberos creds
Default Principal:  chet@CISCO.COM
Valid Starting          Expires                 Service Principal
13-May-1996 15:06:25    14-May-1996 00:08:29    krbtgt/CISCO.COM@CISCO.COM

chet-2500>q
Connection closed by foreign host.
chet-ss20%
```

The following example shows how to map Kerberos instances to Cisco's privilege levels. To map Kerberos instances to privilege levels, you would perform the following tasks:

- Entering configuration mode
- Mapping the Kerberos instance admin to privilege level 15
- Mapping the Kerberos instance restricted to privilege level 3
- Specifying that the instance defined by the **kerberos instance map** command be used for AAA Authorization
- Writing the configuration to the terminal

```
chet-2500# configure term
Enter configuration commands, one per line.  End with CNTL/Z.
chet-2500(config)# kerberos instance map admin 15
chet-2500(config)# kerberos instance map restricted 3
chet-2500(config)# aaa authorization exec default krb5-instance
chet-2500(config)#
chet-2500#
```

```
%SYS-5-CONFIG_I: Configured from console by console
chet-2500# write term
Building configuration...

Current configuration:
!
! Last configuration change at 14:59:05 PDT Mon May 13 1996
!
version 11.2
service udp-small-servers
service tcp-small-servers
!
hostname chet-2500
!
aaa new-model
aaa authentication login default krb5-telnet krb5
aaa authentication login console none
aaa authentication ppp default krb5 local
aaa authorization exec default krb5-instance
enable password sMudgKin
!
username chet-2500 password 7 sMudgkin
username chet-3000 password 7 sMudgkin
username chetin password 7 sMudgkin
ip domain-name cisco.com
ip name-server 192.168.0.0
kerberos local-realm CISCO.COM
kerberos srvtab entry host/chet-2500.cisco.com@CISCO.COM 0 832015393 1 1 8 7 sMudgkin
kerberos server CISCO.COM 172.71.54.14
kerberos instance map admin 15
kerberos instance map restricted 3
kerberos credentials forward
clock timezone PST -8
clock summer-time PDT recurring
!
interface Ethernet0
 ip address 172.16.0.0 255.255.255.0
!
interface Serial0
 no ip address
 shutdown
 no fair-queue
!
interface Serial1
 no ip address
 shutdown
 no fair-queue
!
interface Async2
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic routing
 async mode dedicated
 no cdp enable
 ppp authentication pap local
 no tarp propagate
!
interface Async3
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic address
 async dynamic routing
```

```
 async mode dedicated
 no cdp enable
 ppp authentication pap local
 no tarp propagate
!
router eigrp 109
 network 172.17.0.0
no auto-summary
!
ip default-gateway 172.30.55.64
ip classless
!
!
line con 0
 exec-timeout 0 0
 login authentication console
line 1 16
 transport input all
line aux 0
 transport input all
line vty 0 4
 password sMudgKin
!
ntp clock-period 17179703
ntp peer 172.19.10.0
ntp peer 172.19.0.0
end

chet-2500#
```

The following example shows output from the three types of sessions now possible for user chet with Kerberos instances turned on:

```
chet-ss20% telnet chet-2500
Trying 172.16.0.0 ...
Connected to chet-2500.cisco.com.
Escape character is '^]'.

User Access Verification

Username: chet
Password:

chet-2500> show kerberos creds
Default Principal:  chet@CISCO.COM
Valid Starting          Expires                 Service Principal
13-May-1996 14:58:28    13-May-1996 22:59:29    krbtgt/CISCO.COM@CISCO.COM

chet-2500> show privilege
Current privilege level is 1
chet-2500> q
Connection closed by foreign host.
chet-ss20% telnet chet-2500
Trying 172.16.0.0 ...
Connected to chet-2500.cisco.com.
Escape character is '^]'.


User Access Verification

Username: chet/admin
Password:

chet-2500# show kerberos creds
```

```
Default Principal:  chet/admin@CISCO.COM
Valid Starting          Expires                 Service Principal
13-May-1996 14:59:44    13-May-1996 23:00:45    krbtgt/CISCO.COM@CISCO.COM
chet-2500# show privilege
Current privilege level is 15
chet-2500# q
Connection closed by foreign host.
chet-ss20% telnet chet-2500
Trying 172.16.0.0 ...
Connected to chet-2500.cisco.com.
Escape character is '^]'.

User Access Verification

Username: chet/restricted
Password:

chet-2500# show kerberos creds
Default Principal:  chet/restricted@CISCO.COM
Valid Starting          Expires                 Service Principal
13-May-1996 15:00:32    13-May-1996 23:01:33    krbtgt/CISCO.COM@CISCO.COM

chet-2500# show privilege
Current privilege level is 3
chet-2500# q
Connection closed by foreign host.
chet-ss20%
```

Encrypted Telnet Session Example

The following example shows how to establish an encrypted Telnet session from a router to a remote host named “host1”:

```
Router> telnet host1 /encrypt kerberos
```

Cisco Systems

Part 3: Traffic Filtering, Firewalls, and Virus Detection

Access Control Lists: Overview and Guidelines

Cisco provides basic traffic filtering capabilities with access control lists (also referred to as *access lists*). Access lists can be configured for all routed network protocols (IP, AppleTalk, and so on) to filter the packets of those protocols as the packets pass through a router.

You can configure access lists at your router to control access to a network: access lists can prevent certain traffic from entering or exiting a network.

In This Chapter

This chapter describes access lists as part of a security solution. This chapter includes tips, cautions, considerations, recommendations, and general guidelines for how to use access lists.

This chapter has these sections:

- About Access Control Lists
- Overview of Access List Configuration
- Finding Complete Configuration and Command Information for Access Lists

About Access Control Lists

This section briefly describes what access lists do; why and when you should configure access lists; and basic versus advanced access lists.

This section has the following sections:

- What Access Lists Do
- Why You Should Configure Access Lists
- When to Configure Access Lists
- Basic Versus Advanced Access Lists

What Access Lists Do

Access lists filter network traffic by controlling whether routed packets are forwarded or blocked at the router's interfaces. Your router examines each packet to determine whether to forward or drop the packet, on the basis of the criteria you specified within the access lists.

Access list criteria could be the source address of the traffic, the destination address of the traffic, the upper-layer protocol, or other information. Note that sophisticated users can sometimes successfully evade or fool basic access lists because no authentication is required.

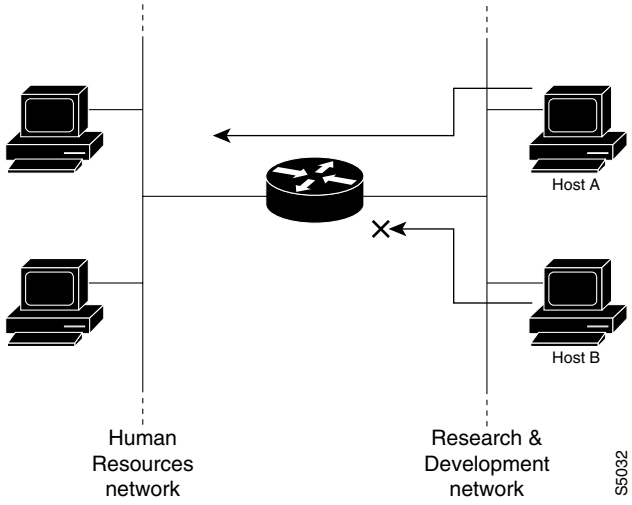
Why You Should Configure Access Lists

There are many reasons to configure access lists; for example, you can use access lists to restrict contents of routing updates or to provide traffic flow control. One of the most important reasons to configure access lists is to provide security for your network, which is the focus of this chapter.

You should use access lists to provide a basic level of security for accessing your network. If you do not configure access lists on your router, all packets passing through the router could be allowed onto all parts of your network.

Access lists can allow one host to access a part of your network and prevent another host from accessing the same area. In Figure 17, host A is allowed to access the Human Resources network, and host B is prevented from accessing the Human Resources network.

Figure 17 Using Traffic Filters to Prevent Traffic from Being Routed to a Network



You can also use access lists to decide which types of traffic are forwarded or blocked at the router interfaces. For example, you can permit e-mail traffic to be routed, but at the same time block all Telnet traffic.

When to Configure Access Lists

Access lists should be used in “firewall” routers, which are often positioned between your internal network and an external network such as the Internet. You can also use access lists on a router positioned between two parts of your network, to control traffic entering or exiting a specific part of your internal network.

To provide the security benefits of access lists, you should at a minimum configure access lists on border routers—routers situated at the edges of your networks. This provides a basic buffer from the outside network, or from a less controlled area of your own network into a more sensitive area of your network.

On these routers, you should configure access lists for each network protocol configured on the router interfaces. You can configure access lists so that inbound traffic or outbound traffic or both are filtered on an interface.

Access lists must be defined on a per-protocol basis. In other words, you should define access lists for every protocol enabled on an interface if you want to control traffic flow for that protocol.

Note Some protocols refer to access lists as *filters*.

Basic Versus Advanced Access Lists

This chapter describes how to use standard and static extended access lists, which are the basic types of access lists. Some type of basic access list should be used with each routed protocol that you have configured for router interfaces.

Besides the basic types of access lists described in this chapter, there are also more advanced access lists available, which provide additional security features and give you greater control over packet transmission. These advanced access lists and features are described in the other chapters within the part "Traffic Filtering and Firewalls."

Overview of Access List Configuration

Each protocol has its own set of specific tasks and rules that are required in order for you to provide traffic filtering. In general, most protocols require at least two basic steps to be accomplished. The first step is to create an access list definition, and the second step is to apply the access list to an interface.

The following sections describe these two steps:

- Creating Access Lists
- Applying Access Lists to Interfaces

Note that some protocols refer to access lists as *filters* and refer to the act of applying the access lists to interfaces as *filtering*.

Creating Access Lists

Create access lists for each protocol you wish to filter, per router interface. For some protocols, you create one access list to filter inbound traffic, and one access list to filter outbound traffic.

To create an access list, you specify the protocol to filter, you assign a unique name or number to the access list, and you define packet filtering criteria. A single access list can have multiple filtering criteria statements.

Cisco recommends that you create your access lists on a TFTP server and then download the access lists to your router. This approach can considerably simplify maintenance of your access lists. For details, see the "Creating and Editing Access List Statements on a TFTP Server" section later in this chapter.

The protocols for which you can configure access lists are identified in Table 22.

This section has the following sections:

- Assigning a Unique Name or Number to Each Access List
- Defining Criteria for Forwarding or Blocking Packets
- Creating and Editing Access List Statements on a TFTP Server

Assigning a Unique Name or Number to Each Access List

When configuring access lists on a router, you must identify each access list uniquely within a protocol by assigning either a name or a number to the protocol's access list.

Note Access lists of some protocols must be identified by a name, and access lists of other protocols must be identified by a number. Some protocols can be identified by either a name or a number. When a number is used to identify an access list, the number must be within the specific range of numbers that is valid for the protocol.

You can specify access lists by names for the following protocols:

- Apollo Domain
- IP
- IPX
- ISO CLNS
- NetBIOS IPX
- Source-route bridging NetBIOS

You can specify access lists by numbers for the protocols listed in Table 22. Table 22 also lists the range of access list numbers that is valid for each protocol.

Table 22 Protocols with Access Lists Specified by Numbers

Protocol	Range
IP	1–99, 1300–1999
Extended IP	100–199, 2000–2699
Ethernet type code	200–299
Ethernet address	700–799
Transparent bridging (protocol type)	200–299
Transparent bridging (vendor code)	700–799
Extended transparent bridging	1100–1199
DECnet and extended DECnet	300–399
XNS	400–499
Extended XNS	500–599
AppleTalk	600–699
Source-route bridging (protocol type)	200–299
Source-route bridging (vendor code)	700–799

Table 22 Protocols with Access Lists Specified by Numbers (continued)

Protocol	Range
IPX	800–899
Extended IPX	900–999
IPX SAP	1000–1099
Standard VINES	1–100
Extended VINES	101–200
Simple VINES	201–300

Defining Criteria for Forwarding or Blocking Packets

When creating an access list, you define criteria that are applied to each packet that is processed by the router; the router decides whether to forward or block each packet on the basis of whether or not the packet matches the criteria.

Typical criteria you define in access lists are packet source addresses, packet destination addresses, and upper-layer protocol of the packet. However, each protocol has its own specific set of criteria that can be defined.

For a single access list, you can define multiple criteria in multiple, separate access list statements. Each of these statements should reference the same identifying name or number, to tie the statements to the same access list. You can have as many criteria statements as you want, limited only by the available memory. Of course, the more statements you have, the more difficult it will be to comprehend and manage your access lists.

The Implied "Deny All Traffic" Criteria Statement

At the end of every access list is an implied "deny all traffic" criteria statement. Therefore, if a packet does not match any of your criteria statements, the packet will be blocked.

Note For most protocols, if you define an inbound access list for traffic filtering, you should include explicit access list criteria statements to permit routing updates. If you do not, you might effectively lose communication from the interface when routing updates are blocked by the implicit "deny all traffic" statement at the end of the access list.

The Order in Which You Enter Criteria Statements

Note that each additional criteria statement that you enter is appended to the *end* of the access list statements. Also note that you cannot delete individual statements after they have been created. You can only delete an entire access list.

The order of access list statements is important! When the router is deciding whether to forward or block a packet, the Cisco IOS software tests the packet against each criteria statement in the order in which the statements were created. After a match is found, no more criteria statements are checked.

If you create a criteria statement that explicitly permits all traffic, no statements added later will ever be checked. If you need additional statements, you must delete the access list and retype it with the new entries.

Creating and Editing Access List Statements on a TFTP Server

Because the order of access list criteria statements is important, and because you cannot reorder or delete criteria statements on your router, Cisco recommends that you create all access list statements on a TFTP server, and then download the entire access list to your router.

To use a TFTP server, create the access list statements using any text editor, and save the access list in ASCII format to a TFTP server that is accessible by your router. Then, from your router, use the **copy tftp:***file_id* **system:running-config** command to copy the access list to your router. Finally, perform the **copy system:running-config nvram:startup-config** command to save the access list to your router's NVRAM.

Then, if you ever want to make changes to an access list, you can make them to the text file on the TFTP server, and copy the edited file to your router as before.

Note The first command of an edited access list file should delete the previous access list (for example, type a **no access-list** command at the beginning of the file). If you do not first delete the previous version of the access list, when you copy the edited file to your router you will merely be appending additional criteria statements to the end of the existing access list.

Applying Access Lists to Interfaces

For some protocols, you can apply up to two access lists to an interface: one inbound access list and one outbound access list. With other protocols, you apply only one access list which checks both inbound and outbound packets.

If the access list is inbound, when the router receives a packet, the Cisco IOS software checks the access list's criteria statements for a match. If the packet is permitted, the software continues to process the packet. If the packet is denied, the software discards the packet.

If the access list is outbound, after receiving and routing a packet to the outbound interface, the software checks the access list's criteria statements for a match. If the packet is permitted, the software transmits the packet. If the packet is denied, the software discards the packet.

Note Access lists that are applied to interfaces do not filter traffic that originates from that router.

Finding Complete Configuration and Command Information for Access Lists

The guidelines discussed in this chapter apply in general to all protocols. The specific instructions for creating access lists and applying them to interfaces vary from protocol to protocol, and this specific information is not included in this chapter.

To find complete configuration and command information to configure access lists for a specific protocol, see the corresponding chapters in the Cisco IOS configuration guides and command references. For example, to configure access lists for the IP protocol, refer to the "Configuring IP Services" chapter in the *Cisco IOS IP Configuration Guide*.

For information on dynamic access lists, see the chapter "Configuring Lock-and-Key Security (Dynamic Access Lists)" later in this book.

For information on reflexive access lists, see the chapter "Configuring IP Session Filtering (Reflexive Access Lists)" later in this book.

Cisco IOS Firewall Overview

This chapter describes how you can configure your Cisco networking device to function as a firewall, using Cisco IOS Firewall security features.

This chapter has the following sections:

- About Firewalls
- The Cisco IOS Firewall Solution
- Creating a Customized Firewall
- Other Guidelines for Configuring Your Firewall

About Firewalls

Firewalls are networking devices that control access to your organization's network assets. Firewalls are positioned at the entrance points into your network. If your network has multiple entrance points, you must position a firewall at each point to provide effective network access control.

Firewalls are often placed in between the internal network and an external network such as the Internet. With a firewall between your network and the Internet, all traffic coming from the Internet must pass through the firewall before entering your network.

Firewalls can also be used to control access to a specific part of your network. For example, you can position firewalls at all the entry points into a research and development network to prevent unauthorized access to proprietary information.

The most basic function of a firewall is to monitor and filter traffic. Firewalls can be simple or elaborate, depending on your network requirements. Simple firewalls are usually easier to configure and manage. However, you might require the flexibility of a more elaborate firewall.

The Cisco IOS Firewall Solution

Cisco IOS software provides an extensive set of security features, allowing you to configure a simple or elaborate firewall, according to your particular requirements. You can configure a Cisco device as a firewall if the device is positioned appropriately at a network entry point. Security features that provide firewall functionality are listed in the "Creating a Customized Firewall" section.

In addition to the security features available in standard Cisco IOS feature sets, Cisco IOS Firewall gives your router additional firewall capabilities.

The Cisco IOS Firewall Feature Set

The Cisco IOS Firewall feature set combines existing Cisco IOS firewall technology and Context-based Access Control (CBAC). When you configure the Cisco IOS Firewall on your Cisco router, you turn your router into an effective, robust firewall.

The Cisco IOS Firewall features are designed to prevent unauthorized external individuals from gaining access to your internal network and to block attacks on your network, while at the same time allowing authorized users to access network resources.

You can use the Cisco IOS Firewall features to configure your Cisco IOS router as one of the following:

- An Internet firewall or part of an Internet firewall
- A firewall between groups in your internal network
- A firewall providing secure connections to or from branch offices
- A firewall between your company's network and your company's partners' networks

The Cisco IOS Firewall features provide the following benefits:

- Protection of internal networks from intrusion
- Monitoring of traffic through network perimeters
- Enabling of network commerce via the World Wide Web

Creating a Customized Firewall

To create a firewall customized to fit your organization's security policy, you should determine which Cisco IOS Firewall features are appropriate, and configure those features. At a minimum, you must configure basic traffic filtering to provide a basic firewall. You can configure your Cisco networking device to function as a firewall by using the following Cisco IOS Firewall features:

- Standard Access Lists and Static Extended Access Lists
- Reflexive Access Lists
- Lock-and-Key (Dynamic Access Lists)
- TCP Intercept
- Context-based Access Control
- Intrusion Prevention System (IPS) (formerly known as Cisco IOS Firewall Intrusion Detection System)
- Authentication Proxy
- Port to Application Mapping
- Security Server Support
- Network Address Translation
- IPSec Network Security
- Neighbor Router Authentication
- Event Logging
- User Authentication and Authorization

In addition to configuring these features, you should follow the guidelines listed in the "Other Guidelines for Configuring Your Firewall" section. This section outlines important security practices to protect your firewall and network. Table 23 describes Cisco IOS security features.

Table 23 Cisco IOS Features for a Robust Firewall

Feature	Chapter	Comments
Standard Access Lists and Static Extended Access Lists	"Access Control Lists: Overview and Guidelines"	Standard and static extended access lists provide basic traffic filtering capabilities. You configure criteria that describe which packets should be forwarded, and which packets should be dropped at an interface, based on each packet's network layer information. For example, you can block all UDP packets from a specific source IP address or address range. Some extended access lists can also examine transport layer information to determine whether to block or forward packets. To configure a basic firewall, you should at a minimum configure basic traffic filtering. You should configure basic access lists for all network protocols that will be routed through your firewall, such as IP, IPX, AppleTalk, and so forth.
Lock-and-Key (Dynamic Access Lists)	"Configuring Lock-and-Key Security (Dynamic Access Lists)"	Lock-and-Key provides traffic filtering with the ability to allow temporary access through the firewall for certain individuals. These individuals must first be authenticated (by a username/password mechanism) before the firewall allows their traffic through the firewall. Afterwards, the firewall closes the temporary opening. This provides tighter control over traffic at the firewall than with standard or static extended access lists.
Reflexive Access Lists	"Configuring IP Session Filtering (Reflexive Access Lists)"	Reflexive access lists filter IP traffic so that TCP or UDP "session" traffic is only permitted through the firewall if the session originated from within the internal network. You would only configure Reflexive Access Lists when not using Context-based Access Control.
TCP Intercept	"Configuring TCP Intercept (Preventing Denial-of-Service Attacks)"	TCP Intercept protects TCP servers within your network from TCP SYN-flooding attacks, a type of denial-of-service attack. You would only configure TCP Intercept when not using Context-based Access Control.

Table 23 Cisco IOS Features for a Robust Firewall (continued)

Feature	Chapter	Comments
Context-based Access Control	"Configuring Context-Based Access Control"	CBAC examines not only network layer and transport layer information, but also examines the application-layer protocol information (such as FTP information) to learn about the state of TCP and UDP connections. CBAC maintains connection state information for individual connections. This state information is used to make intelligent decisions about whether packets should be permitted or denied, and dynamically creates and deletes temporary openings in the firewall. CBAC also generates real-time alerts and audit trails. Enhanced audit trail features use SYSLOG to track all network transactions. Real-time alerts send SYSLOG error messages to central management consoles upon detecting suspicious activity. Using CBAC inspection rules, you can configure alerts and audit trail information on a per-application protocol basis. CBAC is only available in the Cisco IOS Firewall feature set.
Cisco IOS Intrusion Prevention System (IPS)	"Configuring Cisco IOS Intrusion Prevention System (IPS) "	The Cisco IOS IPS acts as an in-line intrusion detection sensor, "watching" packets and sessions as they flow through the router and scanning each packet to match any of the Cisco IOS IPS signatures. When Cisco IOS IPS detects suspicious activity, it responds before network security can be compromised and logs the event through Cisco IOS syslog messages or Security Device Event Exchange (SDEE). The network administrator can configure Cisco IOS IPS to choose the appropriate response to various threats. When packets in a session match a signature, Cisco IOS IPS can take any of the following actions, as appropriate: • Send an alarm to a syslog server or a centralized management interface • Drop the packet • Reset the connection • Deny traffic from the source IP address of the attacker for a specified amount of time • Deny traffic on the connection for which the signature was seen for a specified amount of time
Authentication Proxy	"Configuring Authentication Proxy"	The Cisco IOS Firewall authentication proxy feature allows network administrators to apply specific security policies on a per-user basis. Previously, user identity and related authorized access was associated with a user's IP address, or a single security policy had to be applied to an entire user group or sub network. Now, users can be identified and authorized on the basis of their per-user policy, and access privileges tailored on an individual basis are possible, as opposed to general policy applied across multiple users.

Table 23 Cisco IOS Features for a Robust Firewall (continued)

Feature	Chapter	Comments
Port to Application Mapping	"Configuring Port to Application Mapping"	Port to Application Mapping (PAM) is a feature of Cisco IOS Firewall. PAM allows you to customize TCP or UDP port numbers for network services or applications. PAM uses this information to support network environments that run services using ports that are different from the registered or well-known ports associated with an application. The information in the PAM table enables CBAC supported services to run on nonstandard ports.
Security Server Support	"Configuring TACACS+," "Configuring RADIUS," and "Configuring Kerberos"	The Cisco IOS Firewall feature set can be configured as a client of the following supported security servers: • TACACS+ (including CiscoSecure) • RADIUS • Kerberos You can use any of these security servers to store a database of user profiles. To gain access into your firewall or to gain access through the firewall into another network, users must enter authentication information (such as a username and password), which is matched against the information on the security server. When users pass authentication, they are granted access according to their specified privileges.
Network Address Translation	"Configuring NAT for IP Address Conservation"	You can use Network Address Translation (NAT) to hide internal IP network addresses from the world outside the firewall. NAT was designed to provide IP address conservation and for internal IP networks that have unregistered (not globally unique) IP addresses: NAT translates these unregistered IP addresses into legal addresses at the firewall. NAT can also be configured to advertise only one address for the entire internal network to the outside world. This provides security by effectively hiding the entire internal network from the world. NAT gives you limited spoof protection because internal addresses are hidden. Additionally, NAT removes all your internal services from the external name space. NAT does not work with the application-layer protocols RPC, VDOLive, or SQL*Net "Redirected." (NAT does work with SQL*Net "Bequeathed.") Do not configure NAT with networks that will carry traffic for these incompatible protocols.

Table 23 Cisco IOS Features for a Robust Firewall (continued)

Feature	Chapter	Comments
IPSec Network Security	"Configuring Security for VPNs with IPSec "	IPSec is a framework of open standards developed by the Internet Engineering Task Force (IETF) that provides security for transmission of sensitive information over unprotected networks such as the Internet. IPSec acts at the network layer, protecting and authenticating IP packets between participating IPSec devices ("peers") such as Cisco routers.
Neighbor Router Authentication	"Neighbor Router Authentication: Overview and Guidelines"	Neighbor router authentication requires the firewall to authenticate all neighbor routers before accepting any route updates from that neighbor. This ensures that the firewall receives legitimate route updates from a trusted source.
Event Logging	"System Monitoring and Logging" chapter in the *Cisco IOS Network Management Configuration Guide*	Event logging automatically logs output from system error messages and other events to the console terminal. You can also redirect these messages to other destinations such as virtual terminals, internal buffers, or syslog servers. You can also specify the severity of the event to be logged, and you can configure the logged output to be timestamped. The logged output can be used to assist real-time debugging and management, and to track potential security breaches or other nonstandard activities throughout a network.
User Authentication and Authorization	"Configuring Authentication" and "Configuring Authorization"	Authentication and authorization help protect your network from access by unauthorized users.

Other Guidelines for Configuring Your Firewall

As with all networking devices, you should always protect access into the firewall by configuring passwords as described in the module "Configuring Security with Passwords, Privilege Levels, and Login Usernames for CLI Sessions on Networking Devices." You should also consider configuring user authentication, authorization, and accounting as described in the chapters in the "Authentication, Authorization, and Accounting (AAA)" part of this guide.

You should also consider the following recommendations:

- When setting passwords for privileged access to the firewall, use the **enable secret** command rather than the **enable password** command, which does not have as strong an encryption algorithm.
- Put a password on the console port. In authentication, authorization, and accounting (AAA) environments, use the same authentication for the console as for elsewhere. In a non-AAA environment, at a minimum configure the **login** and **password** *password* commands.
- Think about access control *before* you connect a console port to the network in any way, including attaching a modem to the port. Be aware that a *break* on the console port might give total control of the firewall, even with access control configured.
- Apply access lists and password protection to all virtual terminal ports. Use access lists to limit who can Telnet into your router.

- Do not enable any local service (such as SNMP or NTP) that you do not use. Cisco Discovery Protocol (CDP) and Network Time Protocol (NTP) are on by default, and you should turn these off if you do not need them.

  To turn off CDP, enter the **no cdp run** global configuration command. To turn off NTP, enter the **ntp disable** interface configuration command on each interface not using NTP.

  If you must run NTP, configure NTP only on required interfaces, and configure NTP to listen only to certain peers.

  Any enabled service could present a potential security risk. A determined, hostile party might be able to find creative ways to misuse the enabled services to access the firewall or the network.

  For local services that are enabled, protect against misuse. Protect by configuring the services to communicate only with specific peers, and protect by configuring access lists to deny packets for the services at specific interfaces.

- Protect against spoofing: protect the networks on both sides of the firewall from being spoofed from the other side. You could protect against spoofing by configuring input access lists at all interfaces to pass only traffic from expected source addresses, and to deny all other traffic.

  You should also disable source routing. For IP, enter the **no ip source-route** global configuration command. Disabling source routing at *all* routers can also help prevent spoofing.

  You should also disable minor services. For IP, enter the **no service tcp-small-servers** and **no service udp-small-servers** global configuration commands.

- Prevent the firewall from being used as a relay by configuring access lists on any asynchronous Telnet ports.

- Normally, you should disable directed broadcasts for all applicable protocols on your firewall and on all your other routers. For IP, use the **no ip directed-broadcast** command. Rarely, some IP networks do require directed broadcasts; if this is the case, do not disable directed broadcasts.

  Directed broadcasts can be misused to multiply the power of denial-of-service attacks, because every denial-of-service packet sent is broadcast to every host on a subnet. Furthermore, some hosts have other intrinsic security risks present when handling broadcasts.

- Configure the **no ip proxy-arp** command to prevent internal addresses from being revealed. (This is important to do if you do not already have NAT configured to prevent internal addresses from being revealed.)

- Keep the firewall in a secured (locked) room.

Configuring Lock-and-Key Security (Dynamic Access Lists)

This chapter describes how to configure lock-and-key security at your router. Lock-and-key is a traffic filtering security feature available for the IP protocol.

For a complete description of lock-and-key commands, refer to the "Lock-and-Key Commands" chapter of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Finding Additional Feature Support Information" section on page cxvii in the chapter Using Cisco IOS Software for Release 12.4.

In This Chapter

This chapter has the following sections:

- About Lock-and-Key
- Compatibility with Releases Before Cisco IOS Release 11.1
- Risk of Spoofing with Lock-and-Key
- Router Performance Impacts with Lock-and-Key
- Prerequisites to Configuring Lock-and-Key
- Configuring Lock-and-Key
- Verifying Lock-and-Key Configuration
- Maintaining Lock-and-Key
- Lock-and-Key Configuration Examples

About Lock-and-Key

Lock-and-key is a traffic filtering security feature that dynamically filters IP protocol traffic. Lock-and-key is configured using IP dynamic extended access lists. Lock-and-key can be used in conjunction with other standard access lists and static extended access lists.

When lock-and-key is configured, designated users whose IP traffic is normally blocked at a router can gain temporary access through the router. When triggered, lock-and-key reconfigures the interface's existing IP access list to permit designated users to reach their designated host(s). Afterwards, lock-and-key reconfigures the interface back to its original state.

For a user to gain access to a host through a router with lock-and-key configured, the user must first open a Telnet session to the router. When a user initiates a standard Telnet session to the router, lock-and-key automatically attempts to authenticate the user. If the user is authenticated, they will then gain temporary access through the router and be able to reach their destination host.

This section has the following sections:

- Benefits of Lock-and-Key
- When to Use Lock-and-Key
- How Lock-and-Key Works

Benefits of Lock-and-Key

Lock-and-key provides the same benefits as standard and static extended access lists (these benefits are discussed in the chapter "Access Control Lists: Overview and Guidelines"). However, lock-and-key also has the following security benefits over standard and static extended access lists:

- Lock-and-key uses a challenge mechanism to authenticate individual users.
- Lock-and-key provides simpler management in large internetworks.
- In many cases, lock-and-key reduces the amount of router processing required for access lists.
- Lock-and-key reduces the opportunity for network break-ins by network hackers.

With lock-and-key, you can specify which users are permitted access to which source and destination hosts. These users must pass a user authentication process before they are permitted access to their designated hosts. Lock-and-key creates dynamic user access through a firewall, without compromising other configured security restrictions.

When to Use Lock-and-Key

Two examples of when you might use lock-and-key follow:

- When you want a specific remote user (or group of remote users) to be able to access a host within your network, connecting from their remote hosts via the Internet. Lock-and-key authenticates the user, then permits limited access through your firewall router for the individual's host or subnet, for a finite period of time.
- When you want a subset of hosts on a local network to access a host on a remote network protected by a firewall. With lock-and-key, you can enable access to the remote host only for the desired set of local user's hosts. Lock-and-key require the users to authenticate through a TACACS+ server, or other security server, before allowing their hosts to access the remote hosts.

How Lock-and-Key Works

The following process describes the lock-and-key access operation:

1. A user opens a Telnet session to a border (firewall) router configured for lock-and-key. The user connects via the virtual terminal port on the router.
2. The Cisco IOS software receives the Telnet packet, opens a Telnet session, prompts for a password, and performs a user authentication process. The user must pass authentication before access through the router is allowed. The authentication process can be done by the router or by a central access security server such as a TACACS+ or RADIUS server.
3. When the user passes authentication, they are logged out of the Telnet session, and the software creates a temporary entry in the dynamic access list. (Per your configuration, this temporary entry can limit the range of networks to which the user is given temporary access.)
4. The user exchanges data through the firewall.
5. The software deletes the temporary access list entry when a configured timeout is reached, or when the system administrator manually clears it. The configured timeout can either be an idle timeout or an absolute timeout.

Note The temporary access list entry is not automatically deleted when the user terminates a session. The temporary access list entry remains until a configured timeout is reached or until it is cleared by the system administrator.

Compatibility with Releases Before Cisco IOS Release 11.1

Enhancements to the **access-list** command are used for lock-and-key. These enhancements are backward compatible—if you migrate from a release before Cisco IOS Release 11.1 to a newer release, your access lists will be automatically converted to reflect the enhancements. However, if you try to use lock-and-key with a release before Cisco IOS Release 11.1, you might encounter problems as described in the following caution paragraph:

Caution Cisco IOS releases before Release 11.1 are not upwardly compatible with the lock-and-key access list enhancements. Therefore, if you save an access list with software older than Release 11.1, and then use this software, the resulting access list will not be interpreted correctly. *This could cause you severe security problems.* You must save your old configuration files with Cisco IOS Release 11.1 or later software before booting an image with these files.

Risk of Spoofing with Lock-and-Key

Caution Lock-and-key access allows an external event (a Telnet session) to place an opening in the firewall. While this opening exists, the router is susceptible to source address spoofing.

When lock-and-key is triggered, it creates a dynamic opening in the firewall by temporarily reconfiguring an interface to allow user access. While this opening exists, another host might spoof the authenticated user's address to gain access behind the firewall. Lock-and-key does not cause the address spoofing problem; the problem is only identified here as a concern to the user. Spoofing is a problem inherent to all access lists, and lock-and-key does not specifically address this problem.

To prevent spoofing, configure encryption so that traffic from the remote host is encrypted at a secured remote router, and decrypted locally at the router interface providing lock-and-key. You want to ensure that all traffic using lock-and-key will be encrypted when entering the router; this way no hackers can spoof the source address, because they will be unable to duplicate the encryption or to be authenticated as is a required part of the encryption setup process.

Router Performance Impacts with Lock-and-Key

When lock-and-key is configured, router performance can be affected in the following ways:

- When lock-and-key is triggered, the dynamic access list forces an access list rebuild on the silicon switching engine (SSE). This causes the SSE switching path to slow down momentarily.
- Dynamic access lists require the idle timeout facility (even if the timeout is left to default) and therefore cannot be SSE switched. These entries must be handled in the protocol fast-switching path.
- When remote users trigger lock-and-key at a border router, additional access list entries are created on the border router interface. The interface's access list will grow and shrink dynamically. Entries are dynamically removed from the list after either the idle-timeout or max-timeout period expires. Large access lists can degrade packet switching performance, so if you notice performance problems, you should look at the border router configuration to see if you should remove temporary access list entries generated by lock-and-key.

Prerequisites to Configuring Lock-and-Key

Lock-and-key uses IP extended access lists. You must have a solid understanding of how access lists are used to filter traffic, before you attempt to configure lock-and-key. Access lists are described in the chapter "Access Control Lists: Overview and Guidelines."

Lock-and-key employs user authentication and authorization as implemented in Cisco's authentication, authorization, and accounting (AAA) paradigm. You must understand how to configure AAA user authentication and authorization before you configure lock-and-key. User authentication and authorization is explained in the "Authentication, Authorization, and Accounting (AAA)" part of this document.

Lock-and-key uses the **autocommand** command, which you should understand. This command is described in the "Modem Support and Asynchronous Device Commands" chapter of the *Cisco IOS Dial Technologies Command Reference.*

Configuring Lock-and-Key

To configure lock-and-key, use the following commands beginning in global configuration mode. While completing these steps, be sure to follow the guidelines listed in the "Lock-and-Key Configuration Guidelines" section of this chapter.

	Command	Purpose
Step 1	`Router(config)# access-list access-list-number [dynamic dynamic-name [timeout minutes]] {deny | permit} telnet source source-wildcard destination destination-wildcard [precedence precedence] [tos tos] [established] [log]`	Configures a dynamic access list, which serves as a template and placeholder for temporary access list entries.
Step 2	`Router(config)# access-list dynamic-extend`	(Optional) Extends the absolute timer of the dynamic ACL by six minutes when you open another Telnet session into the router to re-authenticate yourself using lock-and-key. Use this command if your job will run past the ACL's absolute timer.
Step 3	`Router(config)# interface type number`	Configures an interface and enters interface configuration mode.
Step 4	`Router(config-if)# ip access-group access-list-number`	Applies the access list to the interface.
Step 5	`Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 6	`Router(config)# line vty line-number [ending-line-number]`	Defines one or more virtual terminal (VTY) ports and enters line configuration mode. If you specify multiple VTY ports, they must all be configured identically because the software hunts for available VTY ports on a round-robin basis. If you do not want to configure all your VTY ports for lock-and-key access, you can specify a group of VTY ports for lock-and-key support only.
Step 7	`Router(config-line)# login tacacs` or `Router(config-line)# password password` or `Router(config-line)# login local` or `Router(config-line)# exit` then `Router(config)# username name password secret`	Configures user authentication in line or global configuration mode.
Step 8	`Router(config-line)# autocommand access-enable [host] [timeout minutes]` or `Router(config)# autocommand access-enable [host] [timeout minutes]`	Enables the creation of temporary access list entries in line or global configuration mode. If the optional **host** keyword is *not* specified, all hosts on the entire network are allowed to set up a temporary access list entry. The dynamic access list contains the network mask to enable the new network connection.

For an example of a lock-and-key configuration, see the section "Lock-and-Key Configuration Examples" later in this chapter.

Lock-and-Key Configuration Guidelines

Before you configure lock-and-key, you should understand the guidelines discussed in the following sections:

- Dynamic Access Lists
- Lock-and-Key Authentication
- The autocommand Command

Dynamic Access Lists

Use the following guidelines for configuring dynamic access lists:

- Do *not* create more than one dynamic access list for any one access list. The software only refers to the first dynamic access list defined.
- Do *not* assign the same *dynamic-name* to another access list. Doing so instructs the software to reuse the existing list. All named entries must be globally unique within the configuration.
- Assign attributes to the dynamic access list in the same way you assign attributes for a static access list. The temporary access list entries inherit the attributes assigned to this list.
- Configure Telnet as the protocol so that users must open a Telnet session into the router to be authenticated before they can gain access through the router.
- Either define an idle timeout now with the **timeout** keyword in the **access-enable** command in the **autocommand** command, or define an absolute timeout value later with the **access-list** command. You must define either an idle timeout or an absolute timeout—otherwise, the temporary access list entry will remain configured indefinitely on the interface (even after the user has terminated their session) until the entry is removed manually by an administrator. (You could configure both idle and absolute timeouts if you wish.)
- If you configure an idle timeout, the idle timeout value should be equal to the WAN idle timeout value.
- If you configure both idle and absolute timeouts, the idle timeout value must be less than the absolute timeout value.
- If you realize that a job will run past the ACL's absolute timer, use the **access-list dynamic-extend** command to extend the absolute timer of the dynamic ACL by six minutes. This command allows you to open a new Telnet session into the router to re-authentication yourself using lock-and-key.
- The only values replaced in the temporary entry are the source or destination address, depending whether the access list was in the input access list or output access list. All other attributes, such as port, are inherited from the main dynamic access list.
- Each addition to the dynamic list is always put at the beginning of the dynamic list. You cannot specify the order of temporary access list entries.
- Temporary access list entries are never written to NVRAM.
- To manually clear or to display dynamic access lists, refer to the section "Maintaining Lock-and-Key" later in this chapter.

Lock-and-Key Authentication

There are three possible methods to configure an authentication query process. These three methods are described in this section.

Note Cisco recommends that you use the TACACS+ server for your authentication query process. TACACS+ provides authentication, authorization, and accounting services. It also provides protocol support, protocol specification, and a centralized security database. Using a TACACS+ server is described in the next section, "Method 1—Configuring a Security Server."

Method 1—Configuring a Security Server

Use a network access security server such as TACACS+ server. This method requires additional configuration steps on the TACACS+ server but allows for stricter authentication queries and more sophisticated tracking capabilities.

```
Router(config-line)# login tacacs
```

Method 2—Configuring the username Command

Use the **username** command. This method is more effective because authentication is determined on a user basis.

```
Router(config)# username name {nopassword | password {mutual-password | encryption-type
encryption-password}}
```

Method 3—Configuring the password and login Commands

Use the **password** and **login** commands. This method is less effective because the password is configured for the port, not for the user. Therefore, any user who knows the password can authenticate successfully.

```
Router(config-line)# password password
Router(config-line)# login local
```

The autocommand Command

Use the following guidelines for configuring the **autocommand** command:

- If you use a TACACS+ server to authenticate the user, you should configure the **autocommand** command on the TACACS+ server as a per-user autocommand. If you use local authentication, use the **autocommand** command on the line.
- Configure all virtual terminal (VTY) ports with the same **autocommand** command. Omitting an **autocommand** command on a VTY port allows a random host to gain EXEC mode access to the router and does not create a temporary access list entry in the dynamic access list.
- If you did not previously define an idle timeout with the **autocommand access-enable** command, you must define an absolute timeout now with the **access-list** command. You must define either an idle timeout or an absolute timeout—otherwise, the temporary access list entry will remain configured indefinitely on the interface (even after the user has terminated their session) until the entry is removed manually by an administrator. (You could configure both idle and absolute timeouts if you wish.)
- If you configure both idle and absolute timeouts, the absolute timeout value must be greater than the idle timeout value.

Verifying Lock-and-Key Configuration

You can verify that lock-and-key is successfully configured on the router by asking a user to test the connection. The user should be at a host that is permitted in the dynamic access list, and the user should have AAA authentication and authorization configured.

To test the connection, the user should Telnet to the router, allow the Telnet session to close, and then attempt to access a host on the other side of the router. This host must be one that is permitted by the dynamic access list. The user should access the host with an application that uses the IP protocol.

The following sample display illustrates what end-users might see if they are successfully authenticated. Notice that the Telnet connection is closed immediately after the password is entered and authenticated. The temporary access list entry is then created, and the host that initiated the Telnet session now has access inside the firewall.

```
Router% telnet corporate
Trying 172.21.52.1 ...
Connected to corporate.example.com.
Escape character is '^]'.
User Access Verification
Password:Connection closed by foreign host.
```

You can then use the **show access-lists** command at the router to view the dynamic access lists, which should include an additional entry permitting the user access through the router.

Maintaining Lock-and-Key

When lock-and-key is in use, dynamic access lists will dynamically grow and shrink as entries are added and deleted. You need to make sure that entries are being deleted in a timely way, because while entries exist, the risk of a spoofing attack is present. Also, the more entries there are, the bigger the router performance impact will be.

If you do not have an idle or absolute timeout configured, entries will remain in the dynamic access list until you manually remove them. If this is the case, make sure that you are extremely vigilant about removing entries.

Displaying Dynamic Access List Entries

You can display temporary access list entries when they are in use. After a temporary access list entry is cleared by you or by the absolute or idle timeout parameter, it can no longer be displayed. The number of matches displayed indicates the number of times the access list entry was hit.

To view dynamic access lists and any temporary access list entries that are currently established, use the following command in privileged EXEC mode:

Command	Purpose
Router# **show access-lists** [*access-list-number*]	Displays dynamic access lists and temporary access list entries.

Manually Deleting Dynamic Access List Entries

To manually delete a temporary access list entry, use the following command in privileged EXEC mode:

Command	Purpose
Router# **clear access-template** [*access-list-number* | *name*] [*dynamic-name*] [*source*] [*destination*]	Deletes a dynamic access list.

Lock-and-Key Configuration Examples

The following sections provide lock-and-key configuration examples:

- Lock-and-Key with Local Authentication Example
- Lock-and-Key with TACACS+ Authentication Example

Cisco recommends that you use a TACACS+ server for authentication, as shown in the second example.

Lock-and-Key with Local Authentication Example

This example shows how to configure lock-and-key access, with authentication occurring locally at the router. Lock-and-key is configured on the Ethernet 0 interface.

```
interface ethernet0
 ip address 172.18.23.9 255.255.255.0
 ip access-group 101 in

access-list 101 permit tcp any host 172.18.21.2 eq telnet
access-list 101 dynamic mytestlist timeout 120 permit ip any any

line vty 0
login local
autocommand access-enable timeout 5
```

The first access-list entry allows only Telnet into the router. The second access-list entry is always ignored until lock-and-key is triggered.

In the **access-list** command, the timeout is the absolute timeout. In this example, the lifetime of the mytestlist ACL is 120 minutes; that is, when a user logs in and enable the **access-enable** command, a dynamic ACL is created for 120 minutes (the maximum absolute time). The session is closed after 120 minutes, whether or not anyone is using it.

In the **autocommand** command, the timeout is the idle timeout. In this example, each time the user logs in or authenticates there is a 5-minute session. If there is no activity, the session closes in 5 minutes and the user has to reauthenticate. If the user uses the connection, the absolute time takes affect and the session closes in 120 minutes.

After a user opens a Telnet session into the router, the router will attempt to authenticate the user. If authentication is successful, the **autocommand** executes and the Telnet session terminates. The **autocommand** creates a temporary inbound access list entry at the Ethernet 0 interface, based on the second access-list entry (mytestlist). This temporary entry will expire after 5 minutes, as specified by the timeout.

Lock-and-Key with TACACS+ Authentication Example

The following example shows how to configure lock-and-key access, with authentication on a TACACS+ server. Lock-and-key access is configured on the BRI0 interface. Four VTY ports are defined with the password “cisco”.

```
aaa authentication login default group tacacs+ enable
aaa accounting exec stop-only group tacacs+
aaa accounting network stop-only group tacacs+
enable password ciscotac
!
isdn switch-type basic-dms100
!
interface ethernet0
ip address 172.18.23.9 255.255.255.0
!
interface BRI0
 ip address 172.18.21.1 255.255.255.0
 encapsulation ppp
 dialer idle-timeout 3600
 dialer wait-for-carrier-time 100
 dialer map ip 172.18.21.2 name diana
 dialer-group 1
 isdn spid1 2036333715291
 isdn spid2 2036339371566
 ppp authentication chap
 ip access-group 102 in
!
access-list 102 permit tcp any host 172.18.21.2 eq telnet
access-list 102 dynamic testlist timeout 5 permit ip any any
!
!
ip route 172.18.250.0 255.255.255.0 172.18.21.2
priority-list 1 interface BRI0 high
tacacs-server host 172.18.23.21
tacacs-server host 172.18.23.14
tacacs-server key test1
tftp-server rom alias all
!
dialer-list 1 protocol ip permit
!
line con 0
 password cisco
line aux 0
 line VTY 0 4
 autocommand access-enable timeout 5
 password cisco
!
```

Configuring IP Session Filtering (Reflexive Access Lists)

This chapter describes how to configure reflexive access lists on your router. Reflexive access lists provide the ability to filter network traffic at a router, based on IP upper-layer protocol "session" information.

For a complete description of reflexive access list commands, refer to the "Reflexive Access List Commands" chapter of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Finding Additional Feature Support Information" section on page cxvii in the chapter Using Cisco IOS Software for Release 12.4.

In This Chapter

This chapter has the following sections:

- About Reflexive Access Lists
- Prework: Before You Configure Reflexive Access Lists
- Reflexive Access Lists Configuration Task List
- Reflexive Access List Configuration Examples

About Reflexive Access Lists

Reflexive access lists allow IP packets to be filtered based on upper-layer session information. You can use reflexive access lists to permit IP traffic for sessions originating from within your network but to deny IP traffic for sessions originating from outside your network. This is accomplished by reflexive filtering, a kind of session filtering.

Reflexive access lists can be defined with extended named IP access lists only. You cannot define reflexive access lists with numbered or standard named IP access lists or with other protocol access lists.

You can use reflexive access lists in conjunction with other standard access lists and static extended access lists.

This section has the following sections:

- Benefits of Reflexive Access Lists
- What Is a Reflexive Access List?
- How Reflexive Access Lists Implement Session Filtering
- Where to Configure Reflexive Access Lists
- How Reflexive Access Lists Work
- Restrictions on Using Reflexive Access Lists

Benefits of Reflexive Access Lists

Reflexive access lists are an important part of securing your network against network hackers, and can be included in a firewall defense. Reflexive access lists provide a level of security against spoofing and certain denial-of-service attacks. Reflexive access lists are simple to use, and, compared to basic access lists, provide greater control over which packets enter your network.

What Is a Reflexive Access List?

Reflexive access lists are similar in many ways to other access lists. Reflexive access lists contain condition statements (entries) that define criteria for permitting IP packets. These entries are evaluated in order, and when a match occurs, no more entries are evaluated.

However, reflexive access lists have significant differences from other types of access lists. Reflexive access lists contain only temporary entries; these entries are automatically created when a new IP session begins (for example, with an outbound packet), and the entries are removed when the session ends. Reflexive access lists are not themselves applied directly to an interface, but are "nested" within an extended named IP access list that is applied to the interface. (For more information about this, see the section "Reflexive Access Lists Configuration Task List" later in this chapter.) Also, reflexive access lists do not have the usual implicit "deny all traffic" statement at the end of the list, because of the nesting.

How Reflexive Access Lists Implement Session Filtering

This section compares session filtering with basic access lists to session filtering with reflexive access lists. This section contains the following sections:

- With Basic Access Lists
- With Reflexive Access Lists

With Basic Access Lists

With basic standard and static extended access lists, you can approximate session filtering by using the **established** keyword with the **permit** command. The **established** keyword filters TCP packets based on whether the ACK or RST bits are set. (Set ACK or RST bits indicate that the packet is not the first in the session, and therefore, that the packet belongs to an established session.) This filter criterion would be part of an access list applied permanently to an interface.

With Reflexive Access Lists

Reflexive access lists, however, provide a truer form of session filtering, which is much harder to spoof because more filter criteria must be matched before a packet is permitted through. (For example, source and destination addresses and port numbers are checked, not just ACK and RST bits.) Also, session filtering uses temporary filters which are removed when a session is over. This limits the hacker's attack opportunity to a smaller time window.

Moreover, the previous method of using the **established** keyword was available only for the TCP upper-layer protocol. So, for the other upper-layer protocols (such as UDP, ICMP, and so forth), you would have to either permit all incoming traffic or define all possible permissible source/destination host/port address pairs for each protocol. (Besides being an unmanageable task, this could exhaust NVRAM space.)

Where to Configure Reflexive Access Lists

Configure reflexive access lists on border routers—routers that pass traffic between an internal and external network. Often, these are firewall routers.

Note In this chapter, the words "within your network" and "internal network" refer to a network that is controlled (secured), such as your organization's intranet, or to a part of your organization's internal network that has higher security requirements than another part. "Outside your network" and "external network" refer to a network that is uncontrolled (unsecured) such as the Internet or to a part of your organization's network that is not as highly secured.

How Reflexive Access Lists Work

A reflexive access list is triggered when a new IP upper-layer session (such as TCP or UDP) is initiated from inside your network, with a packet traveling to the external network. When triggered, the reflexive access list generates a new, temporary entry. This entry will permit traffic to enter your network if the traffic is part of the session, but will not permit traffic to enter your network if the traffic is not part of the session.

For example, if an outbound TCP packet is forwarded to outside of your network, and this packet is the first packet of a TCP session, then a new, temporary reflexive access list entry will be created. This entry is added to the reflexive access list, which applies to inbound traffic. The temporary entry has characteristics as described next.

This section contains the following sections:

- Temporary Access List Entry Characteristics
- When the Session Ends

Temporary Access List Entry Characteristics

- The entry is always a **permit** entry.
- The entry specifies the same protocol (TCP) as the original outbound TCP packet.
- The entry specifies the same source and destination addresses as the original outbound TCP packet, except the addresses are swapped.

- The entry specifies the same source and destination port numbers as the original outbound TCP packet, except the port numbers are swapped.

  (This entry characteristic applies only for TCP and UDP packets. Other protocols, such as ICMP and IGMP, do not have port numbers, and other criteria are specified. For example, for ICMP, type numbers are used instead.)
- Inbound TCP traffic will be evaluated against the entry, until the entry expires. If an inbound TCP packet matches the entry, the inbound packet will be forwarded into your network.
- The entry will expire (be removed) after the last packet of the session passes through the interface.
- If no packets belonging to the session are detected for a configurable length of time (the timeout period), the entry will expire.

When the Session Ends

Temporary reflexive access list entries are removed at the end of the session. For TCP sessions, the entry is removed 5 seconds after two set FIN bits are detected, or immediately after matching a TCP packet with the RST bit set. (Two set FIN bits in a session indicate that the session is about to end; the 5-second window allows the session to close gracefully. A set RST bit indicates an abrupt session close.) Or, the temporary entry is removed after no packets of the session have been detected for a configurable length of time (the timeout period).

For UDP and other protocols, the end of the session is determined differently than for TCP. Because other protocols are considered to be connectionless (sessionless) services, there is no session tracking information embedded in packets. Therefore, the end of a session is considered to be when no packets of the session have been detected for a configurable length of time (the timeout period).

Restrictions on Using Reflexive Access Lists

Reflexive access lists do not work with some applications that use port numbers that change during a session. For example, if the port numbers for a return packet are different from the originating packet, the return packet will be denied, even if the packet is actually part of the same session.

The TCP application of FTP is an example of an application with changing port numbers. With reflexive access lists, if you start an FTP request from within your network, the request will not complete. Instead, you must use Passive FTP when originating requests from within your network.

Prework: Before You Configure Reflexive Access Lists

Before you configure reflexive access lists, you must decide whether to configure reflexive access lists on an internal or external interface, as described in the next section, "Choosing an Interface: Internal or External."

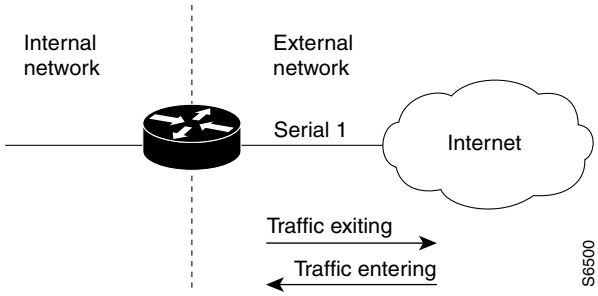
You should also be sure that you have a basic understanding of the IP protocol and of access lists; specifically, you should know how to configure extended named IP access lists. To learn about configuring IP extended access lists, refer to the "Configuring IP Services" chapter of the *Cisco IOS IP Configuration Guide*.

Choosing an Interface: Internal or External

Reflexive access lists are most commonly used with one of two basic network topologies. Determining which of these topologies is most like your own can help you decide whether to use reflexive access lists with an internal interface or with an external interface (the interface connecting to an internal network, or the interface connecting to an external network).

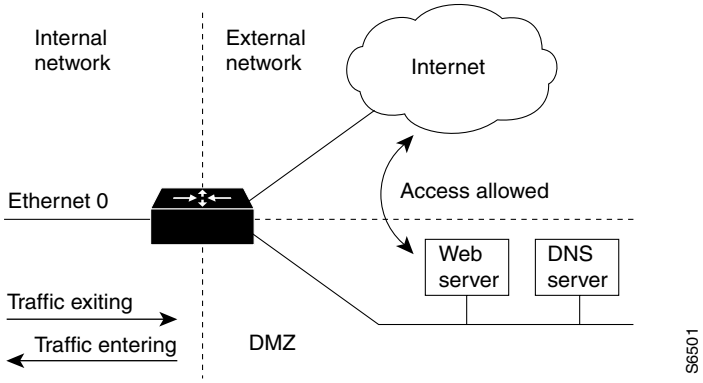
The first topology is shown in Figure 18. In this simple topology, reflexive access lists are configured for the *external* interface Serial 1. This prevents IP traffic from entering the router and the internal network, unless the traffic is part of a session already established from within the internal network.

Figure 18 Simple Topology—Reflexive Access Lists Configured at the External Interface



The second topology is shown in Figure 19. In this topology, reflexive access lists are configured for the *internal* interface Ethernet 0. This allows external traffic to access the services in the Demilitarized Zone (DMZ), such as DNS services, but prevents IP traffic from entering your internal network—unless the traffic is part of a session already established from within the internal network.

Figure 19 DMZ Topology—Reflexive Access Lists Configured at the Internal Interface



Use these two example topologies to help you decide whether to configure reflexive access lists for an internal or external interface.

Reflexive Access Lists Configuration Task List

In the previous section, "Prework: Before You Configure Reflexive Access Lists," you decided whether to configure reflexive access lists for an internal or external interface.

Now, complete the tasks in one of the following configuration task lists:

- External Interface Configuration Task List
- Internal Interface Configuration Task List

For configuration examples, refer to the "Reflexive Access List Configuration Examples" section at the end of this chapter.

External Interface Configuration Task List

To configure reflexive access lists for an external interface, perform the following tasks:

1. Defining the reflexive access list(s) in an *outbound* IP extended named access list
2. Nesting the reflexive access list(s) in an *inbound* IP extended named access list
3. Setting a global timeout value

These tasks are described in the sections following the "Internal Interface Configuration Task List" section.

Note The defined (outbound) reflexive access list evaluates traffic traveling out of your network: if the defined reflexive access list is matched, temporary entries are created in the nested (inbound) reflexive access list. These temporary entries will then be applied to traffic traveling into your network.

Internal Interface Configuration Task List

To configure reflexive access lists for an internal interface, perform the following tasks:

1. Defining the reflexive access list(s) in an *inbound* IP extended named access list
2. Nesting the reflexive access list(s) in an *outbound* IP extended named access list
3. Setting a global timeout value

These tasks are described in the next sections.

Note The defined (inbound) reflexive access list is used to evaluate traffic traveling out of your network: if the defined reflexive access list is matched, temporary entries are created in the nested (outbound) reflexive access list. These temporary entries will then be applied to traffic traveling into your network.

Defining the Reflexive Access List(s)

To define a reflexive access list, you use an entry in an extended named IP access list. This entry must use the **reflect** keyword.

- If you are configuring reflexive access lists for an external interface, the extended named IP access list should be one that is applied to outbound traffic.

- If you are configuring reflexive access lists for an internal interface, the extended named IP access list should be one that is applied to inbound traffic.

To define reflexive access lists, use the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip access-list extended` *name*	External interface: Specifies the outbound access list. or Internal interface: Specifies the inbound access list. (This command enters access-list configuration mode.)
Step 2	`Router(config-ext-nacl)# permit` *protocol* `any any reflect` *name* `[timeout` *seconds*`]`	Defines the reflexive access list using the reflexive **permit** entry. Repeat this step for each IP upper-layer protocol; for example, you can define reflexive filtering for TCP sessions and also for UDP sessions. You can use the same *name* for multiple protocols. For additional guidelines for this task, see the following section, "Mixing Reflexive Access List Statements with Other Permit and Deny Entries."

If the extended named IP access list you just specified has never been applied to the interface, you must also apply the extended named IP access list to the interface.

To apply the extended named IP access list to the interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip access-group` *name* `out` or `Router(config-if)# ip access-group` *name* `in`	External interface: Applies the extended access list to the interface's outbound traffic. Internal interface: Applies the extended access list to the interface's inbound traffic.

Mixing Reflexive Access List Statements with Other Permit and Deny Entries

The extended IP access list that contains the reflexive access list **permit** statement can also contain other normal **permit** and **deny** statements (entries). However, as with all access lists, the order of entries is important, as explained in the next few paragraphs.

If you configure reflexive access lists for an external interface, when an outbound IP packet reaches the interface, the packet will be evaluated sequentially by each entry in the outbound access list until a match occurs.

If the packet matches an entry prior to the reflexive **permit** entry, the packet will not be evaluated by the reflexive **permit** entry, and no temporary entry will be created for the reflexive access list (reflexive filtering will not be triggered).

The outbound packet will be evaluated by the reflexive **permit** entry only if no other match occurs first. Then, if the packet matches the protocol specified in the reflexive **permit** entry, the packet is forwarded out of the interface and a corresponding temporary entry is created in the inbound reflexive access list (unless the corresponding entry already exists, indicating the outbound packet belongs to a session in progress). The temporary entry specifies criteria that permits inbound traffic only for the same session.

Nesting the Reflexive Access List(s)

After you define a reflexive access list in one IP extended access list, you must "nest" the reflexive access list within a different extended named IP access list.

- If you are configuring reflexive access lists for an external interface, nest the reflexive access list within an extended named IP access list applied to inbound traffic.
- If you are configuring reflexive access lists for an internal interface, nest the reflexive access list within an extended named IP access list applied to outbound traffic.

After you nest a reflexive access list, packets heading into your internal network can be evaluated against any reflexive access list temporary entries, along with the other entries in the extended named IP access list.

To nest reflexive access lists, use the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip access-list extended` *name*	External interface: Specifies the inbound access list. or Internal interface: Specifies the outbound access list. (This command enters access-list configuration mode.)
Step 2	`Router(config-ext-nacl)# evaluate` *name*	Adds an entry that "points" to the reflexive access list. Adds an entry for each reflexive access list *name* previously defined.

Again, the order of entries is important. Normally, when a packet is evaluated against entries in an access list, the entries are evaluated in sequential order, and when a match occurs, no more entries are evaluated. With a reflexive access list nested in an extended access list, the extended access list entries are evaluated sequentially up to the nested entry, then the reflexive access list entries are evaluated sequentially, and then the remaining entries in the extended access list are evaluated sequentially. As usual, after a packet matches *any* of these entries, no more entries will be evaluated.

If the extended named IP access list you just specified has never been applied to the interface, you must also apply the extended named IP access list to the interface.

To apply the extended named IP access list to the interface, use one of the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# ip access-group name in` or `Router(config-if)# ip access-group name out`	External interface: Applies the extended access list to the interface's inbound traffic. Internal interface: Applies the extended access list to the interface's outbound traffic.

Setting a Global Timeout Value

Reflexive access list entries expire after no packets in the session have been detected for a certain length of time (the "timeout" period). You can specify the timeout for a particular reflexive access list when you define the reflexive access list. But if you do not specify the timeout for a given reflexive access list, the list will use the global timeout value instead.

The global timeout value is 300 seconds by default. But, you can change the global timeout to a different value at any time.

To change the global timeout value, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip reflexive-list timeout seconds`	Changes the global timeout value for temporary reflexive access list entries. Use a positive integer from 0 to 2,147,483.

Reflexive Access List Configuration Examples

The following sections provide reflexive access list configuration examples:

- External Interface Configuration Example
- Internal Interface Configuration Example

External Interface Configuration Example

This example shows reflexive access lists configured for an external interface, for a topology similar to the one in Figure 18 (shown earlier in this chapter).

This configuration example permits both inbound and outbound TCP traffic at interface Serial 1, but only if the first packet (in a given session) originated from inside your network. The interface Serial 1 connects to the Internet.

Define the interface where the session-filtering configuration is to be applied:

```
interface serial 1
 description Access to the Internet via this interface
```

Apply access lists to the interface, for inbound traffic and for outbound traffic:

```
ip access-group inboundfilters in
ip access-group outboundfilters out
```

Define the outbound access list. This is the access list that evaluates all outbound traffic on interface Serial 1.

```
ip access-list extended outboundfilters
```

Define the reflexive access list *tcptraffic*. This entry permits *all* outbound TCP traffic and creates a new access list named *tcptraffic*. Also, when an outbound TCP packet is the first in a new session, a corresponding temporary entry will be automatically created in the reflexive access list *tcptraffic*.

```
permit tcp any any reflect tcptraffic
```

Define the inbound access list. This is the access list that evaluates all inbound traffic on interface Serial 1.

```
ip access-list extended inboundfilters
```

Define the inbound access list entries. This example shows BGP and Enhanced IGRP running on the interface. Also, no ICMP traffic is permitted. The last entry points to the reflexive access list. If a packet does not match the first three entries, the packet will be evaluated against all the entries in the reflexive access list *tcptraffic*.

```
permit bgp any any
permit eigrp any any
deny icmp any any
evaluate tcptraffic
```

Define the global idle timeout value for all reflexive access lists. In this example, when the reflexive access list *tcptraffic* was defined, no timeout was specified, so *tcptraffic* uses the global timeout. Therefore, if for 120 seconds there is no TCP traffic that is part of an established session, the corresponding reflexive access list entry will be removed.

```
ip reflexive-list timeout 120
```

The example configuration looks as follows:

```
interface Serial 1
 description Access to the Internet via this interface
 ip access-group inboundfilters in
 ip access-group outboundfilters out
!
ip reflexive-list timeout 120
!
ip access-list extended outboundfilters
 permit tcp any any reflect tcptraffic
!
ip access-list extended inboundfilters
 permit bgp any any
 permit eigrp any any
 deny icmp any any
 evaluate tcptraffic
```

With this configuration, before any TCP sessions have been initiated the **show access-list** EXEC command displays the following:

```
Extended IP access list inboundfilters
 permit bgp any any
 permit eigrp any any
 deny icmp any any
 evaluate tcptraffic
Extended IP access list outboundfilters
 permit tcp any any reflect tcptraffic
```

Notice that the reflexive access list does not appear in this output. This is because before any TCP sessions have been initiated, no traffic has triggered the reflexive access list, and the list is empty (has no entries). When empty, reflexive access lists do not show up in **show access-list** output.

After a Telnet connection is initiated from within your network to a destination outside of your network, the **show access-list** EXEC command displays the following:

```
Extended IP access list inboundfilters
 permit bgp any any (2 matches)
 permit eigrp any any
 deny icmp any any
 evaluate tcptraffic
Extended IP access list outboundfilters
 permit tcp any any reflect tcptraffic
Reflexive IP access list tcptraffic
 permit tcp host 172.19.99.67 eq telnet host 192.168.60.185 eq 11005 (5 matches) (time
left 115 seconds)
```

Notice that the reflexive access list *tcptraffic* now appears and displays the temporary entry generated when the Telnet session initiated with an outbound packet.

Internal Interface Configuration Example

This is an example configuration for reflexive access lists configured for an internal interface. This example has a topology similar to the one in Figure 19 (shown earlier in this chapter).

This example is similar to the previous example; the only difference between this example and the previous example is that the entries for the outbound and inbound access lists are swapped. Please refer to the previous example for more details and descriptions.

```
interface Ethernet 0
 description Access from the I-net to our Internal Network via this interface
 ip access-group inboundfilters in
 ip access-group outboundfilters out
!
ip reflexive-list timeout 120
!
ip access-list extended outboundfilters
 permit bgp any any
 permit eigrp any any
 deny icmp any any
 evaluate tcptraffic
!
ip access-list extended inboundfilters
 permit tcp any any reflect tcptraffic
```

Configuring TCP Intercept (Preventing Denial-of-Service Attacks)

This chapter describes how to configure your router to protect TCP servers from TCP SYN-flooding attacks, a type of denial-of-service attack. This is accomplished by configuring the Cisco IOS feature known as TCP Intercept.

For a complete description of TCP Intercept commands, refer to the "TCP Intercept Commands" chapter of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Finding Additional Feature Support Information" section on page cxvii in the chapter Using Cisco IOS Software for Release 12.4.

In This Chapter

This chapter has the following sections:

- About TCP Intercept
- TCP Intercept Configuration Task List
- TCP Intercept Configuration Example

About TCP Intercept

The TCP intercept feature implements software to protect TCP servers from TCP SYN-flooding attacks, which are a type of denial-of-service attack.

A SYN-flooding attack occurs when a hacker floods a server with a barrage of requests for connection. Because these messages have unreachable return addresses, the connections cannot be established. The resulting volume of unresolved open connections eventually overwhelms the server and can cause it to deny service to valid requests, thereby preventing legitimate users from connecting to a web site, accessing e-mail, using FTP service, and so on.

The TCP intercept feature helps prevent SYN-flooding attacks by intercepting and validating TCP connection requests. In intercept mode, the TCP intercept software intercepts TCP synchronization (SYN) packets from clients to servers that match an extended access list. The software establishes a connection with the client on behalf of the destination server, and if successful, establishes the

connection with the server on behalf of the client and knits the two half-connections together transparently. Thus, connection attempts from unreachable hosts will never reach the server. The software continues to intercept and forward packets throughout the duration of the connection.

In the case of illegitimate requests, the software's aggressive timeouts on half-open connections and its thresholds on TCP connection requests protect destination servers while still allowing valid requests.

When establishing your security policy using TCP intercept, you can choose to intercept all requests or only those coming from specific networks or destined for specific servers. You can also configure the connection rate and threshold of outstanding connections.

You can choose to operate TCP intercept in watch mode, as opposed to intercept mode. In watch mode, the software passively watches the connection requests flowing through the router. If a connection fails to get established in a configurable interval, the software intervenes and terminates the connection attempt.

TCP options that are negotiated on handshake (such as RFC 1323 on window scaling) will not be negotiated because the TCP intercept software does not know what the server can do or will negotiate.

TCP Intercept Configuration Task List

To configure TCP intercept, perform the tasks in the following sections. The first task is required; the rest are optional.

- Enabling TCP Intercept (Required)
- Setting the TCP Intercept Mode (Optional)
- Setting the TCP Intercept Drop Mode (Optional)
- Changing the TCP Intercept Timers (Optional)
- Changing the TCP Intercept Aggressive Thresholds (Optional)
- Monitoring and Maintaining TCP Intercept (Optional)

For TCP intercept configuration examples using the commands in this chapter, refer to the "TCP Intercept Configuration Example" section at the end of this chapter.

Enabling TCP Intercept

To enable TCP intercept, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# access-list access-list-number {deny | permit} tcp any destination destination-wildcard`	Defines an IP extended access list.
Step 2	`Router(config)# ip tcp intercept list access-list-number`	Enables TCP intercept.

You can define an access list to intercept all requests or only those coming from specific networks or destined for specific servers. Typically the access list will define the source as **any** and define specific destination networks or servers. That is, you do not attempt to filter on the source addresses because you do not necessarily know who to intercept packets from. You identify the destination in order to protect destination servers.

If no access list match is found, the router allows the request to pass with no further action.

Setting the TCP Intercept Mode

The TCP intercept can operate in either active intercept mode or passive watch mode. The default is intercept mode.

In intercept mode, the software actively intercepts each incoming connection request (SYN) and responds on behalf of the server with an SYN-ACK, then waits for an ACK from the client. When that ACK is received, the original SYN is sent to the server and the software performs a three-way handshake with the server. When this is complete, the two half-connections are joined.

In watch mode, connection requests are allowed to pass through the router to the server but are watched until they become established. If they fail to become established within 30 seconds (configurable with the **ip tcp intercept watch-timeout** command), the software sends a Reset to the server to clear up its state.

To set the TCP intercept mode, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip tcp intercept mode {intercept | watch}`	Sets the TCP intercept mode.

Setting the TCP Intercept Drop Mode

When under attack, the TCP intercept feature becomes more aggressive in its protective behavior. If the number of incomplete connections exceeds 1100 or the number of connections arriving in the last one minute exceeds 1100, each new arriving connection causes the oldest partial connection to be deleted. Also, the initial retransmission timeout is reduced by half to 0.5 seconds (so the total time trying to establish a connection is cut in half).

By default, the software drops the oldest partial connection. Alternatively, you can configure the software to drop a random connection. To set the drop mode, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip tcp intercept drop-mode {oldest | random}`	Sets the drop mode.

Changing the TCP Intercept Timers

By default, the software waits for 30 seconds for a watched connection to reach established state before sending a Reset to the server. To change this value, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip tcp intercept watch-timeout seconds`	Changes the time allowed to reach established state.

By default, the software waits for 5 seconds from receipt of a reset or FIN-exchange before it ceases to manage the connection. To change this value, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip tcp intercept finrst-timeout seconds`	Changes the time between receipt of a reset or FIN-exchange and dropping the connection.

By default, the software still manages a connection for 24 hours after no activity. To change this value, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip tcp intercept connection-timeout seconds`	Changes the time the software will manage a connection after no activity.

Changing the TCP Intercept Aggressive Thresholds

Two factors determine when aggressive behavior begins and ends: total incomplete connections and connection requests during the last one-minute sample period. Both thresholds have default values that can be redefined.

When a threshold is exceeded, the TCP intercept assumes the server is under attack and goes into aggressive mode. When in aggressive mode, the following occurs:

- Each new arriving connection causes the oldest partial connection to be deleted. (You can change to a random drop mode.)
- The initial retransmission timeout is reduced by half to 0.5 seconds, and so the total time trying to establish the connection is cut in half. (When not in aggressive mode, the code does exponential back-off on its retransmissions of SYN segments. The initial retransmission timeout is 1 second. The subsequent timeouts are 2 seconds, 4 seconds, 8 seconds, and 16 seconds. The code retransmits 4 times before giving up, so it gives up after 31 seconds of no acknowledgment.)
- If in watch mode, the watch timeout is reduced by half. (If the default is in place, the watch timeout becomes 15 seconds.)

The drop strategy can be changed from the oldest connection to a random connection with the **ip tcp intercept drop-mode** command.

Note The two factors that determine aggressive behavior are related and work together. When *either* of the **high** values is exceeded, aggressive behavior begins. When *both* quantities fall below the **low** value, aggressive behavior ends.

You can change the threshold for triggering aggressive mode based on the total number of incomplete connections. The default values for **low** and **high** are 900 and 1100 incomplete connections, respectively. To change these values, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip tcp intercept max-incomplete low` *number*	Sets the threshold for stopping aggressive mode.
Step 2	`Router(config)# ip tcp intercept max-incomplete high` *number*	Sets the threshold for triggering aggressive mode.

You can also change the threshold for triggering aggressive mode based on the number of connection requests received in the last 1-minute sample period. The default values for **low** and **high** are 900 and 1100 connection requests, respectively. To change these values, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip tcp intercept one-minute low` *number*	Sets the threshold for stopping aggressive mode.
Step 2	`Router(config)# ip tcp intercept one-minute high` *number*	Sets the threshold for triggering aggressive mode.

Monitoring and Maintaining TCP Intercept

To display TCP intercept information, use either of the following commands in EXEC mode:

Command	Purpose
`Router# show tcp intercept connections`	Displays incomplete connections and established connections.
`Router# show tcp intercept statistics`	Displays TCP intercept statistics.

TCP Intercept Configuration Example

The following configuration defines extended IP access list 101, causing the software to intercept packets for all TCP servers on the 192.168.1.0/24 subnet:

```
ip tcp intercept list 101
!
access-list 101 permit tcp any 192.168.1.0 0.0.0.255
```

Context-Based Access Control

This part consists of the following:

- Configuring Context-Based Access Control
- Cisco IOS Firewall Performance Improvements
- E-mail Inspection Engine
- ESMTP Support for Cisco IOS Firewall
- Firewall ACL Bypass
- Firewall N2H2 Support
- Firewall Stateful Inspection of ICMP
- Firewall Support for SIP
- Firewall Websense URL Filtering
- Firewall Support of Skinny Client Control Protocol (SCCP)
- Granular Protocol Inspection
- HTTP Inspection Engine
- Inspection of Router-Generated Traffic
- Transparent Cisco IOS Firewall
- Virtual Fragmentation Reassembly
- VRF Aware Cisco IOS Firewall

Configuring Context-Based Access Control

This chapter describes how to configure Context-based Access Control (CBAC). CBAC provides advanced traffic filtering functionality and can be used as an integral part of your network's firewall. For more information regarding firewalls, refer to the chapter "Cisco IOS Firewall Overview."

For a complete description of the CBAC commands used in this chapter, refer to the "Context-Based Access Control Commands" chapter in the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Finding Additional Feature Support Information" section on page cxvii in the chapter Using Cisco IOS Software for Release 12.4.

In This Chapter

This chapter has the following sections:

- About Context-Based Access Control
- CBAC Configuration Task List
- Monitoring and Maintaining CBAC
- CBAC Configuration Examples

About Context-Based Access Control

This section describes CBAC features and functions:

- What CBAC Does
- What CBAC Does Not Do
- How CBAC Works
- When and Where to Configure CBAC
- The CBAC Process
- Supported Protocols
- Restrictions
- Memory and Performance Impact

What CBAC Does

CBAC works to provide network protection on multiple levels using the following functions:

- Traffic Filtering
- Traffic Inspection
- Alerts and Audit Trails
- Intrusion Prevention

Traffic Filtering

CBAC intelligently filters TCP and UDP packets based on application-layer protocol session information. You can configure CBAC to permit specified TCP and UDP traffic through a firewall only when the connection is initiated from within the network you want to protect. CBAC can inspect traffic for sessions that originate from either side of the firewall, and CBAC can be used for intranet, extranet, and Internet perimeters of your network.

Without CBAC, traffic filtering is limited to access list implementations that examine packets at the network layer, or at most, the transport layer. However, CBAC examines not only network layer and transport layer information but also examines the application-layer protocol information (such as FTP connection information) to learn about the state of the session. This allows support of protocols that involve multiple channels created as a result of negotiations in the control channel. Most of the multimedia protocols as well as some other protocols (such as FTP, RPC, and SQL*Net) involve multiple channels.

Using CBAC, Java blocking can be configured to filter HTTP traffic based on the server address or to completely deny access to Java applets that are not embedded in an archived or compressed file. With Java, you must protect against the risk of users inadvertently downloading destructive applets into your network. To protect against this risk, you could require all users to disable Java in their browser. If this is not an acceptable solution, you can create a CBAC inspection rule to filter Java applets at the firewall, which allows users to download only applets residing within the firewall and trusted applets from outside the firewall. For extensive content filtering of Java, Active-X, or virus scanning, you might want to consider purchasing a dedicated content filtering product.

Traffic Inspection

CBAC inspects traffic that travels through the firewall to discover and manage state information for TCP and UDP sessions. This state information is used to create temporary openings in the firewall's access lists to allow return traffic and additional data connections for permissible sessions.

Inspecting packets at the application layer, and maintaining TCP and UDP session information, provides CBAC with the ability to detect and prevent certain types of network attacks such as SYN-flooding. A SYN-flood attack occurs when a network attacker floods a server with a barrage of requests for connection and does not complete the connection. The resulting volume of half-open connections can overwhelm the server, causing it to deny service to valid requests. Network attacks that deny access to a network device are called denial-of-service (DoS) attacks.

CBAC helps to protect against DoS attacks in other ways. CBAC inspects packet sequence numbers in TCP connections to see if they are within expected ranges—CBAC drops any suspicious packets. You can also configure CBAC to drop half-open connections, which require firewall processing and memory resources to maintain. Additionally, CBAC can detect unusually high rates of new connections and issue alert messages.

CBAC can help by protecting against certain DoS attacks involving fragmented IP packets. Even though the firewall prevents an attacker from making actual connections to a given host, the attacker can disrupt services provided by that host. This is done by sending many non-initial IP fragments or by sending complete fragmented packets through a router with an ACL that filters the first fragment of a fragmented packet. These fragments can tie up resources on the target host as it tries to reassemble the incomplete packets.

Alerts and Audit Trails

CBAC also generates real-time alerts and audit trails. Enhanced audit trail features use SYSLOG to track all network transactions; recording time stamps, source host, destination host, ports used, and the total number of transmitted bytes, for advanced, session-based reporting. Real-time alerts send SYSLOG error messages to central management consoles upon detecting suspicious activity. Using CBAC inspection rules, you can configure alerts and audit trail information on a per-application protocol basis. For example, if you want to generate audit trail information for HTTP traffic, you can specify that in the CBAC rule covering HTTP inspection.

Intrusion Prevention

CBAC provides a limited amount of intrusion detection to protect against specific SMTP attacks. With intrusion detection, SYSLOG messages are reviewed and monitored for specific "attack signatures." Certain types of network attacks have specific characteristics, or signatures. When CBAC detects an attacks, it resets the offending connections and sends SYSLOG information to the SYSLOG server. Refer to the section "Interpreting Syslog and Console Messages Generated by CBAC" later in this chapter for a list of supported signatures.

In addition to the limited intrusion detection offered by CBAC, the Cisco IOS Firewall feature set offers intrusion detection technology for mid-range and high-end router platforms using the Cisco IOS Intrusion Prevention System (IPS). Cisco IOS IPS restructures and replaces the existing Cisco IOS Intrusion Detection System (IDS). It is ideal for any network perimeter, and especially for locations in which a router is being deployed and additional security between network segments is required. It also can protect intranet and extranet connections where additional security is mandated, and branch-office sites connecting to the corporate office or Internet.

The Cisco IOS IPS acts as an in-line intrusion detection sensor, watching packets and sessions as they flow through the router and scanning each packet to match any of the Cisco IOS IPS signatures. When Cisco IOS IPS detects suspicious activity, it responds before network security can be compromised and logs the event through Cisco IOS syslog messages or Security Device Event Exchange (SDEE).

For more information about Cisco IOS IPS, refer to the module "Configuring Cisco IOS Intrusion Prevention System (IPS)."

What CBAC Does Not Do

CBAC does not provide intelligent filtering for all protocols; it only works for the protocols that you specify. If you do not specify a certain protocol for CBAC, the existing access lists will determine how that protocol is filtered. No temporary openings will be created for protocols not specified for CBAC inspection.

CBAC does not protect against attacks originating from within the protected network unless that traffic travels through a router that has the Cisco IOS Firewall feature set deployed on it. CBAC only detects and protects against attacks that travel through the firewall. This is a scenario in which you might want to deploy CBAC on an intranet-based router.

CBAC protects against certain types of attacks, but not every type of attack. CBAC should not be considered a perfect, impenetrable defense. Determined, skilled attackers might be able to launch effective attacks. While there is no such thing as a perfect defense, CBAC detects and prevents most of the popular attacks on your network.

How CBAC Works

You should understand the material in this section before you configure CBAC. If you do not understand how CBAC works, you might inadvertently introduce security risks by configuring CBAC inappropriately. This section contains the following sections:

- How CBAC Works—Overview
- How CBAC Works—Details

How CBAC Works—Overview

CBAC creates temporary openings in access lists at firewall interfaces. These openings are created when specified traffic exits your internal network through the firewall. The openings allow returning traffic (that would normally be blocked) and additional data channels to enter your internal network back through the firewall. The traffic is allowed back through the firewall only if it is part of the same session as the original traffic that triggered CBAC when exiting through the firewall.

Throughout this chapter, the terms “inbound” and “outbound” are used to describe the direction of traffic relative to the router interface on which CBAC is applied. For example, if a CBAC rule is applied inbound on interface E0, then packets entering interface E0 from the network will be inspected. If a CBAC rule is applied outbound on interface E0, then packets leaving interface E0 to the network will be inspected. This is similar to the way ACLs work.

For example, consider a CBAC inspection rule named *hqusers*, and suppose that rule is applied inbound at interface E0:

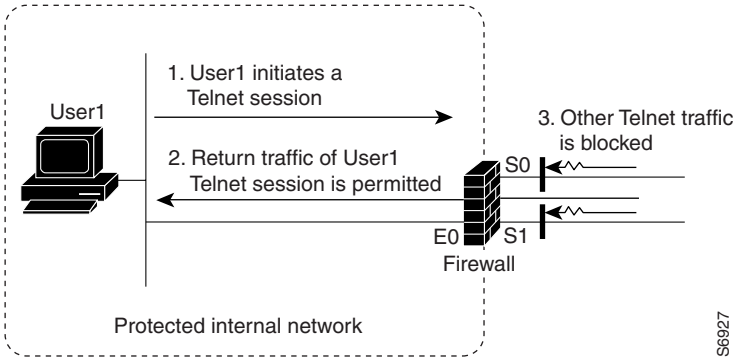
```
router (config-if)# ip inspect hqusers in
```

This command causes CBAC to inspect the packets coming into this interface from the network. If a packet is attempting to initiate a session, CBAC will then determine if this protocol is allowed, create a CBAC session, add the appropriate ACLs to allow return traffic and do any needed content inspection on any future packets for this session.

The terms “input” and “output” are used to describe the interfaces at which network traffic enters or exits the firewall router. A packet enters the firewall router via the input interface, is inspected by the firewall software and then exits the router via the output interface.

In Figure 20, the inbound access lists at S0 and S1 are configured to block Telnet traffic, and there is no outbound access list configured at E0. When the connection request for User1's Telnet session passes through the firewall, CBAC creates a temporary opening in the inbound access list at S0 to permit returning Telnet traffic for User1's Telnet session. (If the same access list is applied to both S0 and S1, the same opening would appear at both interfaces.) If necessary, CBAC would also have created a similar opening in an outbound access list at E0 to permit return traffic.

Figure 20 CBAC Opens Temporary Holes in Firewall Access Lists



How CBAC Works—Details

This section describes how CBAC inspects packets and maintains state information about sessions to provide intelligent filtering.

Packets Are Inspected

With CBAC, you specify which protocols you want to be inspected, and you specify an interface and interface direction (in or out) where inspection originates. Only specified protocols will be inspected by CBAC.

Packets entering the firewall are inspected by CBAC only if they first pass the inbound access list at the input interface and outbound access list at the output interface. If a packet is denied by the access list, the packet is simply dropped and not inspected by CBAC.

CBAC inspection tracks sequence numbers in all TCP packets, and drops those packets with sequence numbers that are not within expected ranges.

CBAC inspection recognizes application-specific commands (such as illegal SMTP commands) in the control channel, and detects and prevents certain application-level attacks.

When CBAC suspects an attack, the DoS feature can take several actions:

- Generate alert messages
- Protect system resources that could impede performance
- Block packets from suspected attackers

CBAC uses timeout and threshold values to manage session state information, helping to determine when to drop sessions that do not become fully established. Setting timeout values for network sessions helps prevent DoS attacks by freeing up system resources, dropping sessions after a specified amount of time. Setting threshold values for network sessions helps prevent DoS attacks by controlling the number of half-open sessions, which limits the amount of system resources applied to half-open sessions. When a

session is dropped, CBAC sends a reset message to the devices at both end points (source and destination) of the session. When the system under DoS attack receives a reset command, it releases, or frees up, processes and resources related to that incomplete session.

CBAC provides three thresholds against DoS attacks:

- The total number of half-open TCP or UDP sessions
- The number of half-open sessions based upon time
- The number of half-open TCP-only sessions per host

If a threshold is exceeded, CBAC has two options:

- Send a reset message to the end points of the oldest half-open session, making resources available to service newly arriving SYN packets.
- In the case of half open TCP only sessions, CBAC blocks all SYN packets temporarily for the duration configured by the threshold value. When the router blocks a SYN packet, the TCP three-way handshake is never initiated, which prevents the router from using memory and processing resources needed for valid connections.

DoS detection and prevention requires that you create a CBAC inspection rule and apply that rule on an interface. The inspection rule must include the protocols that you want to monitor against DoS attacks. For example, if you have TCP inspection enabled on the inspection rule, then CBAC can track all TCP connections to watch for DoS attacks. If the inspection rule includes FTP protocol inspection but not TCP inspection, CBAC tracks only FTP connections for DoS attacks.

For detailed information about setting timeout and threshold values in CBAC to detect and prevent DoS attacks, refer in the "Configuring Global Timeouts and Thresholds" section.

A State Table Maintains Session State Information

Whenever a packet is inspected, a state table is updated to include information about the state of the session.

Return traffic will only be permitted back through the firewall if the state table contains information indicating that the packet belongs to a permissible session. CBAC controls the traffic that belongs to a valid session. When return traffic is inspected, the state table information is updated as necessary.

UDP "Sessions" Are Approximated

With UDP—a connectionless service—there are no actual sessions, so the software approximates sessions by examining the information in the packet and determining if the packet is similar to other UDP packets (for example, same source/destination addresses and port numbers) and if the packet was detected soon after another similar UDP packet. "Soon" means within the configurable UDP idle timeout period.

Access List Entries Are Dynamically Created and Deleted to Permit Return Traffic and Additional Data Connections

CBAC dynamically creates and deletes access list entries at the firewall interfaces, according to the information maintained in the state tables. These access list entries are applied to the interfaces to examine traffic flowing back into the internal network. These entries create temporary openings in the firewall to permit only traffic that is part of a permissible session.

The temporary access list entries are never saved to NVRAM.

When and Where to Configure CBAC

Configure CBAC at firewalls protecting internal networks. Such firewalls should be Cisco routers with the Cisco IOS Firewall feature set configured as described previously in the section "Cisco IOS Firewall."

Use CBAC when the firewall will be passing traffic such as the following:

- Standard TCP and UDP Internet applications
- Multimedia applications
- Oracle support

Use CBAC for these applications if you want the application's traffic to be permitted through the firewall only when the traffic session is initiated from a particular side of the firewall (usually from the protected internal network).

In many cases, you will configure CBAC in one direction only at a single interface, which causes traffic to be permitted back into the internal network only if the traffic is part of a permissible (valid, existing) session. This is a typical configuration for protecting your internal networks from traffic that originates on the Internet.

You can also configure CBAC in two directions at one or more interfaces. CBAC is configured in two directions when the networks on both sides of the firewall should be protected, such as with extranet or intranet configurations, and to protect against DoS attacks. For example, if the firewall is situated between two partner companies' networks, you might wish to restrict traffic in one direction for certain applications, and restrict traffic in the opposite direction for other applications.

The CBAC Process

This section describes a sample sequence of events that occurs when CBAC is configured at an external interface that connects to an external network such as the Internet.

In this example, a TCP packet exits the internal network through the firewall's external interface. The TCP packet is the first packet of a Telnet session, and TCP is configured for CBAC inspection.

1. The packet reaches the firewall's external interface.
2. The packet is evaluated against the interface's existing outbound access list, and the packet is permitted. (A denied packet would simply be dropped at this point.)
3. The packet is inspected by CBAC to determine and record information about the state of the packet's connection. This information is recorded in a new state table entry created for the new connection.

   (If the packet's application—Telnet—was not configured for CBAC inspection, the packet would simply be forwarded out the interface at this point without being inspected by CBAC. See the section "Defining an Inspection Rule" later in this chapter for information about configuring CBAC inspection.)
4. Based on the obtained state information, CBAC creates a temporary access list entry which is inserted at the beginning of the external interface's inbound extended access list. This temporary access list entry is designed to permit inbound packets that are part of the same connection as the outbound packet just inspected.
5. The outbound packet is forwarded out the interface.
6. Later, an inbound packet reaches the interface. This packet is part of the same Telnet connection previously established with the outbound packet. The inbound packet is evaluated against the inbound access list, and it is permitted because of the temporary access list entry previously created.

7. The permitted inbound packet is inspected by CBAC, and the connection's state table entry is updated as necessary. Based on the updated state information, the inbound extended access list temporary entries might be modified in order to permit only packets that are valid for the current state of the connection.
8. Any additional inbound or outbound packets that belong to the connection are inspected to update the state table entry and to modify the temporary inbound access list entries as required, and they are forwarded through the interface.
9. When the connection terminates or times out, the connection's state table entry is deleted, and the connection's temporary inbound access list entries are deleted.

In the sample process just described, the firewall access lists are configured as follows:

- An outbound IP access list (standard or extended) is applied to the external interface. This access list permits all packets that you want to allow to exit the network, including packets you want to be inspected by CBAC. In this case, Telnet packets are permitted.
- An inbound extended IP access list is applied to the external interface. This access list denies any traffic to be inspected by CBAC—including Telnet packets. When CBAC is triggered with an outbound packet, CBAC creates a temporary opening in the inbound access list to permit only traffic that is part of a valid, existing session.

  If the inbound access list had been configured to permit *all* traffic, CBAC would be creating pointless openings in the firewall for packets that would be permitted anyway.

Supported Protocols

This section provides a list of CBAC supported protocols and includes a more detailed look at support for multimedia applications, specifically RTSP and H.323.

CBAC Supported Protocols

You can configure CBAC to inspect the following types of sessions:

- All TCP sessions, regardless of the application-layer protocol (sometimes called "single-channel" or "generic" TCP inspection)
- All UDP sessions, regardless of the application-layer protocol (sometimes called "single-channel" or "generic" UDP inspection)

You can also configure CBAC to specifically inspect certain application-layer protocols. The following application-layer protocols can all be configured for CBAC:

- CU-SeeMe (only the White Pine version)
- FTP
- H.323 (such as NetMeeting, ProShare)
- HTTP (Java blocking)
- Microsoft NetShow
- UNIX R-commands (such as rlogin, rexec, and rsh)
- RealAudio
- RTSP (Real Time Streaming Protocol)
- RPC (Sun RPC, not DCE RPC)

- SMTP (Simple Mail Transport Protocol)

Note CBAC can be configured to inspect SMTP but not ESMTP (Extended Simple Mail Transport Protocol). SMTP is described in RFC 821. CBAC SMTP inspect does not inspect the ESMTP session or command sequence. Configuring SMTP inspection is not useful for ESMTP, and it can cause problems.

To determine whether a mail-server is doing SMTP or ESMTP, contact your mail-server software vendor, or telnet to the mail-server port 25 and observe the banner to see if it reports SMTP or ESMTP.

- SQL*Net
- StreamWorks
- TFTP
- VDOLive

When a protocol is configured for CBAC, that protocol traffic is inspected, state information is maintained, and in general, packets are allowed back through the firewall only if they belong to a permissible session.

RTSP and H.323 Protocol Support for Multimedia Applications

CBAC supports a number of protocols for multimedia applications that require delivery of data with real-time properties such as audio and video conferencing. This support includes the following multimedia application protocols:

- Real Time Streaming Protocol (RTSP)
- H.323 Version 2 (H.323 V2)

RTSP and H.323 V2 inspection allows clients on a protected network to receive data associated with a multimedia session from a server on an unprotected network.

RTSP Support

RTSP is the IETF standards-based protocol (RFC 2326) for control over the delivery of data with real-time properties such as audio and video streams. It is useful for large-scale broadcasts and audio or video on demand streaming, and is supported by a variety of vendor products of streaming audio and video multimedia, including Cisco IP/TV, RealNetworks RealAudio G2 Player, and Apple QuickTime 4 software.

RFC 2326 allows RTSP to run over either UDP or TCP, though CBAC currently supports only TCP-based RTSP. RTSP establishes a TCP-based control connection, or channel, between the multimedia client and server. RTSP uses this channel to control commands such as “play” and “pause” between the client and server. These control commands and responses are text-based and are similar to HTTP.

RTSP typically relies on a UDP-based data transport protocol such as standard Real-Time Transport Protocol (RTP) to open separate channels for data and for RTP Control Protocol (RTCP) messages. RTP and RTCP channels occur in pairs, with RTP being an even numbered port and RTCP being the next consecutive port. Understanding the relationship of RTP and RTCP is important for verifying session information using CBAC **show** commands.

The RTSP client uses TCP port 554 or 8554 to open a multimedia connection with a server. The data channel or data control channel (using RTCP) between the client and the server is dynamically negotiated between the client and the server using any of the high UDP ports (1024 to 65536).

CBAC uses this port information along with connection information from the client to create dynamic access control list (ACL) entries in the firewall. As TCP or UDP connections are terminated, CBAC removes these dynamic entries from the appropriate ACLs.

CBAC support for RTSP includes the following data transport modes:

- Standard Real-Time Transport Protocol (RTP)

  RTP is an IETF standard (RFC 1889) supporting delivery of real-time data such as audio and video. RTP uses the RTP Control Protocol (RTCP) for managing the delivery of the multimedia data stream. This is the normal mode of operation for Cisco IP/TV and Apple QuickTime 4 software.

- RealNetworks Real Data Transport (RDT)

  RDT is a proprietary protocol developed by RealNetworks for data transport. This mode uses RTSP for communication control and uses RDT for the data connection and retransmission of lost packets. This is the normal mode of operation for the RealServer G2 from RealNetworks.

- Interleaved (Tunnel Mode)

  In this mode, RTSP uses the control channel to tunnel RTP or RDT traffic.

- Synchronized Multimedia Integration Language (SMIL)

  SMIL is a layout language that enables the creation of multimedia presentations consisting of multiple elements of music, voice, images, text, video and graphics. This involves multiple RTSP control and data streams between the player and the servers. This mode is available only using RTSP and RDT. SMIL is a proposed specification of the World Wide Web Consortium (W3C). The RealNetworks RealServer and RealServer G2 provide support for SMIL—Cisco IP/TV and Apple QuickTime 4 do not.

H.323 Support

CBAC support for H.323 inspection includes H.323 Version 2 and H.323 Version 1. H.323 V2 provides additional options over H.323 V1, including a “fast start” option. The fast start option minimizes the delay between the time that a user initiates a connection and the time that the user gets the data (voice, video). H.323 V2 inspection is backward compatible with H.323 V1.

With H.323 V1, after a TCP connection is established between the client and server (H.225 Channel), a separate channel for media control (H.245 Channel) is opened through which multimedia channels for audit and video are further negotiated.

The H.323 V2 client opens a connection to server which is listening on port 1720. The data channel between the client and the server is dynamically negotiated using any of the high UDP ports (1024 to 65536).

CBAC uses this port information along with connection information from the client to create dynamic access control list (ACL) entries in the firewall. As TCP or UDP connections are terminated, CBAC removes these dynamic entries from the appropriate ACLs.

Restrictions

CBAC has the following restrictions:

- CBAC is available only for IP protocol traffic. Only TCP and UDP packets are inspected. (Other IP traffic, such as ICMP, cannot be inspected with CBAC and should be filtered with basic access lists instead.)
- If you reconfigure your access lists when you configure CBAC, be aware that if your access lists block TFTP traffic into an interface, you will not be able to netboot over that interface. (This is not a CBAC-specific limitation, but is part of existing access list functionality.)
- Packets with the firewall as the source or destination address are not inspected by CBAC.
- CBAC ignores ICMP Unreachable messages.
- H.323 V2 and RTSP protocol inspection supports only the following multimedia client-server applications: Cisco IP/TV, RealNetworks RealAudio G2 Player, Apple QuickTime 4.

You can use CBAC together with all the other firewall features mentioned previously in the "Cisco IOS Firewall Overview" chapter.

CBAC works with fast switching and process switching.

This section also discusses restrictions concerning:

- FTP Traffic and CBAC
- IPSec and CBAC Compatibility

FTP Traffic and CBAC

- With FTP, CBAC does not allow third-party connections (three-way FTP transfer).
- When CBAC inspects FTP traffic, it only allows data channels with the destination port in the range of 1024 to 65535.
- CBAC will not open a data channel if the FTP client-server authentication fails.

IPSec and CBAC Compatibility

When CBAC and IPSec are enabled on the same router, and the firewall router is an endpoint for IPSec for the particular flow, then IPSec is compatible with CBAC (that is, CBAC can do its normal inspection processing on the flow).

If the router is not an IPSec endpoint, but the packet is an IPSec packet, then CBAC will not inspect the packets because the protocol number in the IP header of the IPSec packet is not TCP or UDP. CBAC only inspects UDP and TCP packets.

Memory and Performance Impact

CBAC uses less than approximately 600 bytes of memory per connection. Because of the memory usage, you should use CBAC only when you need to. There is also a slight amount of additional processing that occurs whenever packets are inspected.

Sometimes CBAC must evaluate long access lists, which might have presented a negative impact to performance. However, this impact is avoided, because CBAC evaluates access lists using an accelerated method (CBAC hashes access lists and evaluates the hash).

CBAC Configuration Task List

To configure CBAC, perform the tasks described in the following sections. The tasks in the first seven sections are required; the task of verifying the CBAC configuration is optional.

- Picking an Interface: Internal or External (Required)
- Configuring IP Access Lists at the Interface (Required)
- Configuring Global Timeouts and Thresholds (Required)
- Defining an Inspection Rule (Required)
- Applying the Inspection Rule to an Interface (Required)
- Configuring Logging and Audit Trail (Required)
- Other Guidelines for Configuring a Firewall (Required)
- Verifying CBAC (Optional)

Following CBAC configuration, you can monitor and maintain CBAC using the information in this section.

Note If you try to configure Context-based Access Control (CBAC) but do not have a good understanding of how CBAC works, you might inadvertently introduce security risks to the firewall and to the protected network. You should be sure you understand what CBAC does before you configure CBAC.

Note As with all networking devices, protect access into the firewall by configuring passwords as described in the "Configuring Passwords and Privileges" chapter. You should also consider configuring user authentication, authorization, and accounting as described in the "Authentication, Authorization, and Accounting (AAA)" part of this guide. Additional guidelines to help you establish a good security policy can be found in the "Cisco IOS Firewall Overview" chapter.

For CBAC configuration examples, refer to the "CBAC Configuration Examples" section at the end of this chapter.

Picking an Interface: Internal or External

You must decide whether to configure CBAC on an internal or external interface of your firewall.

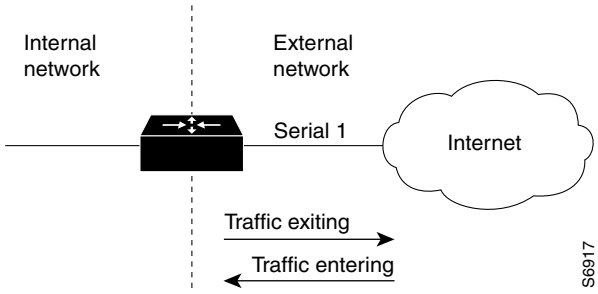
"Internal" refers to the side where sessions must originate for their traffic to be permitted through the firewall. "External" refers to the side where sessions cannot originate (sessions originating from the external side will be blocked).

If you will be configuring CBAC in two directions, you should configure CBAC in one direction first, using the appropriate "internal" and "external" interface designations. When you configure CBAC in the other direction, the interface designations will be swapped. (CBAC can be configured in two directions at one or more interfaces. Configure CBAC in two directions when the networks on both sides of the firewall require protection, such as with extranet or intranet configurations, and for protection against DoS attacks.)

The firewall is most commonly used with one of two basic network topologies. Determining which of these topologies is most like your own can help you decide whether to configure CBAC on an internal interface or on an external interface.

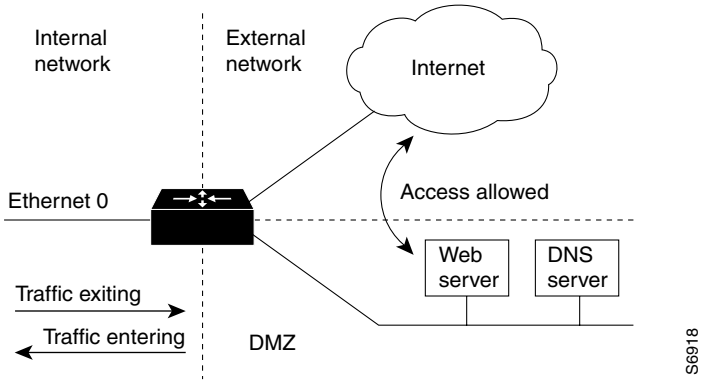
The first topology is shown in Figure 21. In this simple topology, CBAC is configured for the *external* interface Serial 1. This prevents specified protocol traffic from entering the firewall and the internal network, unless the traffic is part of a session initiated from within the internal network.

Figure 21 Simple Topology—CBAC Configured at the External Interface



The second topology is shown in Figure 22. In this topology, CBAC is configured for the *internal* interface Ethernet 0. This allows external traffic to access the services in the Demilitarized Zone (DMZ), such as DNS services, but prevents specified protocol traffic from entering your internal network—unless the traffic is part of a session initiated from within the internal network.

Figure 22 DMZ Topology—CBAC Configured at the Internal Interface



Using these two sample topologies, decide whether to configure CBAC on an internal or external interface.

To view various firewall configuration scenarios, see the "CBAC Configuration Examples" section at the end of this chapter.

Configuring IP Access Lists at the Interface

For CBAC to work properly, you need to make sure that you have IP access lists configured appropriately at the interface.

Follow these three general rules when evaluating your IP access lists at the firewall:

- Start with a basic configuration.

  If you try to configure access lists without a good understanding of how access lists work, you might inadvertently introduce security risks to the firewall and to the protected network. You should be sure you understand what access lists do before you configure your firewall. For more information about access control lists, refer to the "Access Control Lists: Overview and Guidelines" chapter.

  A basic initial configuration allows all network traffic to flow from the protected networks to the unprotected networks, while blocking network traffic from any unprotected networks.

- Permit CBAC traffic to leave the network through the firewall.

  All access lists that evaluate traffic leaving the protected network should permit traffic that will be inspected by CBAC. For example, if Telnet will be inspected by CBAC, then Telnet traffic should be permitted on all access lists that apply to traffic leaving the network.

- Use extended access lists to deny CBAC return traffic entering the network through the firewall.

  For temporary openings to be created in an access list, the access list must be an extended access list. So wherever you have access lists that will be applied to returning traffic, you must use extended access lists. The access lists should deny CBAC return traffic because CBAC will open up temporary holes in the access lists. (You want traffic to be normally blocked when it enters your network.)

Note If your firewall only has two connections, one to the internal network and one to the external network, using all inbound access lists works well because packets are stopped before they get a chance to affect the router itself.

This section contains the following sections:

- Basic Configuration
- External Interface
- Internal Interface

Basic Configuration

The first time you configure the Cisco IOS Firewall, it is helpful to start with a basic access list configuration that makes the operation of the firewall easy to understand without compromising security. The basic configuration allows all network traffic from the protected networks access to the unprotected networks, while blocking all network traffic (with some exceptions) from the unprotected networks to the protected networks.

Any firewall configuration depends on your site security policy. If the basic configuration does not meet your initial site security requirements, configure the firewall to meet your policy. If you are unfamiliar with that policy or need help with the configuration, contact your network administration group for assistance. For additional guidelines on configuring a firewall, refer to the "Other Guidelines for Configuring a Firewall" section in this chapter.

Use the following guidelines for configuring the initial firewall access lists:

- Do not configure an access list for traffic from the protected networks to the unprotected networks, meaning that all traffic from the protected networks can flow through the interface.

  This helps to simplify firewall management by reducing the number of access lists applied at the interfaces. Of course this assumes a high level of trust for the users on the protected networks, and it assumes there are no malicious users on the protected networks who might launch attacks from the "inside." You can fine tune network access for users on the protected networks as you gain experience with access list configuration and the operation of the firewall.

- Configure an access list that includes entries permitting certain ICMP traffic from unprotected networks.

  While an access list that denies all IP traffic not part of a connection inspected by CBAC seems most secure, it is not practical for normal operation of the router. The router expects to see ICMP traffic from other routers in the network. Additionally, ICMP traffic is not inspected by CBAC, meaning specific entries are needed in the access list to permit return traffic for ICMP commands. For example, a user on a protected network uses the **ping** command to get the status of a host on an unprotected network; without entries in the access list that permit **echo reply** messages, the user on the protected network gets no response to the **ping** command.

  Include access list entries to permit the following ICMP messages:

Message	Description
echo reply	Outgoing ping commands require echo-reply messages to come back.
time-exceeded	Outgoing traceroute commands require time-exceeded messages to come back.
packet-too-big	Path MTU discovery requires "too-big" messages to come back.
traceroute	Allow an incoming traceroute.
unreachable	Permit all "unreachable" messages to come back. If a router cannot forward or deliver a datagram, it sends an ICMP unreachable message back to the source and drops the datagram.

- Add an access list entry denying any network traffic from a source address matching an address on the protected network.

  This is known as anti-spoofing protection because it prevents traffic from an unprotected network from assuming the identity of a device on the protected network.

- Add an entry denying broadcast messages with a source address of 255.255.255.255.

  This entry helps to prevent broadcast attacks.

- By default, the last entry in an extended access list is an implicit denial of all IP traffic not specifically allowed by other entries in the access list.

  Although this is the default setting, this final deny statement is not shown by default in an access list. Optionally, you can add an entry to the access list denying IP traffic with any source or destination address with no undesired effects.

For complete information about how to configure IP access lists, refer to the "Configuring IP Services" chapter of the *Cisco IOS IP Addressing Services Configuration Guide*.

For tips on applying access lists at an external or internal interface, review the sections "External Interface" and "Internal Interface" in this chapter.

External Interface

Here are some guidelines for your access lists when you will be configuring CBAC on an external interface:

- If you have an outbound IP access list at the external interface, the access list can be a standard or extended access list. This outbound access list should permit traffic that you want to be inspected by CBAC. If traffic is not permitted, it will not be inspected by CBAC, but will be simply dropped.
- The inbound IP access list at the external interface must be an extended access list. This inbound access list should deny traffic that you want to be inspected by CBAC. (CBAC will create temporary openings in this inbound access list as appropriate to permit only return traffic that is part of a valid, existing session.)
- For complete information about how to configure IP access lists, refer to the "Configuring IP Services" chapter of the *Cisco IOS IP Addressing Services Configuration Guide*.

Internal Interface

Here are some tips for your access lists when you will be configuring CBAC on an internal interface:

- If you have an inbound IP access list at the internal interface or an outbound IP access list at external interface(s), these access lists can be either a standard or extended access list. These access lists should permit traffic that you want to be inspected by CBAC. If traffic is not permitted, it will not be inspected by CBAC, but will be simply dropped.
- The outbound IP access list at the internal interface and the inbound IP access list at the external interface must be extended access lists. These outbound access lists should deny traffic that you want to be inspected by CBAC. (CBAC will create temporary openings in these outbound access lists as appropriate to permit only return traffic that is part of a valid, existing session.) You do not necessarily need to configure an extended access list at both the outbound internal interface and the inbound external interface, but at least one is necessary to restrict traffic flowing through the firewall into the internal protected network.
- For complete information about how to configure IP access lists, refer to the "Configuring IP Services" chapter of the *Cisco IOS IP Addressing Services Configuration Guide*.

Configuring Global Timeouts and Thresholds

CBAC uses timeouts and thresholds to determine how long to manage state information for a session, and to determine when to drop sessions that do not become fully established. These timeouts and thresholds apply globally to all sessions.

You can use the default timeout and threshold values, or you can change to values more suitable to your security requirements. You should make any changes to the timeout and threshold values before you continue configuring CBAC.

Note If you want to enable the more aggressive TCP host-specific denial-of-service prevention that includes the blocking of connection initiation to a host, you must set the **block-time** specified in the **ip inspect tcp max-incomplete host** command (see the last row in Table 24).

All the available CBAC timeouts and thresholds are listed in Table 24, along with the corresponding command and default value. To change a global timeout or threshold listed in the "Timeout of Threshold Value to Change" column, use the global configuration command in the "Command" column:

Table 24 Timeout and Threshold Values

Timeout or Threshold Value to Change	Command	Default
The length of time the software waits for a TCP session to reach the established state before dropping the session.	**ip inspect tcp synwait-time** *seconds*	30 seconds
The length of time a TCP session will still be managed after the firewall detects a FIN-exchange.	**ip inspect tcp finwait-time** *seconds*	5 seconds
The length of time a TCP session will still be managed after no activity (the TCP idle timeout).[1]	**ip inspect tcp idle-time** *seconds*	3600 seconds (1 hour)
The length of time a UDP session will still be managed after no activity (the UDP idle timeout).[1]	**ip inspect udp idle-time** *seconds*	30 seconds
The length of time a DNS name lookup session will still be managed after no activity.	**ip inspect dns-timeout** *seconds*	5 seconds
The number of existing half-open sessions that will cause the software to start deleting half-open sessions.[2]	**ip inspect max-incomplete high** *number*	500 existing half-open sessions
The number of existing half-open sessions that will cause the software to stop deleting half-open sessions.[2]	**ip inspect max-incomplete low** *number*	400 existing half-open sessions
The rate of new sessions that will cause the software to start deleting half-open sessions.[2]	**ip inspect one-minute high** *number*	500 half-open sessions per minute
The rate of new sessions that will cause the software to stop deleting half-open sessions.[2]	**ip inspect one-minute low** *number*	400 half-open sessions per minute
The number of existing half-open TCP sessions with the same destination host address that will cause the software to start dropping half-open sessions to the same destination host address.[3]	**ip inspect tcp max-incomplete host** *number* **block-time** *minutes*	50 existing half-open TCP sessions; 0 minutes

1. The global TCP and UDP idle timeouts can be overridden for specified application-layer protocols' sessions as described in the **ip inspect name** (global configuration) command description, found in the "Context-Based Access Control Commands" chapter of the *Cisco IOS Security Command Reference*.
2. See the following section, "Half-Open Sessions," for more information.
3. Whenever the **max-incomplete host** threshold is exceeded, the software will drop half-open sessions differently depending on whether the **block-time** timeout is zero or a positive non-zero number. If the **block-time** timeout is zero, the software will delete the oldest existing half-open session for the host for every new connection request to the host and will let the SYN packet through. If the **block-time** timeout is greater than zero, the software will delete all existing half-open sessions for the host, and then block all new connection requests to the host. The software will continue to block all new connection requests until the **block-time** expires.

To reset any threshold or timeout to the default value, use the **no** form of the command in Table 24.

Half-Open Sessions

An unusually high number of half-open sessions (either absolute or measured as the arrival rate) could indicate that a denial-of-service attack is occurring. For TCP, “half-open” means that the session has not reached the established state—the TCP three-way handshake has not yet been completed. For UDP, “half-open” means that the firewall has detected no return traffic.

CBAC measures both the total number of existing half-open sessions and the rate of session establishment attempts. Both TCP and UDP half-open sessions are counted in the total number and rate measurements. Rate measurements are made several times per minute.

When the number of existing half-open sessions rises above a threshold (the **max-incomplete high** number), the software will delete half-open sessions as required to accommodate new connection requests. The software will continue to delete half-open requests as necessary, until the number of existing half-open sessions drops below another threshold (the **max-incomplete low** number).

When the rate of new connection attempts rises above a threshold (the **one-minute high** number), the software will delete half-open sessions as required to accommodate new connection attempts. The software will continue to delete half-open sessions as necessary, until the rate of new connection attempts drops below another threshold (the **one-minute low** number). The rate thresholds are measured as the number of new session connection attempts detected in the last one-minute sample period.

Defining an Inspection Rule

After you configure global timeouts and thresholds, you must define an inspection rule. This rule specifies what IP traffic (which application-layer protocols) will be inspected by CBAC at an interface.

Normally, you define only one inspection rule. The only exception might occur if you want to enable CBAC in two directions as described earlier in the section “When and Where to Configure CBAC.” For CBAC configured in both directions at a single firewall interface, you should configure two rules, one for each direction.

An inspection rule should specify each desired application-layer protocol as well as generic TCP or generic UDP if desired. The inspection rule consists of a series of statements each listing a protocol and specifying the same inspection rule name.

Inspection rules include options for controlling alert and audit trail messages and for checking IP packet fragmentation.

To define an inspection rule, follow the instructions in the following sections:

- Configuring Application-Layer Protocol Inspection
- Configuring Generic TCP and UDP Inspection

Configuring Application-Layer Protocol Inspection

This section provides instructions for configuring CBAC with the following inspection information:

- Configuring Application-Layer Protocols
- Configuring Java Blocking
- Configuring IP Packet Fragmentation Inspection

Note For CBAC inspection to work with NetMeeting 2.0 traffic (an H.323 application-layer protocol), you must also configure inspection for TCP, as described later in the "Configuring Generic TCP and UDP Inspection" section. This requirement exists because NetMeeting 2.0 uses an additional TCP channel not defined in the H.323 specification.

Configuring Application-Layer Protocols

To configure CBAC inspection for an application-layer protocol, use one or both of the following commands in global configuration mode:

Command	Purpose
`Router(config)# ip inspect name` *inspection-name protocol* `[alert {on | off}] [audit-trail {on | off}] [timeout` *seconds*`]`	Configures CBAC inspection for an application-layer protocol (except for RPC and Java). Use one of the protocol keywords defined in Table 25. Repeat this command for each desired protocol. Use the same *inspection-name* value to create a single inspection rule.
`Router(config)# ip inspect name` *inspection-name* `rpc program-number` *number* `[wait-time` *minutes*`] [alert {on | off}] [audit-trail {on | off}] [timeout` *seconds*`]`	Enables CBAC inspection for the RPC application-layer protocol. You can specify multiple RPC program numbers by repeating this command for each program number. Use the same *inspection-name* value to create a single inspection rule.

Refer to the description of the **ip inspect name** global configuration command in the "Context-Based Access Control Commands" chapter of the *Cisco IOS Security Command Reference* for more information about how the command works with each application-layer protocol.

To enable CBAC inspection for Java blocking, see the following section, "Configuring Java Blocking."

Table 25 identifies application protocol keywords for the **ip inspect name** command.

Table 25 Application Protocol Keywords for the ip inspect name Command

Application Protocol	Protocol Keyword
CU-SeeMe	**cuseeme**
FTP	**ftp**
H.323	**h323**
Microsoft NetShow	**netshow**
UNIX R commands (rlogin, rexec, rsh)	**rcmd**
RealAudio	**realaudio**
SMTP	**smtp**
SQL*Net	**sqlnet**
StreamWorks	**streamworks**
TFTP	**tftp**
VDOLive	**vdolive**

Configuring Java Blocking

Java applet filtering distinguishes between trusted and untrusted applets by relying on a list of external sites that you designate as "friendly." If an applet is from a friendly site, the firewall allows the applet through. If the applet is not from a friendly site, the applet will be blocked. (Alternately, you could permit applets from all external sites except for those you specifically designate as hostile.)

Note Java blocking forces a strict order on TCP packets. To properly verify that Java applets are not in the response, a firewall will drop any TCP packet that is out of order. Because the network—not the firewall—determines how packets are routed, the firewall cannot control the order of the packets; the firewall can only drop and retransmit all TCP packets that are not in order.

To block all Java applets except for applets from friendly locations, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip access-list standard` *name* `  permit ...` `  deny ...` (Use permit and deny statements as appropriate.) or `Router(config)# access-list` *access-list-number* {**deny** | **permit**} *protocol source* [*source-wildcard*]eq www *destination* [*destination-wildcard*]	Creates a standard access list that permits traffic only from friendly sites, and denies traffic from hostile sites. Use the **any** keyword for the destination as appropriate—but be careful to not misuse the **any** keyword to inadvertently allow all applets through.
Step 2	`Router(config)# ip inspect name inspection-name http` [**java-list** *access-list*] [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout** *seconds*]	Blocks all Java applets except for applets from the friendly sites defined previously in the access list. Java blocking only works with numbered standard access lists. To create a single inspection rule, use the same *inspection-name* value as when you specified other protocols.

Caution CBAC does not detect or block encapsulated Java applets. Therefore, Java applets that are wrapped or encapsulated, such as applets in .zip or .jar format, are *not* blocked at the firewall. CBAC also does not detect or block applets loaded from FTP, gopher, HTTP on a nonstandard port, and so forth.

Configuring IP Packet Fragmentation Inspection

CBAC inspection rules can help protect hosts against certain DoS attacks involving fragmented IP packets.

Using fragmentation inspection, the firewall maintains an *interfragment state* (structure) for IP traffic. Non-initial fragments are discarded unless the corresponding initial fragment was permitted to pass through the firewall. Non-initial fragments received before the corresponding initial fragments are discarded.

Note Fragmentation inspection can have undesirable effects in certain cases, because it can result in the firewall discarding any packet whose fragments arrive out of order. There are many circumstances that can cause out-of-order delivery of legitimate fragments. Applying fragmentation inspection in situations where legitimate fragments, which are likely to arrive out of order, might have a severe performance impact.

Because routers running Cisco IOS software are used in a large variety of networks, and because the CBAC feature is often used to isolate parts of internal networks from one another, the fragmentation inspection feature is disabled by default. Fragmentation detection must be explicitly enabled for an inspection rule using the **ip inspect name** command. Unfragmented traffic is never discarded because it lacks a fragment state. Even when the system is under heavy attack with fragmented packets, legitimate fragmented traffic, if any, gets some fraction of the firewall's fragment state resources, and legitimate, unfragmented traffic can flow through the firewall unimpeded.

Configuring Generic TCP and UDP Inspection

You can configure TCP and UDP inspection to permit TCP and UDP packets to enter the internal network through the firewall, even if the application-layer protocol is not configured to be inspected. However, TCP and UDP inspection do not recognize application-specific commands, and therefore might not permit all return packets for an application, particularly if the return packets have a different port number than the previous exiting packet.

Any application-layer protocol that is inspected will take precedence over the TCP or UDP packet inspection. For example, if inspection is configured for FTP, all control channel information will be recorded in the state table, and all FTP traffic will be permitted back through the firewall if the control channel information is valid for the state of the FTP session. The fact that TCP inspection is configured is irrelevant to the FTP state information.

With TCP and UDP inspection, packets entering the network must exactly match the corresponding packet that previously exited the network. The entering packets must have the same source/destination addresses and source/destination port numbers as the exiting packet (but reversed); otherwise, the entering packets will be blocked at the interface. Also, all TCP packets with a sequence number outside of the window are dropped.

With UDP inspection configured, replies will only be permitted back in through the firewall if they are received within a configurable time after the last request was sent out. (This time is configured with the **ip inspect udp idle-time** command.)

To configure CBAC inspection for TCP or UDP packets, use one or both of the following commands in global configuration mode:

Command	Purpose
`Router(config)# ip inspect name` *inspection-name* **tcp** [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout** *seconds*]	Enables CBAC inspection for TCP packets. To create a single inspection rule, use the same *inspection-name* value as when you specified other protocols.
`Router(config)# ip inspect name` *inspection-name* **udp** [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout** *seconds*]	Enables CBAC inspection for UDP packets. To create a single inspection rule, use the same *inspection-name* value as when you specified other protocols.

Applying the Inspection Rule to an Interface

After you define an inspection rule, you apply this rule to an interface.

Normally, you apply only one inspection rule to one interface. The only exception might occur if you want to enable CBAC in two directions as described earlier in the section "When and Where to Configure CBAC." For CBAC configured in both directions at a single firewall interface, you should apply two rules, one for each direction.

If you are configuring CBAC on an external interface, apply the rule to outbound traffic.

If you are configuring CBAC on an internal interface, apply the rule to inbound traffic.

To apply an inspection rule to an interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip inspect` *inspection-name* `{in | out}`	Applies an inspection rule to an interface.

Configuring Logging and Audit Trail

Turn on logging and audit trail to provide a record of network access through the firewall, including illegitimate access attempts, and inbound and outbound services. To configure logging and audit trail functions, enter the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# service timestamps log datetime`	Adds the date and time to syslog and audit trail messages.
Step 2	`Router(config)# logging` *host*	Specifies the host name or IP address of the host where you want to send syslog messages.
Step 3	`Router(config)# logging facility` *facility-type*	Configures the syslog facility in which error messages are sent.
Step 4	`Router(config)# logging trap` *level*	(Optional) Uses this command to limit messages logged to the syslog servers based on severity. The default is level 7 (informational).
Step 5	`Router(config)# ip inspect audit-trail`	Turns on CBAC audit trail messages.

For information on how to interpret the syslog and audit trail messages, refer to the "Interpreting Syslog and Console Messages Generated by CBAC" section.

To configure audit trail functions on a per-application basis, refer to the "Defining an Inspection Rule" section for more information.

For complete information about how to configure logging, refer to the "Troubleshooting the Router" chapter of the *Cisco IOS Network Management Configuration Guide*.

Other Guidelines for Configuring a Firewall

As with all networking devices, you should always protect access into the firewall by configuring passwords as described in the module "Configuring Security with Passwords, Privilege Levels, and Login Usernames for CLI Sessions on Networking Devices." You should also consider configuring user authentication, authorization, and accounting as described in the "Authentication, Authorization, and Accounting (AAA)" part of this guide.

You should also consider the following recommendations:

- When setting passwords for privileged access to the firewall, use the **enable secret** command rather than the **enable password** command, which does not have as strong an encryption algorithm.
- Put a password on the console port. In authentication, authorization, and accounting (AAA) environments, use the same authentication for the console as for elsewhere. In a non-AAA environment, at a minimum configure the **login** and **password** *password* commands.
- Think about access control *before* you connect a console port to the network in any way, including attaching a modem to the port. Be aware that a *break* on the console port might give total control of the firewall, even with access control configured.
- Apply access lists and password protection to all virtual terminal ports. Use access lists to limit who can Telnet into your router.
- Do not enable any local service (such as SNMP or NTP) that you do not use. Cisco Discovery Protocol (CDP) and Network Time Protocol (NTP) are on by default, and you should turn these off if you do not need them.

  To turn off CDP, enter the **no cdp run** global configuration command. To turn off NTP, enter the **ntp disable** interface configuration command on each interface not using NTP.

  If you must run NTP, configure NTP only on required interfaces, and configure NTP to listen only to certain peers.

  Any enabled service could present a potential security risk. A determined, hostile party might be able to find creative ways to misuse the enabled services to access the firewall or the network.

  For local services that are enabled, protect against misuse. Protect by configuring the services to communicate only with specific peers, and protect by configuring access lists to deny packets for the services at specific interfaces.
- Protect against spoofing: protect the networks on both sides of the firewall from being spoofed from the other side. You could protect against spoofing by configuring input access lists at all interfaces to pass only traffic from expected source addresses, and to deny all other traffic.

  You should also disable source routing. For IP, enter the **no ip source-route** global configuration command. Disabling source routing at *all* routers can also help prevent spoofing.

  You should also disable minor services. For IP, enter the **no service tcp-small-servers** and **no service udp-small-servers** global configuration commands. In Cisco IOS Release 12.0 and later, these services are disabled by default.
- Prevent the firewall from being used as a relay by configuring access lists on any asynchronous Telnet ports.
- Normally, you should disable directed broadcasts for all applicable protocols on your firewall and on all your other routers. For IP, use the **no ip directed-broadcast** command. Rarely, some IP networks do require directed broadcasts; if this is the case, do not disable directed broadcasts.

  Directed broadcasts can be misused to multiply the power of denial-of-service attacks, because every denial-of-service packet sent is broadcast to every host on a subnet. Furthermore, some hosts have other intrinsic security risks present when handling broadcasts.

- Configure the **no proxy-arp** command to prevent internal addresses from being revealed. (This is important to do if you do not already have NAT configured to prevent internal addresses from being revealed.)
- Keep the firewall in a secured (locked) room.

Verifying CBAC

You can view and verify CBAC configuration, status, statistics, and session information by using one or more of the following commands in EXEC mode:

Command	Purpose
`Router# show ip access-lists`	Displays the contents of all current IP access lists.
`Router# show ip inspect name` *inspection-name*	Shows a particular configured inspection rule.
`Router# show ip inspect config`	Shows the complete CBAC inspection configuration.
`Router# show ip inspect interfaces`	Shows interface configuration with regards to applied inspection rules and access lists.
`Router# show ip inspect session [detail]`	Shows existing sessions that are currently being tracked and inspected by CBAC.
`Router# show ip inspect all`	Shows all CBAC configuration and all existing sessions that are currently being tracked and inspected by CBAC.

In most cases, you can tell whether CBAC is inspecting network traffic properly because network applications are working as expected. In some cases, however, you might want to verify CBAC operation. For example, to verify RTSP or H.323 inspection, initiate an RTSP- or H.323-based application through the firewall. Use the **show ip inspect session** and **show ip access lists** commands to verify CBAC operation. These commands display the dynamic ACL entries and the established connections for a multimedia session.

In the case of RTSP inspection, session output can vary based on the multimedia protocol and the transport mode. This section uses examples of RTSP and H.323 V2 sessions to illustrate verification procedures and to illustrate how session information, and the interpretation of that session information, varies based on the protocol being inspected. This section provides the following sample session output:

- RTSP with RDT
- RTSP with TCP Only (Interleaved Mode)
- RTSP with SMIL
- RTSP with RTP (IP/TV)
- H.323 V2

RTSP with RDT

The following example illustrates the result of the **show ip inspect session** command. It shows that a control channel (rtsp) and data channel (rtsp-data) are open between hosts 192.168.155.2 and 192.168.35.1.

```
router# show ip inspect session
Established Sessions
 Session 616B4F1C (192.168.155.2:7548)=>(192.168.35.1:6970) rtsp-data SIS_OPEN
 Session 611E2904 (192.168.35.1:1221)=>(192.168.155.2:554) rtsp SIS_OPEN
```

The following example illustrates the result of the **show ip access-list** command. It shows that two dynamic entries (permit statements) were added to ACL 100 for the multimedia session. The TCP entry creates a dynamic opening through the firewall between port 554 (RTSP protocol port) on the client and port 1221 on the server. The UDP entry creates a dynamic opening between data port 7548 on the client and data port 6970 on the server.

```
router# show ip access-list
Extended IP access list 100
 permit udp host 192.168.155.2 eq 7548 host 192.168.35.1 eq 6970 (31 matches)
 permit tcp host 192.168.155.2 eq 554 host 192.168.35.1 eq 1221 (27 matches)
```

After closing the multimedia session, review the session output using the **show** commands to verify the firewall software has removed the dynamic entries from the configuration.

RTSP with TCP Only (Interleaved Mode)

The following example illustrates the result of the **show ip inspect session** command. It shows that only a single control channel (rtsp) is open between hosts 192.168.155.2 and 192.168.35.1. In this mode, data is tunneled through the firewall using the TCP connection to interleave RDT or RTP data.

```
router# show ip inspect session
Established Sessions
 Session 611E2904 (192.168.35.1:1228)=>(192.168.155.2:554) rtsp SIS_OPEN
```

The following example illustrates the result of the **show ip access-list** command. It shows that a single dynamic entry (permit statement) was added to ACL 100 for the multimedia session. The TCP entry creates a dynamic opening through the firewall between port 554 (RTSP protocol port) on the client and port 1228 on the server.

```
router# show ip access-lists
Extended IP access list 100
 permit tcp host 192.168.155.2 eq 554 host 192.168.35.1 eq 1228 (391 matches)
```

After closing the multimedia session, review the session output using the **show** commands to verify the firewall software has removed the dynamic entries from the configuration.

RTSP with SMIL

The following example illustrates the result of the **show ip inspect session** command for RTSP using Synchronized Multimedia Integration Language (SMIL). It shows that a single control channel (rtsp) and multiple data channels (rtsp-data) are open between hosts 192.168.155.2 and 192.168.35.1. The data channels appear as half open sessions because the UDP data flows in one direction only, which is from the server to the client.

```
router# show ip inspect session
Established Sessions
 Session 616CA914 (192.168.155.2:30616)=>(192.168.35.1:6974) rtsp-data SIS_OPEN
 Session 616B4E78 (192.168.35.1:1230)=>(192.168.155.2:554) rtsp SIS_OPEN
 Session 614AB61C (192.168.155.2:29704)=>(192.168.35.1:6976) rtsp-data SIS_OPEN
 Session 616CAA88 (192.168.155.2:26764)=>(192.168.35.1:6972) rtsp-data SIS_OPEN
Half-open Sessions
 Session 614AAEF0 (192.168.155.2:15520)=>(192.168.35.1:6970) rtsp-data SIS_OPENING
```

The following example illustrates the result of the **show ip access-lists** command. It shows that multiple dynamic entries (permit statements) were added to ACL 100 for the multimedia session. The TCP entry creates a dynamic opening through the firewall between port 554 (RTSP protocol port) on the client and port 1230 on the server. The UDP entries create dynamic openings between negotiated data ports on the client (192.168.155.2) and the server (192.168.35.1).

```
router# show ip access-list
Extended IP access list 100
 permit udp host 192.168.155.2 eq 29704 host 192.168.35.1 eq 6976 (182 matches)
 permit udp host 192.168.155.2 eq 30616 host 192.168.35.1 eq 6974 (268 matches)
 permit udp host 192.168.155.2 eq 26764 host 192.168.35.1 eq 6972 (4 matches)
 permit udp host 192.168.155.2 eq 15520 host 192.168.35.1 eq 6970 (12 matches)
 permit tcp host 192.168.155.2 eq 554 host 192.168.35.1 eq 1230 (41 matches)
```

After closing the multimedia session, review the session output using the **show** commands to verify the firewall software has removed the dynamic entries from the configuration.

RTSP with RTP (IP/TV)

The following example illustrates the result of the **show ip inspect session** command for RTSP with the Cisco IP/TV application. The output shows that a single control channel (rtsp) and multiple data channels (rtsp-data) are open between hosts 192.168.2.15 and 192.168.102.23. The data channels appear as half-open sessions because the UDP data flows in one direction only, which is from the server to the client.

```
router# show ip inspect session
Established Sessions
 Session 611493C0 (192.168.2.15:2571)=>(192.168.102.23:8554) rtsp SIS_OPEN
Half-open Sessions
 Session 6114A22C (192.168.102.23:2428)=>(192.168.2.15:20112) rtsp-data SIS_OPENING
 Session 61149F44 (192.168.102.23:2428)=>(192.168.2.15:20113) rtsp-data SIS_OPENING
 Session 6114A0B8 (192.168.102.23:2429)=>(192.168.2.15:20115) rtsp-data SIS_OPENING
 Session 6114A3A0 (192.168.102.23:2429)=>(192.168.2.15:20114) rtsp-data SIS_OPENING
```

The following example illustrates the result of the **show ip access-lists** command. It shows that multiple dynamic entries (permit statements) were added to ACL 100 for the multimedia session. The TCP entry creates a dynamic opening through the firewall between port 554 (RTSP protocol port) on the client and port 1230 on the server. The UDP entries create dynamic openings between negotiated data ports on the client (192.168.2.15) and the server (192.168.102.23).

```
router# show ip access-lists
Extended IP access list 100
 permit udp host 192.168.102.23 eq 2428 host 192.168.2.15 eq 20113 (11 matches)
 permit udp host 192.168.102.23 eq 2428 host 192.168.2.15 eq 20112 (256 matches)
 permit udp host 192.168.102.23 eq 2429 host 192.168.2.15 eq 20115 (11 matches)
 permit udp host 192.168.102.23 eq 2429 host 192.168.2.15 eq 20114 (4598 matches)
 permit tcp host 192.168.102.23 eq 8554 host 192.168.2.15 eq 2571 (22 matches)
```

After closing the multimedia session, review the session output using the **show** commands to verify that the firewall software has removed the dynamic entries from the configuration.

H.323 V2

The following example illustrates the result of the **show ip inspect session** command for H.323 V2. It shows a single H.323 control channel, an RTP Control Protocol channel for both audio and video data, and an RTP data channel between hosts 192.168.155.2 and 192.168.35.1.

```
Session 615E2688 (192.168.35.1:49609)=>(192.168.155.1:49609) H323-RTCP-audio SIS_OPEN
Session 615E2688 (192.168.35.1:49508)=>(192.168.155.1:49508) H323-RTP-audio SIS_OPEN
Session 615E2688 (192.168.35.1:49410)=>(192.168.155.1:49410) H323-RTP-video SIS_OPEN
Session 615E2688 (192.168.35.1:49611)=>(192.168.155.1:49611) H323-RTCP-video SIS_OPEN
Session 615E1640 (192.168.35.1:4414)=>(192.168.155.1:1720) H323 SIS_OPEN
```

The following example illustrates the result of the **show ip access-lists** command. It shows that multiple dynamic entries (permit statements) were added to ACL 100 for the multimedia session. The TCP entry creates a dynamic opening through the firewall between port 1720 (H.323 V2 protocol port) on the client and port 4414 on the server. The UDP entries create dynamic openings between negotiated data ports on the client (192.168.155.1) and the server (192.168.35.1).

```
router# show ip access-lists
Extended IP access list 100
 permit udp host 192.168.155.1 eq 49609 host 192.168.35.1 eq 49609 (11 matches)
 permit udp host 192.168.155.1 eq 49508 host 192.168.35.1 eq 49508 (256 matches)
 permit udp host 192.168.155.1 eq 49411 host 192.168.35.1 eq 49411 (11 matches)
 permit udp host 192.168.155.1 eq 49610 host 192.168.35.1 eq 49610 (4598 matches)
 permit tcp host 192.168.155.1 eq 1720 host 192.168.35.1 eq 4414 (22 matches)
```

Monitoring and Maintaining CBAC

You can watch for network attacks and investigate network problems using debug commands and system messages. This section has the following sections:

- Debugging Context-Based Access Control
- Interpreting Syslog and Console Messages Generated by CBAC
- Turning Off CBAC

Debugging Context-Based Access Control

To assist CBAC debugging, you can turn on audit trail messages that will be displayed on the console after each CBAC session closes. Audit trail information is also configurable on a per-application basis using the CBAC inspection rules.

To turn on audit trail messages, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip inspect audit-trail`	Turns on CBAC audit trail messages.

If required, you can also use the CBAC **debug** commands listed in this section. (Debugging can be turned off for each of the commands in this section by using the **no** form of the command. To disable all debugging, use the privileged EXEC commands **no debug all** or **undebug all**.)

The following **debug** commands are available:

- Generic Debug Commands
- Transport Level Debug Commands
- Application Protocol Debug Commands

For a complete description of the debug commands, refer to the *Cisco IOS Debug Command Reference*.

Generic Debug Commands

You can use the following generic **debug** commands, entered in privileged EXEC mode:

Command	Purpose
`Router# debug ip inspect function-trace`	Displays messages about software functions called by CBAC.
`Router# debug ip inspect object-creation`	Displays messages about software objects being created by CBAC. Object creation corresponds to the beginning of CBAC-inspected sessions.
`Router# debug ip inspect object-deletion`	Displays messages about software objects being deleted by CBAC. Object deletion corresponds to the closing of CBAC-inspected sessions.
`Router# debug ip inspect events`	Displays messages about CBAC software events, including information about CBAC packet processing.
`Router# debug ip inspect timers`	Displays messages about CBAC timer events such as when a CBAC idle timeout is reached.
`Router# debug ip inspect detail`	Enables the detailed option, which can be used in combination with other options to get additional information.

Transport Level Debug Commands

You can use the following transport-level **debug** commands, entered in privileged EXEC mode:

Command	Purpose
`Router# debug ip inspect tcp`	Displays messages about CBAC-inspected TCP events, including details about TCP packets.
`Router# debug ip inspect udp`	Displays messages about CBAC-inspected UDP events, including details about UDP packets.

Application Protocol Debug Commands

You can use the following application protocol **debug** command, entered in privileged EXEC mode:

Command	Purpose
`Router# debug ip inspect` *protocol*	Displays messages about CBAC-inspected protocol events, including details about the protocol's packets. Refer to Table 26 to determine the protocol keyword.

Table 26 identifies application protocol keywords for the **debug ip inspect** command.

Table 26 Application Protocol Keywords for the debug ip inspect Command

Application Protocol	Protocol Keyword
CU-SeeMe	**cuseeme**
FTP commands and responses	**ftp-cmd**
FTP token (enables tracing of the FTP tokens parsed)	**ftp-token**
H.323	**h323**
HTTP (Java applets)	**http**
Microsoft NetShow	**netshow**
UNIX R commands (rlogin, rexec, rsh)	**rcmd**
RealAudio	**realaudio**
RPC	**rpc**
SMTP	**smtp**
SQL*Net	**sqlnet**
StreamWorks	**streamworks**
TFTP	**tftp**
VDOLive	**vdolive**

Interpreting Syslog and Console Messages Generated by CBAC

CBAC provides syslog messages, console alert messages, and audit trail messages. These messages are useful because they can alert you to network attacks and because they provide an audit trail that provides details about sessions inspected by CBAC. While they are generally referred to as error messages, not all error messages indicate problems with your system.

Audit trail and alert information is configurable on a per-application basis using the CBAC inspection rules.

The following types of messages can be generated by CBAC:

- Denial-of-Service Attack Detection Error Messages
- SMTP Attack Detection Error Messages
- Java Blocking Error Messages
- FTP Error Messages
- Audit Trail Messages

For explanations and recommended actions related to the error messages mentioned in this section, refer to the *Cisco IOS System Error Messages*.

Denial-of-Service Attack Detection Error Messages

CBAC detects and blocks denial-of-service attacks and notifies you when denial-of-service attacks occur. Error messages such as the following may indicate that denial-of-service attacks have occurred:

```
%FW-4-ALERT_ON: getting aggressive, count (550/500) current 1-min rate: 250
%FW-4-ALERT_OFF: calming down, count (0/400) current 1-min rate: 0
```

When `%FW-4-ALERT_ON and %FW-4-ALERT_OFF` error messages appear together, each "aggressive/calming" pair of messages indicates a separate attack. The preceding example shows one separate attack.

Error messages such as the following may indicate that a denial-of-service attack has occurred on a specific TCP host:

```
%FW-4-HOST_TCP_ALERT_ON: Max tcp half-open connections (50) exceeded for host
172.21.127.242.
%FW-4-BLOCK_HOST: Blocking new TCP connections to host 172.21.127.242 for 2 minutes
(half-open count 50 exceeded)
%FW-4-UNBLOCK_HOST: New TCP connections to host 172.21.127.242 no longer blocked
```

SMTP Attack Detection Error Messages

CBAC detects and blocks SMTP attacks (illegal SMTP commands) and notifies you when SMTP attacks occur. Error messages such as the following may indicate that an SMTP attack has occurred:

```
%FW-4-SMTP_INVALID_COMMAND: Invalid SMTP command from initiator (192.168.12.3:52419)
```

CBAC also detects a limited number of SMTP attack signatures. A signature in a SYSLOG message indicates a possible attack against the protected network, such as the detection of illegal SMTP commands in a packet. Whenever a signature is detected, the connection will be reset.

The Cisco IOS Firewall supports the following SMTP attack signatures:

Signature	Description
Mail: bad rcpt	Triggers on any mail message with a “pipe” (|) symbol in the recipient field.
Mail: bad from	Triggers on any mail message with a “pipe” (|) symbol in the “From:” field.
Mail: old attack	Triggers when “wiz” or “debug” commands are sent to the SMTP port.
Mail: decode	Triggers on any mail message with a “:decode@” in the header.
Majordomo	A bug in the Majordomo program will allow remote users to execute arbitrary commands at the privilege level of the server.

The following is a sample SMTP attack signature message:

```
02:04:55: %FW-4-TCP_MAJORDOMO_EXEC_BUG: Sig:3107:Majordomo Execute Attack - from
192.168.25.1 to 192.168.205.1:
```

Java Blocking Error Messages

CBAC detects and selectively blocks Java applets and notifies you when a Java applet has been blocked. Error messages such as the following may indicate that a Java applet has been blocked:

```
%FW-4-HTTP_JAVA_BLOCK: JAVA applet is blocked from (172.21.127.218:80) to
(172.16.57.30:44673).
```

FTP Error Messages

CBAC detects and prevents certain FTP attacks and notifies you when this occurs. Error messages such as the following may appear when CBAC detects these FTP attacks:

```
%FW-3-FTP_PRIV_PORT: Privileged port 1000 used in PORT command -- FTP client 10.0.0.1  FTP
server 10.1.0.1
%FW-3-FTP_SESSION_NOT_AUTHENTICATED: Command issued before the session is authenticated
-- FTP client 10.0.0.1
%FW-3-FTP_NON_MATCHING_IP_ADDR: Non-matching address 172.19.148.154 used in PORT command
-- FTP client 172.19.54.143  FTP server 172.16.127.242
```

Audit Trail Messages

CBAC provides audit trail messages to record details about inspected sessions. Audit trail information is configurable on a per-application basis using the CBAC inspection rules. To determine which protocol was inspected, use the responder’s port number. The port number follows the responder’s address. The following are sample audit trail messages:

```
%FW-6-SESS_AUDIT_TRAIL: tcp session initiator (192.168.1.13:33192) sent 22 bytes --
responder (192.168.129.11:25) sent 208 bytes
%FW-6-SESS_AUDIT_TRAIL: http session initiator (172.16.57.30:44673) sent 1599 bytes --
responder (172.21.127.218:80) sent 93124 bytes
```

Turning Off CBAC

You can turn off CBAC using the **no ip inspect** global configuration command.

Note The **no ip inspect** command removes all CBAC configuration entries and resets all CBAC global timeouts and thresholds to the defaults. All existing sessions are deleted and their associated access lists removed.

In most situations, turning off CBAC has no negative security impact because CBAC creates "permit" access lists. Without CBAC configured, no "permit" access lists are maintained. Therefore, no derived traffic (returning traffic or traffic from the data channels) can go through the firewall. The exception is SMTP and Java blocking. With CBAC turned off, unacceptable SMTP commands or Java applets may go through the firewall.

CBAC Configuration Examples

The following sections provide CBAC configuration examples:

- Ethernet Interface Configuration Example
- ATM Interface Configuration Example
- Remote Office to ISP Configuration Example
- Remote Office to Branch Office Configuration Example
- Two-Interface Branch Office Configuration Example
- Multiple-Interface Branch Office Configuration Example

The first example develops a CBAC inspection rule for specific protocols and a supporting access control list (ACL). This example focuses how to configure CBAC; it does not provide a complete router configuration and does not describe other elements of the configuration.

The next example develops a CBAC inspection rule for sites that might have remote traffic through an ATM interface. This example further illustrates on how to configure CBAC and emphasizes the application of the configuration rule at the interface, whatever that interface might be. This example does not provide a complete router configuration and does not describe other elements of the configuration.

The remote-office examples also focus on the firewall configuration but do not provide detailed descriptions of other configuration elements, such as the Basic Rate Interface (BRI) and dialer interface configurations.

Other examples provide more complete firewall configurations, further illustrating ways in which to apply CBAC.

In each example, configuring protocol inspection using CBAC has four components:

- Defining an access list with the appropriate permissions.
- Applying the ACL at an interface where you want to control access.
- Defining an inspection rule that includes the protocol that you want to inspect.
- Applying the inspection rule at an interface where you want to inspect traffic.

Ethernet Interface Configuration Example

This example looks at each of these four components. For this example, CBAC is being configured to inspect RTSP and H.323 protocol traffic inbound from the protected network on a router with two Ethernet interfaces. Interface Ethernet1/0 is the protected network and interface Ethernet1/1 is the unprotected network. The security policy for the protected site uses access control lists (ACLs) to restrict inbound traffic on the unprotected interface to specific ICMP protocol traffic, denying inbound access for TCP and UDP protocol traffic. Inbound access for specific protocol traffic is provided through dynamic access lists, which are generated according to CBAC inspection rules.

ACL 100 denies TCP and UDP traffic from any source or destination while permitting specific ICMP protocol traffic. The final deny statement is not required, but is included for explicitness—the final entry in any ACL is an implicit denial of all IP protocol traffic.

```
Router(config)# access-list 100 deny tcp any any
Router(config)# access-list 100 deny udp any any
Router(config)# access-list 100 permit icmp any any echo-reply
Router(config)# access-list 100 permit icmp any any time-exceeded
Router(config)# access-list 100 permit icmp any any packet-too-big
Router(config)# access-list 100 permit icmp any any traceroute
Router(config)# access-list 100 permit icmp any any unreachable
Router(config)# access-list 100 deny ip any any
```

ACL 100 is applied inbound at interface Ethernet1/1 to block all access from the unprotected network to the protected network.

```
Router(config)# interface Ethernet1/1
Router(config-if)# ip access-group 100 in
```

An inspection rule is created for “hqusers” that covers two protocols: RTSP and H.323.

```
Router(config)# ip inspect name hqusers rtsp
Router(config)# ip inspect name hqusers h323
```

The inspection rule is applied inbound at interface Ethernet1/0 to inspect traffic from users on the protected network. When CBAC detects multimedia traffic from the protected network, CBAC creates dynamic entries in access list 100 to allow return traffic for multimedia sessions.

```
Router(config)# interface Ethernet1/0
Router(config-if)# ip inspect hqusers in
```

ATM Interface Configuration Example

In this example, CBAC inspection (firewall protection) is required against inbound traffic on an ATM interface. This example might apply to sites where local hosts require access to hosts or services on a remote network. The security policy for this site uses access control lists (ACLs) to restrict inbound traffic on the ATM interface to specific ICMP protocol traffic, denying inbound access for TCP and UDP protocol traffic. Inbound access for specific TCP and UDP protocol traffic is provided through dynamic access lists, which are generated according to CBAC inspection rules.

For information on how to select the interface on which to apply CBAC, refer to the “Picking an Interface: Internal or External” section.

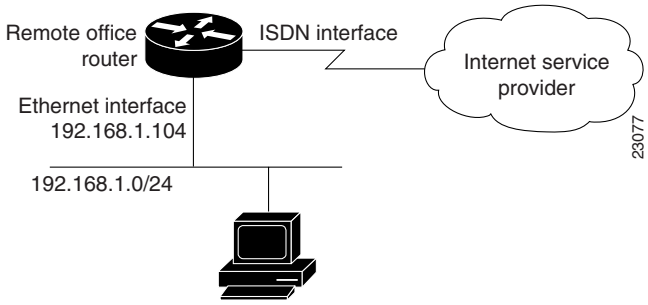
Note For Frame Relay or ATM interfaces, you can apply CBAC inspection rules separately on each sub-interface, even though the subinterfaces are physically connected through one interface.

```
! -------------------------
! Create the Inspection Rule
! -------------------------
!
! Create the CBAC inspection rule "test", allowing inspection of the protocol traffic
! specified by the rule. This inspection rule sets the timeout value to 30 seconds for
! each protocol (except for RPC). The timeout value defines the maximum time that a
! connection for a given protocol can remain active without any traffic passing through
! the router. When these timeouts are reached, the dynamic ACLs that are inserted to
! permit the returning traffic are removed, and subsequent packets (possibly even valid
! ones) are not permitted.
ip inspect name test cuseeme timeout 30
ip inspect name test ftp timeout 30
ip inspect name test h323 timeout 30
ip inspect name test realaudio timeout 30
ip inspect name test rpc program-number 100000
ip inspect name test streamworks timeout 30
ip inspect name test vdolive timeout 30
!
! ------------------------------
! Create the Access Control List
! ------------------------------
!
! In this example, ACL 105 denies all TCP and UDP protocol traffic. ICMP traffic from
! subnet 192.168.1.0 is permitted to allow access for routing and control traffic.
! ACL 105 specifies that only the return traffic for protocols defined in the
! inspection rule is allow access through the interface where this rule is applied. The
! final deny statement is added for explicitness.
access-list 105 deny TCP any any
access-list 105 deny UDP any any
access-list 105 permit icmp any any echo-reply
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 time-exceeded
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 packet-too-big
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 traceroute
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 unreachable
access-list 105 deny ip any any
!
! --------------------------------
! Apply the Inspection Rule and ACL
! --------------------------------
!
! In this example, the inspection rule "test" is applied to traffic at interface ATM3/0
! for connections initiated in the outbound direction; that is, from hosts that are
! located on a local network. CBAC creates dynamic access list entries for traffic
! initiated by local hosts. These dynamic entries allow inbound (returning) traffic for
! that connection. ACL 105 is applied at interface ATM3/0 in the inbound direction to
! block traffic initiated from hosts on a remote network that is not part of an
! existing connection.
interface ATM3/0
    ip address 10.1.10.1 255.0.0.0
    ip access-group 105 in
    no ip directed-broadcast
    ip inspect test out
    no shutdown
    atm clock INTERNAL
    atm pvc 7 7 7 aal5snap
    map-group atm
```

Remote Office to ISP Configuration Example

This example describes one possible Cisco IOS Firewall configuration for a remote office router connected to an Internet service provider (ISP). In this configuration, the site security policy allows hosts on the local network to initiate traffic to the ISP while traffic inbound to the router from the ISP is blocked at the ISDN interface. Specific ICMP control message traffic is permitted through the firewall. No mail or Web services are available from the local network. Figure 23 illustrates this example.

Figure 23 Remote Office to ISP Sample Configuration



The firewall has two interfaces:

- An Ethernet interface connects to the internal protected network.

  Interface Ethernet0 has no ACL applied to it, meaning that all traffic initiated on the LAN is allowed access to the ISP. In this configuration example, Network Address Translation (NAT) is not turned on, and the addresses on interface Ethernet0 are reserved IP addresses. In a production environment, addresses on Ethernet0 either must be registered network addresses, or you must turn on NAT to hide these inside addresses from being visible on the Internet.

- An ISDN Basic Rate Interface (BRI) connects the router to the ISP. In this example, a dialer profile is used to control the BRI interface. This means that the ACL and CBAC inspection rules are applied at the dialer interface, not directly at the physical ISDN (BRI) interface using a dialer map.

```
! ------------------------------------
! General Cisco IOS Firewall Guidelines
! ------------------------------------
! The following global configuration entries illustrate good security practices.
enable secret 5 <elided>
no ip source-route
no cdp run
!
! --------------------------------
! Create the CBAC inspection rule
! --------------------------------
! Create the CBAC inspection rule STOP to allow inspection of the protocol traffic
! specified by the rule.
ip inspect name STOP tcp
ip inspect name STOP ftp
ip inspect name STOP smtp
ip inspect name STOP h323
ip inspect name STOP rcmd
!
! --------------------------------
! Create Access Control List 105
! --------------------------------
! ACL 105 denies all IP protocol traffic except for specific ICMP control traffic.
```

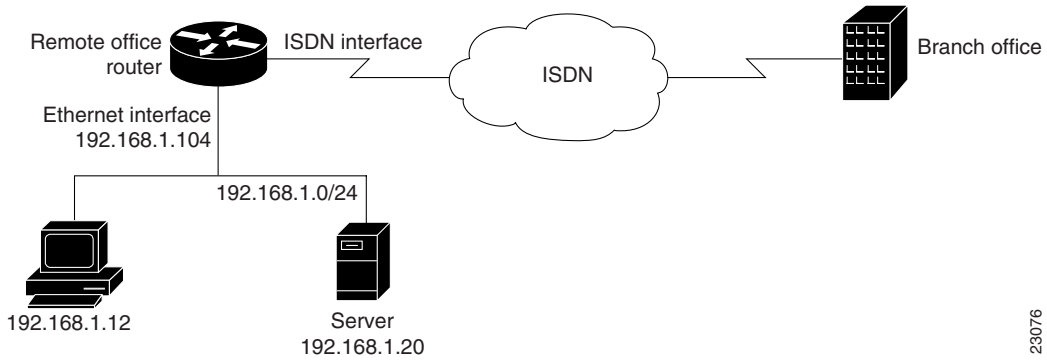
```
! This means that only the return traffic for protocols defined in the
! inspection rule and the specified ICMP traffic is allowed access through the
! interface where this rule is applied.
!
! Deny broadcast messages with a source address of 255.255.255.255; this helps to
! prevent broadcast attacks.
access-list 105 deny ip host 255.255.255.255 any
!
! Add anti-spoofing protection by denying traffic with a source address matching a host
! on the Ethernet interface.
acl 105 deny ip 192.168.1.0 0.0.0.255 any
!
! ICMP traffic is not inspected by CBAC. To control the type of ICMP traffic at the
! interface, add static access list entries. This example has the following ICMP
! requirements: outgoing ping commands require echo-reply messages to come back,
! outgoing traceroute commands require time-exceeded messages to come back, path MTU
! discovery requires "too-big" messages to come back, and incoming traceroute
! messages must be allowed. Additionally, permit all "unreachable" messages to come
! back; that is, if a router cannot forward or deliver a datagram, it sends an ICMP
! unreachable message back to the source and drops the datagram.
access-list 105 permit icmp any any echo-reply
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 time-exceeded
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 packet-too-big
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 traceroute
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 unreachable
!
! Final deny for explicitness. This entry is not required but helps complete the access
! list picture. By default, the final entry in any access list is an implicit deny of
! IP protocol traffic. This ensures that the firewall blocks any traffic not explicitly
! permitted by the access list.
access-list 105 deny ip any any
!
! ------------------------------------------------------------
! Configure the interface
! ------------------------------------------------------------
! In this example, no ACLs or inspection rules are applied at interface Ethernet0,
! meaning that all traffic on the local network is allowed to go out. This assumes a
! high-level of trust for the users on the local network.
interface Ethernet0
    ip address 192.168.1.104 255.255.255.0
!
no ip directed-broadcast
!
! This example uses a dialer profile, so the ACL and CBAC inspection rules are applied
! at the dialer interface, not the physical BRI interface. The dialer pool-member
! command is used to associate the physical interface with a dialer profile.
interface BRI0
    no ip address
    no ip directed-broadcast
    encapsulation ppp
    dialer pool-member 1
    isdn switch-type basic-5ess
!
! ------------------------------------------------------------------------
! Create the dialer profile.
! ------------------------------------------------------------------------
! Through the dialer profile, the ACL and CBAC inspection rules are
! applied to every pool member. In this example, the ACL is applied in, meaning that it
! applies to traffic inbound from the ISP. The CBAC inspection rule STOP is applied
! out, meaning that CBAC monitors the traffic through the interface and controls return
! traffic to the router for an existing connection.
```

```
interface Dialer0
    ip address negotiated
    ip access-group 105 in
    no ip directed-broadcast
    ip inspect STOP out
    encapsulation ppp
    dialer remote-name <ISP router>
    dialer idle-timeout 500
    dialer string <elided>
    dialer pool 1
    dialer-group 1
    ppp authentication callin
!
! ----------------------------------------------------------
! Additional entries
! ----------------------------------------------------------
! Configure the router to forward packets destined for an unrecognized subnet of
! a directly connected network.
ip classless
! Route traffic to the dialer interface.
ip route 0.0.0.0 0.0.0.0 Dialer0
! Include a dialer list protocol entry to specify the protocol that triggers dialing.
dialer-list 1 protocol ip permit
! Add a user name (name of the router your are configuring) and password for caller
! identification and password authentication with the ISP router.
username <router host name> password 5 <elided>
```

Remote Office to Branch Office Configuration Example

This example describes one possible Cisco IOS Firewall configuration for a remote office router connected to a branch office. In this configuration, the site security policy allows hosts on the local network to initiate traffic to the branch office. Mail or Web services are available from a server on the local network, and access to these services is available from the branch office. Traffic from the branch office, except for mail and Web traffic, is blocked at the outside interface. Specific ICMP control message traffic is permitted through the firewall. Figure 24 illustrates this example.

Figure 24 Remote Office to Branch Office Sample Configuration



The firewall has two interfaces:

- An Ethernet interface connects to the internal protected network.

  Interface Ethernet0 has no ACL applied to it, meaning that all traffic initiated from the LAN is allowed access through the firewall.

- An ISDN Basic Rate Interface (BRI) connects the router to the branch office. In this example, a dialer profile is used to control the BRI interface. This means that the ACL and CBAC inspection rules are applied at dialer interface, not directly at the physical ISDN (BRI) interface.

```
! ----------------------------------------
! General firewall configuration guidelines
! ----------------------------------------
! The following global configuration entries illustrate good security practices.
enable secret 5 <elided>
no ip source-route
no cdp run
!
! --------------------------
! Create the Inspection Rule
! --------------------------
! Create the CBAC inspection rule STOP to allow inspection of the specified protocol
! traffic. Create the inspection rule GO to allow inspection of SMTP traffic.
ip inspect name STOP tcp
ip inspect name STOP ftp
ip inspect name STOP smtp
ip inspect name STOP h323
ip inspect name GO smtp
!
! ---------------------------------------------------------------
! Create Access Control Lists 106 and 51
! ---------------------------------------------------------------
! ACL 106 permits mail and Web traffic from any host to the specified server. ACL 106
! denies all other ip protocol traffic except for specific ICMP control traffic.
! This means that only the return traffic for protocols defined in the
! inspection rule and the specified ICMP traffic is allowed access through the
! interface where this rule is applied.
!
! Deny broadcast messages with a source address of 255.255.255.255; this helps to
! prevent broadcast attacks.
access-list 106 deny ip host 255.255.255.255 any
!
! Add anti-spoofing protection by denying traffic with a source address matching a host
! on the Ethernet interface.
access-list 106 deny ip 192.168.1.0 0.0.0.255 any
!
! ICMP traffic is not inspected by CBAC. To control the type of ICMP traffic at the
! interface, add static access list entries. This example has the following ICMP
! requirements: outgoing ping commands require echo-reply messages to come back,
! outgoing traceroute commands require time-exceeded messages to come back, path MTU
! discovery requires "too-big" messages to come back, and incoming traceroute must be
! allowed. Additionally, permit all "unreachable" messages to come back; that is, if a
! router cannot forward or deliver a datagram, it sends an ICMP unreachable message
! back to the source and drops the datagram.
access-list 106 permit icmp any any echo-reply
access-list 106 permit icmp any 192.168.1.0 0.0.0.255 time-exceeded
access-list 106 permit icmp any 192.168.1.0 0.0.0.255 packet-too-big
access-list 106 permit icmp any 192.168.1.0 0.0.0.255 traceroute
access-list 106 permit icmp any 192.168.1.0 0.0.0.255 unreachable
!
! Permit mail and Web access to a specific server.
access-list 106 permit tcp any host 192.168.1.20 eq smtp
access-list 106 permit tcp any host 192.168.1.20 eq www
!
```

```
! Final deny for explicitness. This entry is not required but helps complete the access
! list picture. By default, the final entry in any access list is an implicit deny of
! IP protocol traffic. This ensures that the firewall blocks any traffic not explicitly
! permitted by the access list.
access-list 106 deny ip any any
!
! ----------------------------------------------------------
! Configure the interface.
! ----------------------------------------------------------
! In this example, no ACLs or inspection rules are applied at interface Ethernet0,
! meaning that all traffic on the local network is allowed to go out. This assumes a
! high-level of trust for the users on the local network.
interface Ethernet0
    ip address 192.168.1.104 255.255.255.0
    no ip directed-broadcast
!
! This example uses a dialer profile, so the ACL and CBAC inspection rules are applied
! at the dialer interface, not the physical BRI interface. The dialer pool-member
! command is used to associate the physical interface with a dialer profile.
interface BRI0
    no ip address
    no ip directed-broadcast
    encapsulation ppp
    dialer pool-member 1
    isdn switch-type basic-5ess
!
! ------------------------------------------------------------------
! Apply the ACL and CBAC inspection rules at the dialer interface.
! ------------------------------------------------------------------
! Through the dialer profile, the ACL and CBAC inspection rules are
! applied to every pool member. In this example, the ACL is applied in, meaning that it
! applies to traffic inbound from the branch office. The CBAC inspection rule STOP is
! applied out, meaning that CBAC monitors the traffic and controls return traffic to
! the router for an existing connection. The CBAC inspection rule GO is applied in,
! protecting against certain types of DoS attacks as described in this document. Note
! that the GO inspection rule does not control return traffic because there is no ACL
! blocking traffic in that direction; however, it does monitor the connections.
interface Dialer0
    ip address <ISDN interface address>
    ip access-group 106 in
    no ip directed-broadcast
    ip inspect STOP out
    ip inspect GO in
    encapsulation ppp
    dialer remote-name <branch office router>
    dialer idle-timeout 500
    dialer string <elided>
    dialer pool 1
    dialer-group 1
    ppp authentication
!
! ----------------------------------------------------------
! Additional entries
! ----------------------------------------------------------
! Configure the router to forward packets destined for an unrecognized subnet of
! a directly connected network.
ip classless
! Route traffic to the dialer interface.
ip route 0.0.0.0 0.0.0.0 Dialer0
! Include a dialer list protocol entry to specify the protocol that triggers dialing.
dialer-list 1 protocol ip permit
! Add a user name (name of the router your are configuring) and password for caller
! identification and password authentication with the ISP router.
username <router host name> password 5 <elided>
```

Two-Interface Branch Office Configuration Example

This sample configuration file describes a firewall configured with CBAC. The firewall is positioned between a protected field office's internal network and a WAN connection to the corporate headquarters. CBAC is configured on the firewall in order to protect the internal network from potential network threats coming from the WAN side.

The firewall has two interfaces configured:

- Interface Ethernet0 connects to the internal protected network
- Interface Serial0 connects to the WAN with Frame Relay

```
! ------------------------------------------------------------------------
! This first section contains some configuration that is not required for CBAC,
! but illustrates good security practices. Note that there are no
! services on the Ethernet side. Email is picked up via POP from a server on the
! corporate side.
! ------------------------------------------------------------------------
!
hostname user1-examplecorp-fr
!
boot system flash c1600-fw1600-l
enable secret 5 <elided>
!
username user1 password <elided>
ip subnet-zero
no ip source-route
ip domain-name example.com
ip name-server 172.19.2.132
ip name-server 198.92.30.32
!
!
! ------------------------------------------------------------------------
! The next section includes configuration required specifically for CBAC.
! ------------------------------------------------------------------------
!
! The following commands define the inspection rule "myfw", allowing
! the specified protocols to be inspected. Note that Java applets will be permitted
! according to access list 51, defined later in this configuration.
ip inspect name myfw cuseeme timeout 3600
ip inspect name myfw ftp timeout 3600
ip inspect name myfw http java-list 51 timeout 30
ip inspect name myfw rcmd timeout 3600
ip inspect name myfw realaudio timeout 3600
ip inspect name myfw smtp timeout 3600
ip inspect name myfw tftp timeout 30
ip inspect name myfw udp timeout 15
ip inspect name myfw tcp timeout 3600
!
! The following interface configuration applies the "myfw" inspection rule to
! inbound traffic at Ethernet 0. Since this interface is on the internal network
! side of the firewall, traffic entering Ethernet 0 is actually
! exiting the internal network. Applying the inspection rule to this interface causes
! inbound traffic (which is exiting the network) to be inspected; return traffic will
! only be permitted back through the firewall if part of a session which began from
! within the network.
! Also note that access list 101 is applied to inbound traffic at Ethernet 0.
! (Traffic blocked by the access list will not be inspected.)
interface Ethernet0
description ExampleCorp Ethernet chez user1
ip address 172.19.139.1 255.255.255.248
ip broadcast-address 172.19.131.7
```

```
no ip directed-broadcast
no ip proxy-arp
ip inspect myfw in
ip access-group 101 in
no cdp enable
!
interface Serial0
description Frame Relay (Telco ID 22RTQQ062438-001) to ExampleCorp HQ
no ip address
ip broadcast-address 0.0.0.0
encapsulation frame-relay IETF
no arp frame-relay
bandwidth 56
service-module 56k clock source line
service-module 56k network-type dds
frame-relay lmi-type ansi
!
! Note that the following interface configuration applies access list 111 to
! inbound traffic at the external serial interface. (Inbound traffic is
! entering the network.) When CBAC inspection occurs on traffic exiting the
! network, temporary openings will be added to access list 111 to allow returning
! traffic that is part of existing sessions.
!
interface Serial0.1 point-to-point
ip unnumbered Ethernet0
ip access-group 111 in
bandwidth 56
no cdp enable
frame-relay interface-dlci 16
!
ip classless
ip route 0.0.0.0 0.0.0.0 Serial0.1
!
! The following access list defines "friendly" and "hostile" sites for Java
! applet blocking. Because Java applet blocking is defined in the inspection
! rule "myfw" and references access list 51, applets will be actively denied
! if they are from any of the "deny" addresses and allowed only if they are from
! either of the two "permit" networks.
!
access-list 51 deny   172.19.1.203
access-list 51 deny   172.19.2.147
access-list 51 permit 172.18.0.0 0.1.255.255
access-list 51 permit 192.168.1.0 0.0.0.255
access-list 51 deny   any
!
! The following access list 101 is applied to interface Ethernet 0 above.
! This access list permits all traffic that should be CBAC inspected, and also
! provides anti-spoofing. The access list is deliberately set up to deny unknown
! IP protocols, because no such unknown protocols will be in legitimate use.
!
access-list 101 permit tcp 172.19.139.0 0.0.0.7 any
access-list 101 permit udp 172.19.139.0 0.0.0.7 any
access-list 101 permit icmp 172.19.139.0 0.0.0.7 any
access-list 101 deny   ip any any
!
! The following access list 111 is applied to interface Serial 0.1 above.
! This access list filters traffic coming in from the external side. When
! CBAC inspection occurs, temporary openings will be added to the beginning of
! this access list to allow return traffic back into the internal network.
! This access list should restrict traffic that will be inspected by
! CBAC. (Remember that CBAC will open holes as necessary to permit returning traffic.)
! Comments precede each access list entry. These entries are not all specifically
! related to CBAC, but are created to provide general good security.
```

```
!
! Anti-spoofing.
access-list 111 deny   ip 172.19.139.0 0.0.0.7 any
! Sometimes EIGRP is run on the Frame Relay link. When you use an
! input access list, you have to explicitly allow even control traffic.
! This could be more restrictive, but there would have to be entries
! for the EIGRP multicast as well as for the office's own unicast address.
access-list 111 permit igrp any any
!
! These are the ICMP types actually used...
! administratively-prohibited is useful when you are trying to figure out why
! you cannot reach something you think you should be able to reach.
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 administratively-prohibited
!
! This allows network admins at headquarters to ping hosts at the field office:
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 echo
!
! This allows the field office to do outgoing pings
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 echo-reply
!
! Path MTU discovery requires too-big messages
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 packet-too-big
!
! Outgoing traceroute requires time-exceeded messages to come back
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 time-exceeded
!
! Incoming traceroute
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 traceroute
!
! Permits all unreachables because if you are trying to debug
! things from the remote office, you want to see them. If nobody ever did
! any debugging from the network, it would be more appropriate to permit only
! port unreachables or no unreachables at all.
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 unreachable
!
!
! These next two entries permit users on most ExampleCorp networks to Telnet to
! a host in the field office. This is for remote administration by the network admins.
access-list 111 permit tcp 172.18.0.0 0.1.255.255 host 172.19.139.1 eq telnet
access-list 111 permit tcp 192.168.1.0 0.0.0.255 host 172.19.139.1 eq telnet
!
! Final deny for explicitness
access-list 111 deny ip any any
!
no cdp run
snmp-server community <elided> RO
!
line con 0
exec-timeout 0 0
password <elided>
login local
line vty 0
exec-timeout 0 0
password <elided>
login local
length 35
line vty 1
exec-timeout 0 0
password 7 <elided>
login local
line vty 2
exec-timeout 0 0
password 7 <elided>
login local
```

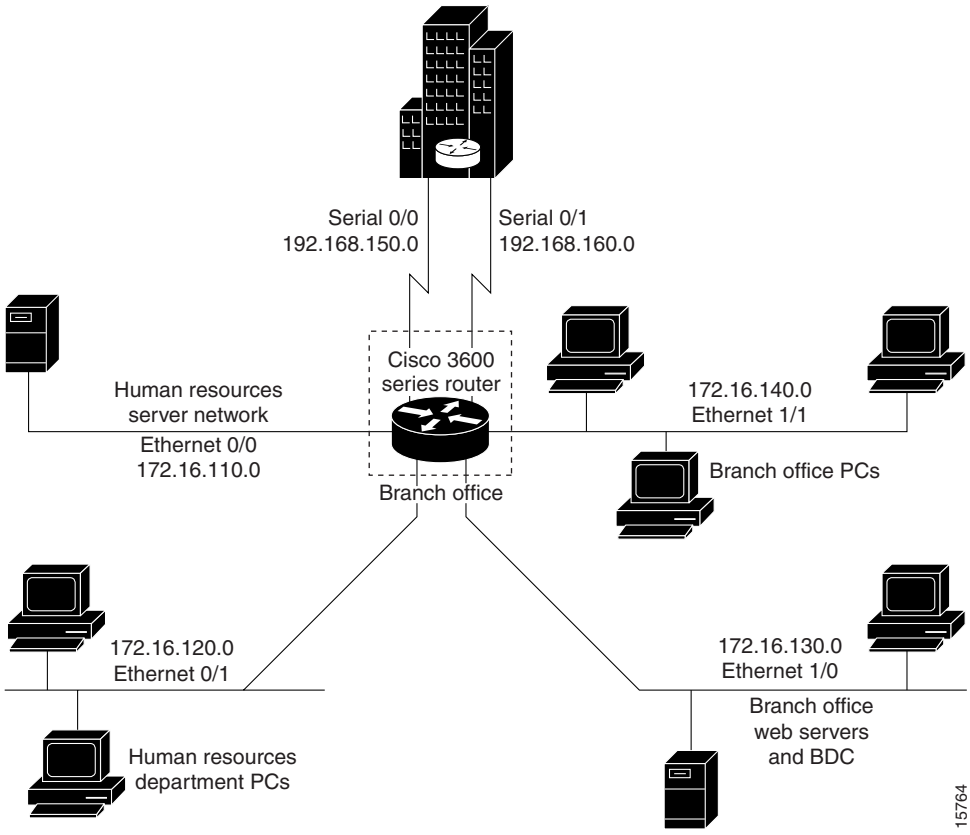
```
line vty 3
exec-timeout 0 0
password 7 <elided>
login local
line vty 4
exec-timeout 0 0
password 7 <elided>
login local
!
scheduler interval 500
end
```

Multiple-Interface Branch Office Configuration Example

In this configuration example, a single Cisco 3600 series firewall router is positioned at a branch office. It has four internal networks and two WAN connections to the corporate headquarters. CBAC is configured on the firewall to protect two of the internal networks from potential network threats coming from the WAN side and from less secure internal networks. Anti-spoofing protection is added at each interface with client systems. Figure 25 illustrates this configuration.

Note This example shows a moderately high level of trust by the administrators toward the expected users. Additional protection could be added to this configuration for a situation in a lower level of trust. That configuration would include ICMP filtering statements, significantly more protocol and address control through the use of more restrictive access control lists, and anti-spoofing applied everywhere. This configuration does not contain those additional restrictions because that would detract from the CBAC example.

Figure 25 Sample Cisco IOS Firewall Application Environment



The branch office has this sample network configuration:

- Ethernet interface 0/0 supports the Human Resources department servers. This network includes an email (SMTP and POP3) host and a Windows NT server. The Windows NT server is the Primary Domain Controller (PDC) for the Human Resources domain and has a trust relationship with the rest of the company; however, it contains applications and databases that must not be accessed by the rest of the company or the other groups in the branch office. The devices on this LAN are accessible only by users in the Human Resources department on Ethernet interface 0/1. The Mail server must be able to send and receive email (through SMTP sessions) with all other devices. The Windows 95 machines can use this machine as their email server (for sending email through SMTP sessions) and as a repository for accumulating email that they can then download through POP3 sessions. No one else in the company is allowed to form POP3 sessions to any machine on this LAN.
- Ethernet interface 0/1 supports the Windows 95 computers in the Human Resources department. These users must have access to the Human Resources mail servers located on Ethernet interface 0/0 as well as access to the rest of the company. Access to the Windows NT server resources are controlled through the Windows NT permissions assigned to each user in the Windows NT domain.

- Ethernet interface 1/0 supports the branch office web servers, which can be accessed by everyone in the company. These servers use TCP ports 80 (HTTP) and 443 (SHTTP) for inbound Web access. This network also includes a backup domain controller (BDC) for the overall domain that is also used as file, print, and service server.

  Ethernet interface 1/1 supports all users who are not in the Human Resources department. These users have no access to the Human Resources department servers, but they can access the other network interfaces and the serial interfaces for WAN connectivity. Serial interface 0/0 and 0/1 connect to the WAN with T1 links (links to corporate headquarters). In this sample configuration, the Domain Name System (DNS) servers are located somewhere within the rest of the company.

  Additionally, network management (SNMP) and Telnet sessions are limited to the management network (192.168.55.0), which is located somewhere within the rest of the company across the serial interface.

```
! --------------------------------------------------------------------
! This first section contains some configuration that is not required
! for CBAC, but illustrates good security practices.
! --------------------------------------------------------------------
! Add this line to get timestamps on the syslog messages.
service timestamps log datetime localtime show-timezone
!
hostname Router1
!
boot system flash c3600-fw3600-l
!
! Configure AAA user authentication.
aaa new-model
aaa authentication login lista group tacacs+ enable
!
enable secret 5 <elided>
ip subnet-zero
!
! Disable source routing to help prevent spoofing.
no ip source-route
!
! Set up the domain name and server IP addresses.
ip domain-name example.com
ip name-server 192.168.55.132
ip name-server 192.168.27.32
!
! The audit-trail command enables the delivery of specific CBAC messages
! through the syslog notification process.
ip inspect audit-trail
!
! Establish the time-out values for DNS queries. When this idle-timer expires,
! the dynamic ACL entries that were created to permit the reply to a DNS request
! will be removed and any subsequent packets will be denied.
ip inspect dns-timeout 10
!
! ------------------------------------------------------------------------------
! The next section includes configuration statements required specifically for CBAC.
! ------------------------------------------------------------------------------
! Define the CBAC inspection rule "inspect1", allowing the specified protocols to be
! inspected. The first rule enables SMTP specific inspection. SMTP inspection causes
! the exchange of the SMTP session to be inspected for illegal commands. Any packets
! with illegal commands are dropped, and the SMTP session will hang and eventually
! time out.
ip inspect name inspect1 smtp timeout 30
```

```
!
! In the next two lines of inspect1, define the maximum time that each of the UDP and
! TCP sessions are allowed to continue without any traffic passing
! through the router. When these timeouts are reached, the dynamic ACLs that
! are inserted to permit the returning traffic are removed and subsequent packets
! (possibly even valid ones) will not be permitted.
ip inspect name inspect1 udp timeout 30
ip inspect name inspect1 tcp timeout 30
!
! Define the CBAC inspection rule "inspect2", allowing the specified protocols to be
! inspected. These rules are similar to those used in the inspection rule "inspect1,"
! except that on the interfaces where this rule is applied, SMTP sessions are not
! expected to go through; therefore, the SMTP rule element is not applied here.
ip inspect name inspect2 udp timeout 30
ip inspect name inspect2 tcp timeout 30
!
! ------------------------------------------------------------------------
! The next section shows the Ethernet interface configuration statements for each
! interface, including access lists and inspections rules.
! ------------------------------------------------------------------------
! Apply the "inspect1" inspection rule to sessions that are initiated in the outbound
! direction (toward the LAN) at Ethernet interface 0/0. All packets in these sessions
! will be inspected by CBAC. Provided that network traffic passes the Access Control
! List (ACL) restrictions, traffic is then inspected by CBAC for access through the
! Cisco Secure Integrated Software. Traffic blocked by the access list is not inspected
! by CBAC. Access list 110 is applied to outbound traffic on this interface.
interface Ethernet0/0
    description HR_Server Ethernet
    ip address 172.16.110.1 255.255.255.0
    ip access-group 110 out
    no ip directed-broadcast
    no ip proxy-arp
    ip inspect inspect1 out
    no cdp enable
!
! Apply access list 120 to inbound traffic on Ethernet interface 0/1.
! Applying access list 120 to inbound traffic provides anti-spoofing on this interface
! by dropping traffic with a source address matching the IP address on a network other
! than Ethernet 0/1. The IP helper address lists the IP address of the DHCP server on
! Ethernet interface 1/0.
interface Ethernet0/1
    description HR_client Ethernet
    ip address 172.16.120.1 255.255.255.0
    ip access-group 120 in
    ip helper-address 172.16.130.66
    no ip directed-broadcast
    no ip proxy-arp
    no cdp enable
!
! Apply the "inspect2" inspection rule to sessions that are initiated in the outbound
! direction (toward the LAN) at Ethernet interface 1/0. Provided that network traffic
! passes the Access Control List (ACL) restrictions, traffic is then inspected by CBAC
! through the Cisco Secure Integrated Software. Traffic blocked by the access list is
! not inspected
! by CBAC. Access list 130 is applied to outbound traffic on this interface.
interface Ethernet1/0
    description Web_server Ethernet
    ip address 172.16.130.1 255.255.255.0
    ip access-group 130 out
    no ip directed-broadcast
    no ip proxy-arp
    ip inspect inspect2 out
    no cdp enable
```

```
!
! Apply access list 140 to inbound traffic at Ethernet interface 1/1. This
! provides anti-spoofing on the interface by dropping traffic with a source address
! matching the IP address of a network other than Ethernet 1/1. The IP helper address
! lists the IP address of the DHCP server on Ethernet interface 1/0.
interface Ethernet1/1
    description Everyone_else Ethernet
    ip address 172.16.140.1 255.255.255.0
    ip access-group 140 in
    ip helper-address 172.16.130.66
    no ip directed-broadcast

    no ip proxy-arp
    no cdp enable
!
! ------------------------------------------------------------------------
! The next section configures the serial interfaces, including access lists.
! ------------------------------------------------------------------------
! Apply access list 150 to Serial interfaces 0/0. This provides anti-spoofing on the
! serial interface by dropping traffic with a source address matching the IP address
! of a host on Ethernet interface 0/0, 0/1, 1/0, or 1/1.
interface Serial0/0
    description T1 to HQ
    ip address 192.168.150.1 255.255.255.0
    ip access-group 150 in
    bandwidth 1544
!
interface Serial1/1
    description T1 to HQ
    ip address 192.168.160.1 255.255.255.0
    ip access-group 150 in
    bandwidth 1544
!
! ------------------------------
! Configure routing information.
! ------------------------------
router igrp 109
network 172.16.0.0
network 192.168.150.0
network 192.168.160.0
!
! Define protocol forwarding on the firewall. When you turn on a related command,
! ip helper-address, you forward every IP broadcast in the ip forward protocol
! command list, including several which are on by default: TFTP (port 69),
! DNS (port 53), Time service (port 37), NetBIOS Name Server (port 137),
! NetBIOS Datagram Server (port 138), BOOTP client and server datagrams
! (ports 67 and 68), and TACACS service (port 49). One common
! application that requires helper addresses is Dynamic Host Configuration
! Protocol (DHCP). DHCP information is carried inside of BOOTP packets. The
! "no ip forward protocol" statements turn off forwarding for the specified protocols.
no ip forward-protocol udp netbios-ns
no ip forward-protocol udp netbios-dgm
no ip forward-protocol udp tacacs
no ip forward-protocol udp tftp
ip forward-protocol udp bootpc
!
! Add this line to establish where router SYSLOG messages are sent. This includes the
! CBAC messages.
logging 192.168.55.131
```

```
!
! ----------------------------------------------------------------
! Define the configuration of each access list.
! ----------------------------------------------------------------
! Defines Telnet controls in access list 12.
access-list 12 permit 192.168.55.0 0.0.0.255
!
! Defines SNMP controls in access list 13.
access-list 13 permit 192.168.55.12
access-list 13 permit 192.168.55.19
!
! Access list 110 permits TCP and UDP protocol traffic for specific ports and with a
! source address on Ethernet interface 0/1. The access list denies IP protocol traffic
! with any other source and destination address. The access list permits ICMP access
! for any source and destination address. Access list 110 is deliberately set up to
! deny unknown IP protocols because no such unknown protocols will be in legitimate
! use. Access list 110 is applied to outbound traffic at Ethernet interface 0/0. In ACL
! 110, network traffic is being allowed access to the ports on any server on the HR
! server network. In less trusted environments, this can be a security problem;
! however, you can limit access more severely by specifying specific destination
! addresses in the ACL statements.
access-list 110 permit tcp 172.16.120.0 0.0.0.255 any eq smtp
access-list 110 permit tcp 172.16.120.0 0.0.0.255 any eq pop3
access-list 110 permit tcp 172.16.120.0 0.0.0.255 any eq 110
access-list 110 permit udp any any eq 137
access-list 110 permit udp any any eq 138
access-list 110 permit udp any any eq 139
access-list 110 permit icmp any any
access-list 110 deny ip any any!
!
! Access-list 120 permits TCP, UDP, and ICMP protocol traffic with a source address
! on Ethernet interface 0/1, but denies all other IP protocol traffic. Access list
! 120 is applied to inbound traffic on Ethernet interface 0/1.
access-list 120 permit tcp 172.16.120.0 0.0.0.255 any
access-list 120 permit udp 172.16.120.0 0.0.0.255 any
access-list 120 permit icmp 172.16.120.0 0.0.0.255 any
access-list 120 deny ip any any
!
! Access list 130 permits TCP, UDP, and ICMP protocol traffic for specific ports and
! with any source and destination address. It opens access to the web server and to
! all NBT services to the rest of the company, which can be controlled through the
! trust relations on the Windows NT servers. The bootpc entry permits access to the
! DHCP server. Access list 130 denies all other IP protocol traffic. Access list 130 is
! applied to outbound traffic at Ethernet interface 1/0.
access-list 130 permit tcp any any eq www
access-list 130 permit tcp any any eq 443
access-list 130 permit tcp any any eq 110
access-list 130 permit udp any any eq 137
access-list 130 permit udp any any eq 138
access-list 130 permit udp any any eq 139
access-list 130 permit udp any any eq bootpc
access-list 130 permit icmp any any
access-list 130 deny ip any any
!
! Access list 140 permits TCP, UDP, and ICMP protocol traffic with a source address on
! Ethernet interface 1/1, and it denies all other IP protocol traffic. Access list 140
! is applied to inbound traffic at Ethernet interface 1/1.
access-list 140 permit tcp 172.16.140.0 0.0.0.255 any
access-list 140 permit udp 172.16.140.0 0.0.0.255 any
access-list 140 permit icmp 172.16.140.0 0.0.0.255 any
access-list 140 deny ip any any
```

```
!
! Access list 150 denies IP protocol traffic with a source address on Ethernet
! interfaces 0/0, 0/1, 1/0, and 1/1, and it permits IP protocol traffic with any other
! source and destination address. Access list 150 is applied to inbound traffic
! on each of the serial interfaces.
access-list 150 deny ip 172.16.110.0 0.0.0.255 any
access-list 150 deny ip 172.16.120.0 0.0.0.255 any
access-list 150 deny ip 172.16.130.0 0.0.0.255 any
access-list 150 deny ip 172.16.140.0 0.0.0.255 any
access-list 150 permit ip any any
!
! Disable Cisco Discovery Protocol.
no cdp run
!
snmp-server community <elided> ro 13
tacacs-server host 192.168.55.2
tacacs-server key <elided>
!
! ----------------------------------------------------------------------------
! Configures the router console port and the virtual terminal line interfaces,
! including AAA authentication at login. Authentication is required for users defined
! in "lista." Access-class 12 is applied on each line, restricting Telnet access to
! connections with a source address on the network management network.
! ----------------------------------------------------------------------------
line console 0
exec-timeout 3 00
login authentication lista
line aux 0
exec-timeout 3 00
login authentication lista
line vty 0
    exec-timeout 1 30
    login authentication lista
    access-class 12 in
line vty 1
    exec-timeout 1 30
    login authentication lista
    access-class 12 in
line vty 2
    exec-timeout 1 30
    login authentication lista
    access-class 12 in
line vty 3
    exec-timeout 1 30
    login authentication lista
    access-class 12 in
line vty 4
    exec-timeout 1 30
    login authentication lista
    access-class 12 in
!
end
```

Cisco IOS Firewall Performance Improvements

Feature History

Release	Modification
12.2(8)T	This feature was introduced.

This document describes the Cisco IOS Firewall Performance Improvements feature in Cisco IOS Release 12.2(8)T. It includes the following sections:

- Feature Overview, page 525
- Supported Platforms, page 527
- Supported Standards, MIBs, and RFCs, page 528
- Configuration Tasks, page 529
- Configuration Examples, page 529
- Command Reference, page 530

Feature Overview

The Cisco IOS Firewall Performance Improvements feature introduces three performance metrics for Context-Based Access Control (CBAC)—Throughput Improvement, Connections Per Second Improvement, and CPU Utilization Improvement.

CBAC is a context-based firewall that performs the following:

- Inspects traffic in one direction for network, transport, and application layer information
- Extracts relevant port information
- Dynamically creates access list entries for return traffic
- Closes ports at the end of a connection

CBAC also forces protocol conformance for some protocols, has a limited vulnerability signature detection mechanism, and extensive denial-of-service (DOS) prevention mechanisms. However, many of these features are CPU intensive, thereby, adversely affecting the performance of the router. The router is also affected during times of heavy traffic due to the processing of the base engine itself. With this feature, the performance of the router running CBAC is no longer subdued.

Throughput Improvement

Throughput is a metric defined by the number of packets transferred from the input interface to the output interface within 1 second by a router running CBAC. When the CBAC base engine inspects packets that belong to an existing session, it must find out which session the packet belongs to; thus, the base engine implements a hash table to search for the session of the packet.

Collisions in a hash table result in poor hash function distribution because many entries are hashed into the same bucket for certain patterns of addresses. Even if a hash function distribution evenly dispenses the input across all of the buckets, a small hashtable size will not scale well if there are a large number of sessions. As the number of sessions increase, the collisions increase, which increases the length of the linked lists, thereby, deteriorating the throughput performance.

The Cisco IOS Firewall Performance Improvements feature allows users to dynamically change the size of the session hash table without reloading the router by using the **ip inspect hashtable** command. By increasing the size of the hash table, the number of sessions per hash bucket can be reduced, which improves the throughput performance of the base engine.

Connections Per Second Improvement

Connections per second is a metric defined by the number of short-lived connections that can be created and deleted within 1 second by a router running CBAC. (These connections apply only to TCP connections because UDP is a connectionless protocol.)

Initially, CBAC had several restrictions that limited the connections per second metric. While a packet was being processed for connection setup and connection teardown of TCP connections, the base engine (which allocates and de-allocates memory while processing packets) would “bump up” several packets to the process switching path. Bumping up these packets drastically slowed down their processing. Also, the base engine had to process each packet again when it was bumped up into the process switching path, which also contributed to the degrading performance.

The Cisco IOS Firewall Performance Improvements feature prevents these restrictions by allowing only the first packet of any connection to be bumped up to the process switching path while the remaining packets are processed by the base engine in the fast path. Thus, the base engine is no longer slowed down by bumping up several packets or by processing packets twice.

Note In this document, a connection is defined as creating a session, sending a data packet, and immediately deleting a session.

CPU Utilization Improvement

The CPU utilization of the router running CBAC can be measured while a specific throughput or connections per second metric is maintained. This improvement is used in conjunction with the throughput and connections per second metrics.

Benefits

Layer 4 Processing Performance Improvement

This enhancement improves the connections per second metric and the CPU utilization. The code path for connection initiation and teardown was rewritten, thereby, enabling quicker creation of the connections per second metric, which reduces CPU utilization per connection.

Hash Table Function Performance Improvement

With this enhancement, the hash function has been rewritten to ensure better distribution. This newly improved feature allows users to dynamically configure the size of the session hash table from 1K to 8K. When a packet belonging to an existing session comes into the router, a hash table is used to map the packet to an existing firewall session. As the number of sessions increases, the number of sessions hashing into the same bucket increases if the size of the hash table is fixed. By allowing the user to change the size of the hash table when the number of concurrent sessions increases or to reduce the search time for the session, the throughput performance of the base engine is greatly improved.

Application Module Tuning Performance Improvement

This enhancement makes changes to application modules, ensuring that only the connection-initiating packet from all the packets belonging to the connection initiation and teardown is bumped up to the process switching path. Thus, the connections per second metric is significantly improved.

Restrictions

To benefit from the performance enhancements, your router must be running CBAC.

Related Documents

- "Traffic Filtering and Firewalls" part in the *Cisco IOS Security Configuration Guide*, Release 12.2
- "Traffic Filtering and Firewalls" part in the *Cisco IOS Security Command Reference*, Release 12.2

Supported Platforms

- Cisco 800 series
- Cisco 805
- Cisco 820
- Cisco 827
- Cisco 1400 series
- Cisco 1600 series
- Cisco 1700 series
- Cisco 2500 series
- Cisco 2600 series
- Cisco 3620
- Cisco 3640

- Cisco 3660
- Cisco 7100 series
- Cisco 7200 series
- Cisco 7500 series
- Cisco Catalyst 6500 series MSFC software
- Cisco uBR7200 series
- Cisco uBR905
- Cisco uBR925

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

None

Configuration Tasks

See the following sections for configuration tasks for the Cisco IOS Firewall Performance Improvements feature. Each task in the list is identified as either required or optional.

- Changing the Size of the Hash Table (required)
- Verifying CBAC Configurations (optional)

Changing the Size of the Hash Table

You can increase the hash table to improve packet distribution. To change the size of the session hash table, use the following command in global configuration mode:

Command	Purpose
`Router# ip inspect hashtable number`	Changes the size of the hash table. *number* specifies the size of the hash table in terms of buckets. Possible values for the hash table are 1024, 2048, 4096, and 8192; the default value is 1024. **Note** You should increase the hash table size when the total number of sessions running through the CBAC router is approximately twice the current hash size; decrease the hash table size when the total number of sessions is reduced to approximately half the current hash size. Essentially, try to maintain a 1:1 ratio between the number of sessions and the size of the hash table.

Verifying CBAC Configurations

To verify all CBAC configurations and all existing sessions that are currently being tracked and inspected by CBAC, use the **show ip inspect all** command in EXEC mode.

Configuration Examples

This section provides the following configuration example:

- Changing the Size of the Hash Table Example

Changing the Size of the Hash Table Example

The following example shows how to change the size of the hash table to 2048 buckets:

```
ip inspect hashtable 2048
```

Command Reference

The following new command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **ip inspect hashtable**

E-mail Inspection Engine

The E-mail Inspection Engine feature allows the Cisco IOS Firewall to inspect Post Office Protocol 3 (POP3) and Internet Message Access Protocol (IMAP) e-mail, in addition to Simple Mail Transfer Protocol (SMTP) and Extended Simple Mail Transfer Protocol (ESMTP) e-mail which were previously supported.

The **secure-login** enhancement allows people to download external POP3 e-mail only if authentication methods are secure.

Feature History for E-mail Inspection Engine

Release	Modification
12.3(14)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for E-mail Inspection Engine, page 532
- Restrictions for E-mail Inspection Engine, page 532
- Information About E-mail Inspection Engine, page 532
- How to Configure E-mail Inspection Engine, page 534
- Configuration Examples for E-mail Inspection Engine, page 537
- Additional References, page 538

- Command Reference, page 539
- Glossary, page 540

Prerequisites for E-mail Inspection Engine

- Configure CBAC.
- Enable SSL VPN tunnels.

Restrictions for E-mail Inspection Engine

None.

Information About E-mail Inspection Engine

To configure E-mail Inspection Engine, you need to understand the following concepts:

- E-mail Inspection Engine Operation, page 532
- Inspection, page 533
- POP3, page 533
- IMAP Protocol, page 533
- Client Command Validation, page 534
- SMTP, page 534
- SSL, page 534

E-mail Inspection Engine Operation

The client/server communication is validated from the time the TCP connection is initialized until the client is authenticated. The Cisco IOS Firewall uses a state router to track each stage of authentication. After the client is authenticated, the Cisco IOS Firewall allows all the client/server commands without further L7 inspection. TCP L4 inspection continues until the connection is closed. At the end of the e-mail session when the client host quits and before the TCP connection is closed, no further client/server interaction is allowed unless the client is reauthenticated.

During the authentication, any unrecognized command causes the Cisco IOS Firewall to drop the packet and close the connection.

If encryption is negotiated between the client and server control channel, no further validation occurs.

An e-mail client logging in from a nonsecure location may need to use encryption for authentication. For information about secure logins, see the description of the **secure-login** keyword of the **ip inspect name** command.

Inspection

Context Based Access Control (CBAC) inspects traffic that travels through the firewall to discover and manage state information for TCP and User Datagram Protocol (UDP) sessions. This state information is used to create temporary openings in the firewall's access lists to allow return traffic and additional data connections for permissible sessions.

Inspecting packets at the application layer and maintaining TCP and UDP session information provides CBAC with the ability to detect and prevent certain types of network attacks such as SYN-flooding. A SYN-flood attack occurs when a network attacker floods a server with a barrage of requests for connection and does not complete the connection. The resulting volume of half-open connections can overwhelm the server, causing it to deny service to valid requests. Network attacks that deny access to a network device are called denial-of-service (DoS) attacks.

POP3

The Post Office Protocol, Version 3 (POP3) is used to receive e-mail that is stored on a mail server. Unlike IMAP, POP only retrieves mail from a remote host.

POP3 works best when there is only one computer because it supports "offline" message access where messages are downloaded and then deleted from the mail server. This mode of access is not compatible with access from multiple computers because it tends to sprinkle messages across all the computers used for mail access.

With POP3-based e-mail clients, messages are downloaded to the user's local message store and can also be deleted from the mail server. Deletion is optional in most clients. When a new voice message arrives, the subscriber's only immediate notification is the activation of the MWI on the phone. New messages are displayed in the Inbox only after the client's local message store is updated with the Exchange message store. After the subscriber downloads new messages, the message state automatically changes from "new" to "read" on the server, even though the subscriber has not actually listened to the voice messages. MWIs on the subscriber's phone are extinguished, and the message state between the TUI and the subscriber's Inbox are not synchronized.

IMAP Protocol

The Internet Message Access Protocol (IMAP) is a method of accessing electronic mail or bulletin board messages that are kept on a mail server that may be shared. It permits a "client" e-mail program to access remote messages as though they were local. For example, e-mail stored on an IMAP server can be retrieved, sent, and managed from a desktop computer at home, from a workstation at the office, or from a laptop without transferring messages or files back and forth between the computers.

Only the message header and sender information are displayed in the Inbox until the user downloads the entire message, including attachments, from the server. When a new voice message arrives, the subscriber's only immediate notification is the activation of the Message Waiting Indication (MWI) on the phone. New messages are displayed in the Inbox only after the client's local message store is updated with the Exchange message store. When the subscriber listens to a new message by using the telephone user interface (TUI), the MWI is extinguished. In this case again, the message state is not updated in the Inbox until the client's message store is refreshed. However, if the subscriber uses an installed multimedia player to listen to the WaveForm Audio (WAV) attachment from the e-mail client's Inbox, message state changes are automatically synchronized with the TUI.

How message state changes are conveyed to the Cisco Unity subscriber, and how these changes are synchronized with the TUI, depend on whether the subscriber's e-mail client is configured to use POP3 or IMAP4 to access Exchange.

Client Command Validation

The Cisco IOS Firewall authenticates an e-mail client accessing an IMAP or POP3 server before allowing complete access into the server. The firewall searches the IMAP/POP3 TCP stream for valid protocol commands. If the client's commands are outside the protocol's definition, the Cisco IOS Firewall drops the packets and resets the connection.

Client command validation is typically needed in a DeMilitarized Zone (DMZ). Client access is allowed into the DMZ only if the e-mail server validates the user authentication. After the client is authenticated, the client becomes a trusted user and access is permitted.

SMTP

The Simple Mail Transfer Protocol (SMTP) is used to transfer e-mail between servers and clients on the Internet. E-mail clients and mail servers that use protocols other than Message Application Programming Interface (MAPI) can use the SMTP protocol to transfer a message from a client to the server, and then forward it to a message recipient's server. To retrieve, send, and manage these messages from the e-mail client use POP3 or IMAP4.

Cisco Unity uses SMTP to route voice messages via the Internet Voice Connector (IVC) gateway between other Exchange servers that are not connected by using a Site Message Connector. There is an IVC gateway on either end of the SMTP connection between Exchange servers. This ensures that MAPI message attributes survive the outbound transit between SMTP connections. It also ensures that the MIME-encoded attributes survive the inbound transit, and are included with the message stored in the Exchange message store.

SSL

The Secure Socket Layer (SSL) protocol is the standard protocol that delivers secure content over the Internet. It is a point-to-point security protocol that secures communication between a client and a server. SSL usually does not require a special client (that is, a Web browser often will suffice) and it does not require any additional operating system software.

SSL includes client and server authentication and data encryption for a limited set of applications (for example, the Web, e-mail, news, and file transfer). SSL is useful for securing e-commerce transactions over the Internet, and the protocol is well suited for extranets and remote access because it is relatively simple to deploy.

How to Configure E-mail Inspection Engine

This section contains the following procedures:

- Configuring Firewall Inspection of POP3 or IMAP E-mail, page 535 (required)
- Verifying the E-mail Inspection Engine Configuration, page 536 (optional)

Configuring Firewall Inspection of POP3 or IMAP E-mail

To allow the Cisco IOS Firewall to inspect POP3 or IMAP e-mail, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip inspect name** *inspection-name protocol* [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}][**reset**] [**secure-login**] [**timeout** *seconds*]
4. **interface** *type slot/port*
5. **ip inspect name** *inspection-name* {**in** | **out**}
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip inspect name** *inspection-name protocol* [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}][**reset**] [**secure-login**] [**timeout** *seconds*] **Example:** `Router(config)# ip inspect name mail-guard pop3`	Defines a set of inspection rules.
Step 4	**interface** *type slot/port* **Example:** `Router(config-if)# interface 1/0`	Configures an interface type.
Step 5	**ip inspect name** *inspection-name* {**in** | **out**} **Example:** `Router(config-if)# ip inspect name mail-guard in`	Enables the Cisco IOS Firewall on an interface.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

Verifying the E-mail Inspection Engine Configuration

To verify the E-mail Inspection Engine configuration, perform the following steps.

SUMMARY STEPS

1. **debug ip inspect imap**
2. **debug ip inspect pop3**
3. **show ip inspect** {**name** *inspection-name* | **config** | **interfaces** | **session** [**detail**] | **all**}

DETAILED STEPS

Step 1 **debug ip inspect imap**

Use this command to display messages about Cisco IOS Firewall events related to IMAP protocol e-mail messages.

```
Router# debug ip inspect imap
```

Step 2 **debug ip inspect pop3**

Use this command to display messages about Cisco IOS Firewall events related to POP3 protocol e-mail messages.

```
Router# debug ip inspect pop3
```

Step 3 **show ip inspect** {**name** *inspection-name* | **config** | **interfaces** | **session** [**detail**] | **all**}

Use this command to view CBAC configuration and session information.

```
Router# show ip inspect

Session audit trail is disabled
one-minute (sampling period) thresholds are [400:500] connections
max-incomplete sessions thresholds are [400:500]
max-incomplete tcp connections per host is 50. Block-time 0 minute.
tcp synwait-time is 30 sec -- tcp finwait-time is 5 sec
tcp idle-time is 3600 sec -- udp idle-time is 30 sec
dns-timeout is 5 sec
Inspection Rule Configuration
 Inspection name mail-guard
  tcp timeout 3600
  tdp timeout 30
  ftp timeout 3600
```

Configuration Examples for E-mail Inspection Engine

- Configuring IMAP and POP3 Protocol E-mail: Example, page 537

Configuring IMAP and POP3 Protocol E-mail: Example

The following example configures the Cisco IOS Firewall inspection of IMAP and POP3 protocol e-mail:

```
configure terminal
ip inspect name mail-guard pop3
ip inspect name mail-guard imap
exit
```

The following commands enable this functionality on an interface:

```
configure terminal
interface 1/0
ip inspect name mail-guard in
exit
```

Additional References

The following sections provide references related to E-Mail Inspection Engine.

Related Documents

Related Topic	Document Title
IMAP and POP3	White Paper: *Deploying Cisco Unity in Diverse Messaging Environments (All Versions with Microsoft Exchange)*
CBAC	*Cisco IOS Security Configuration Guide*, Release 12.3 *Cisco IOS Security Command Reference*, Release 12.3T

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 1939	J Myers and M. Rose, "Post Office Protocol, Version 3 (POP3)," May 1996.
RFC 3501	M. Crispin, *"Internet Message Access Protocol (IMAP4rev1,"* March 2003.

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **debug ip inspect**
- **ip inspect name**
- **show ip inspect**

Glossary

authentication—Process during which any unrecognized command causes the Cisco IOS Firewall to drop the packet and close the connection.

CBAC—Context-Based Access Control. A Cisco IOS Firewall set feature that scrutinizes source and destination addresses to enhance security for TCP and UDP applications that use well-known ports, such as FTP and e-mail traffic.

ESMTP—Extended Simple Mail Transfer Protocol. An extended version of the Simple Mail Transfer Protocol (SMTP), which includes additional functionality, such as delivery notification and session delivery.

IMAP—Internet Message Access Protocol. A method of accessing e-mail or bulletin board messages kept on a mail server that can be shared. IMAP permits client e-mail applications to access remote message stores as if they were local without actually transferring the message.

POP—Post Office Protocol. A protocol that client e-mail applications use to retrieve mail from a mail server.

SMTP—Simple Mail Transfer Protocol. An Internet protocol providing e-mail services.

SSL—Secure Socket Layer Protocol. This protocol is used to deliver secure information over the Internet.

state router—A router that tracks the client/server commands until the client is authenticated.

TCP—Transmission Control Protocol. A connection-oriented transport layer protocol that provides reliable full-duplex data transmission. TCP is part of the TCP/IP protocol stack.

UDP—User Datagram Protocol. A connectionless transport-layer protocol for exchanging datagrams without acknowledgments or guaranteed delivery.

VPN—Virtual Private Network. A network that enables IP traffic to travel securely over a public TCP/IP network by encrypting all traffic from one network to another. A VPN network uses “tunneling” to encrypt all information at the IP level.

Note Refer to *Internetworking Terms and Acronyms* for terms not included in this glossary.

ESMTP Support for Cisco IOS Firewall

The ESMTP Support for Cisco IOS Firewall feature enhances the Cisco IOS Firewall to support Extended Simple Mail Transport Protocol (ESMTP), allowing customers who install mail servers behind Cisco IOS firewalls to install their servers on the basis of ESMTP (instead of Simple Mail Transport Protocol [SMTP]).

Feature History for ESMTP Support for Cisco IOS Firewall

Release	Modification
12.3(7)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for ESMTP Support for Cisco IOS Firewall, page 541
- Information About ESMTP Support for Cisco IOS Firewall, page 542
- How to Configure a Firewall to Support ESMTP, page 546
- Configuration Examples for Firewall ESMTP Support, page 548
- Additional References, page 548
- Command Reference, page 550

Prerequisites for ESMTP Support for Cisco IOS Firewall

To enable this feature, your Cisco IOS image must contain the Cisco IOS firewall.

Information About ESMTP Support for Cisco IOS Firewall

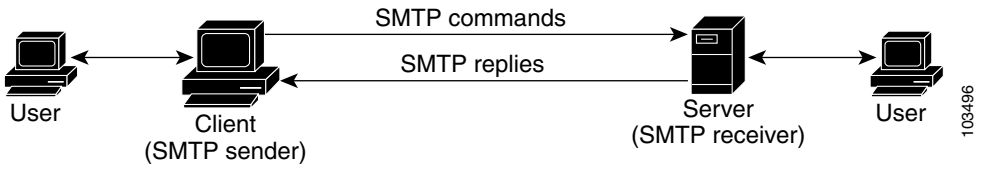
To configure a Cisco IOS firewall to inspect an ESMTP session and command sequence, you should understand the following concepts:

- SMTP Functionality Overview, page 542
- ESMTP Overview, page 543
- SMTP Firewall and ESMTP Firewall Comparison, page 543

SMTP Functionality Overview

SMTP inspection provides a basic method for exchanging e-mail messages. Figure 26 and the following steps outline a basic SMTP session.

Figure 26 Sample SMTP Exchange Topology



After a user sends an e-mail request to the client (the "SMTP sender"), the client established a TCP channel with the server (the "SMTP receiver"). Thereafter, the client and the server exchange SMTP commands and responses until the mail transaction is complete. The steps of typical SMTP transaction are as follows:

4. The client establishes a TCP connection with the server.
5. The client sends a HELO command with its domain name. If the server can accept mail from that domain name, it responds with a 250 reply code, which allows the client to continue with the mail transaction. (If the server does not respond with a 250 reply code, the client will send a QUIT command and terminate the TCP session.)
6. The client sends the MAIL command, indicating who initiated the mail. If the server accepts the mail, it responds with an OK reply. Then, the client sends the RCPT command, identifying the recipient of the mail. If the server accepts mail for the specified recipient, it responds with an OK reply; if the server cannot accept mail for the specified recipient, it rejects the recipient but not the entire transaction. (Several recipients can be negotiated.)
7. After the list of recipients has been negotiated between the client and the server, the client sends a DATA command. If the server is ready to receive data, it responds with a 354 reply code. If the server is not ready to receive data, it responds with a error reply, and the client terminates the transaction.
8. The client sends mail data ending with a special sequence. When the server sees the end of the message, it sends a 250 code reply.
9. The client sends a QUIT command, waits for the server to respond, then terminates the session.

ESMTP Overview

Like SMTP, ESMTP inspection provides a basic method for exchanging e-mail messages. Although an ESMTP session is similar to SMTP, there is one difference—the EHLO command.

After the TCP connection has been established between the client (the ESMTP sender) and the server (the ESMTP receiver), the client sends the EHLO command (instead of the HELO command that is used for SMTP). If the server does not support ESMTP, it sends a failure reply to the client because it did not recognize the EHLO command. If it supports ESMTP, the server responds with the code 250 and a list of extensions that the server supports. (Refer to RFC 1869 for an explanation of the extensions that your server may support.)

The server may send any of the following error codes if it supports ESMTP but is unable to function as normal:

- Error code 501—The server recognizes the EHLO command but is unable to accept it.
- Error code 502—The server recognizes the EHLO command but does not implement it.
- Error code 554—The server is unable to list the service extensions it supports.

If the client receives any of these error codes, it should issue the HELO command to revert to SMTP mode or issue the QUIT command to end the session.

After the client receives a successful response to the EHLO command, it will work the same way as SMTP, except that the client may issue new extended commands, and it may add a few parameters to the MAIL FROM and REPT TO commands.

SMTP Firewall and ESMTP Firewall Comparison

Although a SMTP firewall and an ESMTP firewall support the same functionality—command inspection, session conversion, and Intrusion Detection System (IDS) detection—slight variations exist between the protocols. Table 27 explains the firewall functionality and protocol-specific differences.

Table 27 SMTP and ESMTP Firewalls Functionality Comparison

Functionality	SMTP Firewall Description	ESMTP Firewall Description
Command Inspection	The SMTP firewall inspects commands for illegal commands. Illegal commands found in a packet are modified to an "xxxx" pattern and forwarded to the server. This process causes the server to send a negative reply, forcing the client to issue a valid command. An illegal SMTP command is any command except the following: DATA, HELO, HELP, MAIL, NOOP, QUIT, RCPT, RSET, SAML, SEND, SOML, and VRFY. **Note** Prior to Cisco IOS Release 12.3(7)T, an SMTP firewall will reset the TCP connection upon detection of an illegal command. That is, an SMTP firewall no longer resets the TCP connection upon detecting an illegal command.	ESMTP command inspection is the same as SMTP command inspection, except that ESMTP supports three additional commands—AUTH, EHLO, and ETRN. An illegal ESMTP command is any command except the following: AUTH, DATA, EHLO, ETRN, HELO, HELP, MAIL, NOOP, QUIT, RCPT, RSET, SAML, SEND, SOML, and VRFY.
Parameter Inspection	Not applicable.	The ESMTP firewall inspects the following extensions by performing deeper command inspection: • Message Size Declaration (SIZE) • Remote Queue Processing Declaration (ETRN) • Binary MIME (BINARYMIME) • Command Pipelining • Authentication • Delivery Status Notification (DSN) • Enhanced Status Code (ENHANCEDSTATUSCODE) • 8bit-MIMEtransport (8BITMIME) **Note** All other extensions, including private extensions, are not supported.

Table 27 SMTP and ESMTP Firewalls Functionality Comparison (continued)

Functionality	SMTP Firewall Description	ESMTP Firewall Description
EHLO Reply Inspection	Not applicable.	The ESMTP firewall inspects the EHLO reply, which contains a list of SMTP extensions that the server supports. Any unsupported extension that is found in the server's reply will be replaced with the "XXXX" pattern, which labels that extension "private." Thus, the client will no longer use the unsupported extension.
ESMTP to SMTP Session Conversion	The SMTP firewall forces a client that initiates an ESMTP session to use SMTP. When a client attempts to initiate an ESMTP session by sending the ELHO command, the firewall treats the EHLO command as an illegal command and modified it to the "xxxx" pattern. This response causes the server to send a 5xx code reply, forcing the client to revert to SMTP mode. **Note** Prior to Cisco IOS Release 12.3(7)T, the firewall intercepts the EHLO command and changes it to the NOOP command. The server responds with a 250 code reply. The firewall intercepts the response and modifies it to 502 code reply, which tells the client that the EHLO command is not supported.	Not applicable (because EHLO is supported in ESMTP).
IDS Signature Detection	The SMTP and ESMTP firewalls scan for a set of hard-coded IDS signatures. There are 11 signatures—6 are hard coded in the firewall and are enabled by default. The other 5 signatures remain in the IDS code and are disabled by default.	

Table 27 SMTP and ESMTP Firewalls Functionality Comparison (continued)

Functionality	SMTP Firewall Description	ESMTP Firewall Description
Command Pipelining	Not available. (The client sends a command to the server and must wait for a reply before sending another command.)	An ESMTP firewall can inspect commands that are in the pipeline. That is, commands that are sent before a response is received are inspected.
Resetting a Connection	Both SMTP and ESMTP firewalls will always send a "5xx" error code and close the connection upon detection of an unsupported parameter or an IDS signature in a command. That is, the firewall sends an appropriate reply code and closes the connection with proper TCP closing sequence packets (such as FIN or FIN+ACK) so the client does not continually attempt to send the same message. **Note** Prior to Cisco IOS Release 12.3(7)T, an SMTP firewall will reset the TCP connection upon detection of an illegal command or IDs signature. This behavior causes the client to keep trying to send the same message for up to 4 days (which is when the original message is bounced back to the user).	

How to Configure a Firewall to Support ESMTP

This section contains the following procedures:

- Configuring a Firewall for ESMTP Inspection, page 546

Configuring a Firewall for ESMTP Inspection

Use this task to configure a Cisco IOS Firewall to inspect an ESMTP session and command sequence.

Restrictions

SMTP and ESMTP cannot exist simultaneously. If SMTP is already configured, an attempt to configure ESMTP will result in the error message, "%ESMTP cannot coexist with SMTP, please unconfigure SMTP and try again...." If ESMTP is already configured, an attempt to configure SMTP will result in the error message, "%SMTP cannot coexist with ESMTP, please unconfigure ESMTP and try again...."

The following example illustrates how the router will react if you attempt to configure both protocols:

```
Router(config)# ip inspect name mail-guard smtp
Router(config)# ip inspect name mail-guard esmtp
ESMTP cannot coexist with SMTP, please unconfigure SMTP and try again...
Router(config)# end
Router# show running-config
.
.
.
ip inspect name mail-guard smtp
.
.
.
```

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip inspect name** *inspection-name* {**smtp** | **esmtp**} [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**max-data** *number*] [**timeout** *seconds*]
4. **interface** *type number*
5. **ip inspect** *inspection-name* {**in** | **out**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip inspect name** *inspection-name* {**smtp** | **esmtp**} [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**max-data** *number*] [**timeout** *seconds*] **Example:** `Router(config)# ip inspect name test esmtp`	Configures inspection of a SMTP or an ESMTP session.
Step 4	**interface** *type number* **Example:** `Router(config)# interface ethernet0`	Configures an interface type and enters interface configuration mode.
Step 5	**ip inspect** *inspection-name* {**in** | **out**} **Example:** `Router(config-if)# ip inspect test in`	Applies an inspection rule to an interface.

Troubleshooting Tips

To view and verify the inspection configuration, status, or session information, you can use any of the following EXEC commands:

- **show ip inspect name** *inspection-name*—Shows a particular configured inspection rule.
- **show ip inspect session**—Shows existing sessions that are currently being tracked and inspected by the firewall.
- **show ip inspect all**—Shows all inspection configuration and all existing sessions that are currently being tracked and inspected by the firewall.

Alert Messages

The existing SMTP-related alert message will not change. This message is logged every time the firewall detects an illegal or unsupported command. The message format is as follows:

```
FW-3-SMTP_INVALID_COMMAND: Invalid SMTP command (%s) (total %d chars) from initiator
(%i:%d)
```

A new alert message is added. This message is logged whenever the firewall detects an illegal parameter in an SMTP command. The message includes the address and port of the sender as well as the illegal parameter. The message format is as follows:

```
FW-3-SMTP_INVALID_PARAMETER: Invalid SMTP parameter (%s) from initiator (%i:%d)
```

What to Do Next

To provide a record of network access through the firewall, including illegitimate access attempts, and inbound and outbound services, you should turn on logging and audit trail. For information on completing this task, refer to the section "Configuring Logging and Audit Trail" in the chapter "Configuring Context-Based Access Control" in the *Cisco IOS Security Configuration Guide*.

Configuration Examples for Firewall ESMTP Support

This section contains the following configuration example:

- ESMTP Inspection Configuration: Example, page 548

ESMTP Inspection Configuration: Example

The following example shows how to configure inspection of ESMTP traffic:

```
Router# configure terminal
Router(config)# ip inspect name mail-guard esmtp timeout 30
```

Additional References

The following sections provide references related to ESMTP Support for Cisco IOS Firewall.

Related Documents

Related Topic	Document Title
Cisco IOS Firewall configuration	The section "Traffic Filtering and Firewalls" in the *Cisco IOS Security Configuration Guide*
Cisco IOS Firewall commands	*Cisco IOS Security Command Reference*, Release 12.3 T

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 821	*Simple Mail Transfer Protocol*
RFC 1652	*SMTP Service Extension for 8bit-MIMEtransport*
RFC 1845	*SMTP Service Extension for Checkpoint/Restart*
RFC 1869	*SMTP Service Extensions*
RFC 1870	*SMTP Service Extension for Message Size Declaration*
RFC 1891	*SMTP Service Extension for Delivery Status Notifications*
RFC 1985	*SMTP Service Extension for Remote Message Queue Starting*
RFC 2034	*SMTP Service Extension for Returning Enhanced Error Codes*
RFC 2554	*SMTP Service Extension for Authentication*
RFC 2645	*ON-DEMAND MAIL RELAY (ODMR) SMTP with Dynamic IP Addresses*
RFC 2920	*SMTP Service Extension for Command Pipelining*
RFC 3030	*SMTP Service Extensions for Transmission of Large and Binary MIME Messages*
RFC 3207	*SMTP Service Extension for Secure SMTP over Transport Layer Security*

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following modified command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **ip inspect name**

Firewall ACL Bypass

The Firewall ACL Bypass feature allows a packet to avoid redundant access control list (ACL) checks by allowing the firewall to permit the packet on the basis of existing inspection sessions instead of dynamic ACLs. Thus, input and output dynamic ACLs searches are eliminated, improving the overall throughput performance of the base engine.

Feature History for Firewall ACL Bypass

Release	Modification
12.3(4)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Information About Firewall ACL Bypass, page 551
- How to Use Firewall ACL Bypass, page 552
- Configuration Examples for Verifying Firewall Session Information, page 552
- Additional References, page 554
- Command Reference, page 555
- Glossary, page 556

Information About Firewall ACL Bypass

To better understand how dynamic ACL bypass works, you should understand the following concepts:

- Benefits of Firewall ACL Bypass, page 552
- Firewall ACL Bypass Functionality Overview, page 552

Benefits of Firewall ACL Bypass

Because input and output dynamic ACLs are no longer necessary, the need for context-based access control (CBAC) to create dynamic ACLs on the interface is eliminated. Thus, the following benefits are now available:

- Improved connections per second performance of the firewall
- Reduced run-time memory consumption of the firewall

Firewall ACL Bypass Functionality Overview

Before ACL bypassing was implemented, a packet could be subjected to as many as three redundant searches—an input ACL search, an output ACL search, and an inspection session search. Each dynamic ACL that CBAC creates corresponds to a single inspection session. Thus, a matching dynamic ACL entry for a given packet implies that a matching inspection session exists and that the packet should be permitted through the ACL. Because a matching inspection session is often found in the beginning of IP processing, the input and output dynamic ACL searches are no longer necessary and can be eliminated.

ACL bypassing subjects the packet to one search—the inspection session search—during its processing path through the router. When a packet is subjected to a single inspection session search before the ACL checks, the packet is matched against the list of session identifiers that already exist on the interface. (Session identifiers keep track of the source and destination IP addresses and ports of the packets and on which interface the packet arrived.)

Note Session identifiers are not created on interfaces for inspection sessions that are only Intrusion Detection Sessions (IDS).

How to Use Firewall ACL Bypass

After your firewall is configured for inspection, ACL bypassing is performed by default. That is, you should configure inspection as normal.

To configure CBAC for your firewall, see the following chapter “Configuring Context-Based Access Control” in the *Cisco IOS Security Configuration Guide*.

Configuration Examples for Verifying Firewall Session Information

After you have configured your firewall for inspection, you can use the **show ip inspect sessions detail** command to view session inspection information. The following examples show how eliminating dynamic ACLs changes the sample output:

- Old show ip inspect CLI Output: Example, page 553
- New show ip inspect CLI Output: Example, page 553

Old show ip inspect CLI Output: Example

The following is sample output from the **show ip inspect session detail** command, which shows that an outgoing ACL and an inbound ACL (dynamic ACLs) have been created to allow return traffic:

```
Router# show ip inspect session detail

Established Sessions
 Session 80E87274 (192.168.1.116:32956)=>(192.168.101.115:23) tcp SIS_OPEN
   Created 00:00:08, Last heard 00:00:04
   Bytes sent (initiator:responder) [140:298] acl created 2
   Outgoing access-list 102 applied to interface FastEthernet0/0
   Inbound access-list 101 applied to interface FastEthernet0/1

Router# show access-lists

Extended IP access list 101
    permit tcp host 192.168.101.115 eq telnet host 192.168.1.116 eq 32956 (27 matches)
    deny udp any any
    deny tcp any any
    permit ip any any
Extended IP access list 102
    permit tcp host 192.168.101.115 eq telnet host 192.168.1.116 eq 32956 (27 matches)
    deny udp any any
    deny tcp any any
    permit ip any any
```

New show ip inspect CLI Output: Example

The following is sample output from the **show ip inspect session detail** command, which shows related ACL information (such as session identifiers [SID]), but does not show dynamic ACLs, which are no longer created:

```
Router# show ip inspect session detail

Established Sessions
 Session 814063CC (192.168.1.116:32955)=>(192.168.101.115:23) tcp SIS_OPEN
 Created 00:00:10, Last heard 00:00:06
 Bytes sent (initiator:responder) [140:298]
 In  SID 192.168.101.115[23:23]=>192.168.1.117[32955:32955] on ACL 101 (15 matches)
 Out SID 192.168.101.115[23:23]=>192.168.1.116[32955:32955] on ACL 102

Router# show access-list

Extended IP access list 101
    deny udp any any (20229 matches)
    deny tcp any any
    permit ip any any (6 matches)
Extended IP access list 102
    deny udp any any
    deny tcp any any
    permit ip any any (1 match)
```

Additional References

The following sections provide references related to Dynamic ACL Bypass.

Related Documents

Related Topic	Document Title
Cisco IOS Firewalls and ACLs	The section "Traffic Filtering and Firewalls" in the *Cisco IOS Security Configuration Guide*
Cisco IOS Firewall commands	*Cisco IOS Security Command Reference*, Release 12.3 T

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following modified command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **show ip inspect**

Glossary

connections per second— Metric defined by the number of short-lived connections that can be created and deleted within 1 second by a router running CBAC. (These connections apply only to TCP connections because UDP is a connectionless protocol.)

throughput—Metric defined by the number of packets transferred from the input interface to the output interface within 1 second by a router running CBAC.

Note Refer to *Internetworking Terms and Acronyms* for terms not included in this glossary.

Firewall N2H2 Support

The Firewall N2H2 Support feature provides users with an additional option when choosing the URL filter vendor. Just like the Websense URL filtering server, N2H2 interacts with your Cisco IOS firewall (also known as Cisco Secure Integrated Software [CSIS]) to allow you to prevent users from accessing specified websites on the basis of some policy. The Cisco IOS firewall works with the N2H2 Internet Filtering Protocol (IFP) server to know whether a particular URL should be allowed or denied (blocked).

Feature Specifications for the Firewall N2H2 Support feature

Feature History	
Release	**Modification**
12.2(11)YU	This feature was introduced.
12.2(15)T	This feature was integrated into Cisco IOS Release 12.2(15)T.
Supported Platforms	
For platforms supported in Cisco IOS Release 12.2(11)YU and 12.2(15)T, consult Cisco Feature Navigator.	

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Restrictions for Firewall N2H2 Support, page 558
- Information About Cisco N2H2 Support, page 558
- How to Configure N2H2 URL Support, page 561
- Configuration Examples for Firewall and Webserver, page 567
- Additional References, page 572
- Command Reference, page 573
- Glossary, page 575

Restrictions for Firewall N2H2 Support

N2H2 IFP (Server) Requirement

To enable this feature, you must have at least one N2H2 server; however, two or more N2H2 servers are preferred. Although there is no limit to the number of N2H2 servers you may have, and you can configure as many servers as you wish, only one server will be active at any given time—the primary server. URL lookup requests will be sent only to the primary server.

URL Filtering Support Restriction

This feature supports only one active URL filtering scheme at a time. (Before enabling N2H2 URL filtering, you should always ensure that there is not another URL filtering scheme configured, such as Websense.)

Username Restriction

N2H2 requires the username to be supplied with the URL lookup request. Thus, the user-based policy will not work with N2H2 because the current Cisco IOS software does not retrieve the username.

Protocol Used to Communicate Between Firewall and N2H2 Server Restriction

TCP is currently the only protocol used to communicate between the Cisco IOS firewall (UNIX FileSystem [UFS]) and the N2H2 server.

Information About Cisco N2H2 Support

To configure Firewall N2H2 support, you must understand the following concepts:

- Benefits of Firewall N2H2 Support, page 558
- Feature Design of Firewall N2H2 Support, page 560
- Supported N2H2 Filtering Methods, page 561

Benefits of Firewall N2H2 Support

The Cisco IOS Firewall N2H2 Support feature provides an Internet management application that allows you to control web traffic for a given host or user on the basis of a specified security policy. In addition, the following functions are available in this feature:

Primary and Secondary Servers

When users configure multiple N2H2 servers, the firewall will use only one server at a time—the primary server; all other servers are called secondary servers. When the primary server becomes unavailable for any reason, it becomes a secondary server and one of the secondary servers becomes the primary server.

A firewall marks a primary server as down when sending a request to or receiving a response from the server fails. When a primary server goes down, the system will go to the beginning of the configured servers list and try to activate the first server on the list. If the first server on the list is unavailable, it will try the second server on the list; the system will keep trying to activate a server until it is successful or until it reaches the end of the server list. If the system reaches the end of the server list, it will set a flag indicating that all of the servers are down, and it will enter allowmode.

When all the servers are down and the system is in allow mode, a periodic event that occurs for each minute will trace through the server list, trying to bring up a server by opening a TCP connection. If the TCP connection is successfully opened, the server is considered to be up, and the system will return to operational mode.

IP Cache Table

This function provides an IP cache table that contains the IP addresses of web servers whose underlying URLs can be accessed by all users and hosts.

The caching algorithm involves three parameters—the maximum number of IP addresses that can be cached, an idle time, and an absolute time. The algorithm also involves two timers—idle timer and absolute timer. The idle timer is a small periodic timer (1 minute) that checks to see whether the number of cached IP addresses in the cache table exceeds 80 percent of the maximum limit. If the cached IP addresses have exceeded 80 percent, it will start removing idle entries; if it has not exceeded 80 percent, it will quit and wait for the next cycle. The absolute timer is a large periodic timer (1 hour) that is used to remove all of the elapsed entries. (The age of an elapsed entry is greater than the absolute time.) An elapsed entry will also be removed during cache lookup.

The idle time value is fixed at 10 minutes. The absolute time value is taken from the N2H2 lookup response, which is often greater than 15 hours. The absolute value for cache entries made out of exclusive-domains is 12 hours. The maximum number of cache entries is configurable.

To configure cache table parameters, use the **ip urlfilter cache** command.

Packet Buffering

This function allows you to increase the maximum number of HTTP responses that a Cisco IOS firewall can hold. If the HTTP responses arrive prior to an N2H2 server reply, this buffering scheme allows your firewall to store a maximum of 200 HTTP responses. (After 200 responses have been reached, the firewall will drop further responses.) The responses will remain in the buffer until an allow or deny message is received from N2H2: if the status indicates that the URL is allowed, the firewall will release the HTTP responses in the buffer to the browser of the end user; if the status indicates that the URL is blocked, the firewall will discard the HTTP responses in the buffer and close the connection to both ends. This function prevents numerous HTTP responses from overwhelming your system.

To configure the maximum number of HTTP responses for your firewall, use the **ip urlfilter max-resp-pak** command.

Exclusive Domains

This function provides a configurable list of domain names so that the Cisco IOS firewall does not have to send a lookup request to the N2H2 server for the HTTP traffic that is destined for one of the domains in the exclusive list. Thus, the N2H2 server does not have to deal with look-up requests for HTTP traffic that is destined for a host that has already been marked as "allowed."

Flexibility when entering domain names is also provided; that is, the user can enter the complete domain name or a partial domain name. If the user adds a complete domain name such as "www.cisco.com" to the exclusive domain list, all HTTP traffic whose URLs are destined for this domain (such as www.cisco.com/news and www.cisco.com/index) will be excluded from the N2H2 URL filtering policies and, on the basis of the configuration, the URLs will be permitted or blocked (denied).

If the user adds only a partial domain name to the exclusive domain list, such as ".cisco.com," all URLs whose domain names end with this partial domain name (such as www.cisco.com/products and www.cisco.com/eng) will be excluded from the N2H2 URL filtering policies and, based upon the configuration, the URLs will be permitted or blocked (denied).

To configure an exclusive domain list, use the **ip urlfilter exclusive-domain** command.

Allow Mode

The system will go into allow mode when connections to all the N2H2 servers are down. The system will return to normal mode when a connection to at least one web N2H2 server is up. Allow mode directs your system to forward or drop all packets on the basis of the configurable allow mode setting. By default, allow mode is off, so all HTTP requests are forbidden if all N2H2 servers are down.

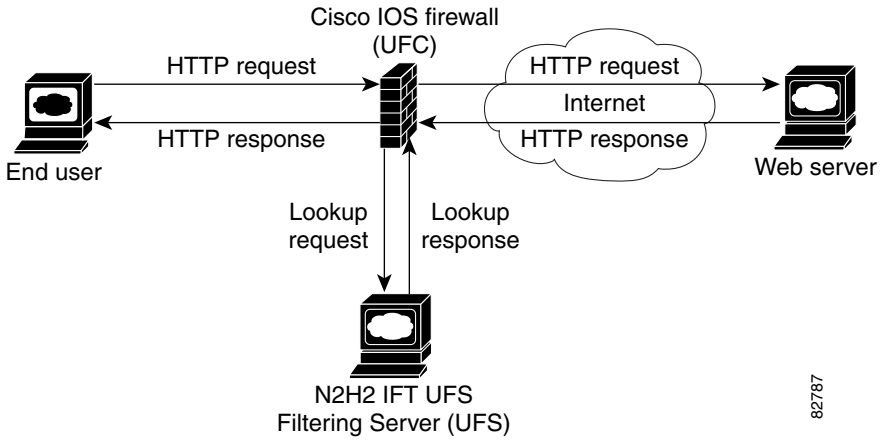
To configure allow mode for your system, use the **ip urlfilter allowmode** command.

Feature Design of Firewall N2H2 Support

Note This feature assumes that the N2H2 server will be part of a protected network and that requests from the Cisco IOS firewall will not travel over any unprotected network to reach the N2H2 server.

Figure 27 and the corresponding steps explain a sample URL filtering network topology.

Figure 27 Cisco IOS Firewall N2H2 URL Filtering Sample Topology



1. The end user browses a page on the web server, and the browser sends an HTTP request.
2. After the Cisco IOS firewall receives this request, it forwards the request to the web server, while simultaneously extracting the URL and sending a look-up request to the N2H2 server.
3. After the N2H2 server receives the look-up request, it checks its database to see whether it should permit or deny the URL; it returns a permit or deny status via a look-up response to the Cisco IOS firewall.
4. After the Cisco IOS Firewall receives this look-up response, it performs one of the following functions:

- If the look-up response permits the URL, it sends the HTTP response to the end user.
- If the look-up response denies the URL, the N2H2 server redirects the user to its own internal web server, which displays a message that describes the category under which the URL is blocked; thereafter, the connection is reset to both ends.

Supported N2H2 Filtering Methods

The Cisco IOS firewall supports most of the filtering methods that are supported by the N2H2 server. Table 28 lists N2H2 filtering methods and identifies which methods are supported by Cisco.

Table 28 N2H2 Filtering Methods Supported on Cisco IOS Firewall

N2H2 Filtering Method	Description	Supported by Cisco IOS Firewall?
Client-IP-based filtering	Filtering is applied to specified client IP addresses	Yes
Global filtering	Filtering is applied to all users, groups, and IP addresses	Yes
User-based filtering	Filtering is applied to a specified user	No

How to Configure N2H2 URL Support

To configure your Cisco IOS firewall to interact with at least one N2H2 server to provide URL filtering, configure the following procedures:

- Configuring Cisco IOS Firewall N2H2 URL Filtering, page 561 (required)
- Verifying Firewall and N2H2 URL Filtering, page 566 (optional)
- Maintaining the Cache Table, page 566 (optional)
- Monitoring the URL Filter Subsytems, page 567 (optional)

Configuring Cisco IOS Firewall N2H2 URL Filtering

N2H2 is based on a pass-through filtering technology, which is the most accurate, reliable, and scalable method of Internet filtering. Pass-through filtering requires all requests for web pages to pass through an Internet control point, such as a firewall, proxy server, or caching device. N2H2 is integrated with these control points and checks each request to determine whether it should be allowed or denied. All responses are logged for reporting purposes.

Prerequisites

- Before enabling N2H2 URL filtering, you should always ensure that there is not another URL filtering scheme configured, such as Websense. If you try to enter a new filtering scheme when one already exists, the new scheme will be ignored, and the system will display an error message that says, “different URL filtering scheme cannot co-exist.”
- URL filtering does not have an interface-specific command. It relies on Cisco IOS firewall C HTTP inspection to classify the traffic that needs filtering. This makes the configuration of Cisco IOS firewall inspection mandatory for the URL filtering feature to work. For more details on Cisco IOS firewall configuration, refer to the chapter “Cisco IOS Firewall Overview” in the IOS IOS Security Configuration Guide, Release 12.2.

Restrictions

Enabling HTTP inspection (via the **ip inspect name** command) triggers the Java applet scanner, which is very CPU intensive. The only way to stop the Java applet scanner is to specify the **java-list** *access-list* option and configure a standard access-list to allow any traffic. Configuring URL filtering without enabling the **java-list** *access-list* option will severely impact performance.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip inspect name** *inspection-name* **http** [**urlfilter**] [**java-list** *access-list*] [**alert** {**on** | **off**}] [**timeout** *seconds*] [**audit-trail** {**on** | **off**}]
4. **ip urlfilter server vendor** {**websense** | **n2h2**} *ip-address* [**port** *port-number*] [**timeout** *seconds*] [**retransmit** *number*]
5. **ip urlfilter alert**
6. **ip urlfilter audit-trail**
7. **ip urlfilter urlf-server-log**
8. **ip urlfilter exclusive-domain** {**permit** | **deny**} *domain-name*
9. **ip urlfilter cache** *number*
10. **ip urlfilter allowmode** [**on** | **off**]
11. **ip urlfilter max-resp-pak** *number*
12. **ip urlfilter max-request** *number*
13. **interface** *type slot*/*port*
14. **ip inspect inspection-name** {**in** | **out**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip inspect name** *inspection-name* **http** [**urlfilter**] [**java-list** *access-list*] [**alert** {**on** | **off**}] [**timeout** *seconds*] [**audit-trail** {**on** | **off**}] **Example:** `Router(config)# ip inspect name fw_urlf http urlfilter java-list 51 timeout 30`	Turns on HTTP inspection. The **urlfilter** keyword associates URL filtering with HTTP inspection. **Note** You may configure two or more inspections in a router, but URL filtering will work only with the inspections in which the **urlfilter** keyword is enabled. **Note** Enabling HTTP inspection with or without any options triggers the Java applet scanner, which is CPU intensive. The only way to stop the Java applet scanner is to specify the **java-list** *access-list* option. Configuring URL filtering without enabling the **java-list** *access-list* option will severely impact performance.
Step 4	**ip urlfilter server vendor** {**websense** | **n2h2**} *ip-address* [**port** *port-number*] [**timeout** *seconds*] [**retransmit** *number*] **Example:** `Router(config)# ip urlfilter server vendor websense 10.201.6.202`	Configures an N2H2 server to interact with the firewall to filter HTTP requests based on a specified policy. • *ip-address*—IP address of the vendor server. • **port** *port-number*—Port number that the vendor server listens on. The default port number is 4005. • **timeout** *seconds*—Length of time that the firewall will wait for a response from the vendor server. The default timeout is 5 minutes. • **retransmit** *number*—Number of times the firewall will retransmit the request when a response does not arrive. The default value is 2 times.
Step 5	**ip urlfilter alert** **Example:** `Router(config)# ip urlfilter alert`	(Optional) Enables the system alert, which displays system messages such as a server entering allow mode or going down. • The system alert is enabled by default.
Step 6	**ip urlfilter audit-trail** **Example:** `Router(config)# ip urlfilter audit-trail`	(Optional) Enables the logging of messages into the syslog server of router. • This function is disabled by default.
Step 7	**ip urlfilter urlf-server-log** **Example:** `Router(config)# ip urlfilter urlf-server-log`	(Optional) Enables the logging of system messages on the URL filtering server (the N2H2 server). This function is disabled by default.

	Command or Action	Purpose
Step 8	**ip urlfilter exclusive-domain** {**permit** | **deny**} *domain-name* **Example:** `Router(config)# ip urlfilter exclusive-domain permit www.cisco.com`	(Optional) Adds a domain name to or from the exclusive domain list so the firewall does not have to send look-up requests to the N2H2 server. • **permit**—Permits all traffic destined for the specified domain name. • **deny**—Denies all traffic destined for the specified domain name. • *domain-name*—Domain name that is added or removed from the exclusive domain list.
Step 9	**ip urlfilter cache** *number* **Example:** `Router(config)# ip urlfilter cache 4500`	(Optional) Configures cache table parameters. • *number*—Specifies the maximum number of destination IP addresses that can be cached into the cache table; the default is 5000.
Step 10	**ip urlfilter allowmode** [**on** | **off**] **Example:** `Router(config)# ip urlfilter allowmode on`	(Optional) Turns on the default mode of the filtering systems. • **on**—Allows HTTP requests to pass to the end user if all N2H2 servers are down. • **off**—Blocks all HTTP requests if all N2H2 servers are down; **off** is the default setting.
Step 11	**ip urlfilter max-resp-pak** *number* **Example:** `Router(config)# ip urlfilter max-resp-pak 150`	(Optional) Configures the maximum number of HTTP responses that the firewall can keep in its packet buffer. • The default value is 200. The maximum value is 20000, so you may set the **max-resp-pak** *number* to a value up to 20000.
Step 12	**ip urlfilter max-request** *number* **Example:** `Router(config)# ip urlfilter max-request 500`	(Optional) Sets the maximum number of outstanding requests that can exist at any given time. • The default value is 1000.
Step 13	**interface** *type slot/port* **Example:** `Router(config)# interface FastEthernet 0/0`	Configures an interface type and enters interface configuration mode
Step 14	**ip inspect inspection-name** {**in** | **out**} **Example:** `Router(config-if)# ip inspect inspection-name out`	Applies a set of inspection rules to an interface. • URL filtering is associated with inspection, and inspection is an interface-specific command. Hence, the **ip inspect** command needs to be configured on an interface.

Troubleshooting Tips

This feature introduces the following alert messages:

- "%URLF-3-SERVER_DOWN: Connection to the URL filter server 10.92.0.9 is down"

  This level three LOG_ERR-type message is displayed when a configured UFS goes down. When this happens, the firewall will mark the configured server as secondary, try to bring up one of the other secondary servers, and mark that server as the primary server. If there is no other server configured, the firewall will enter allow mode and display the "URLF-3-ALLOW_MODE" message.

- %URLF-3-ALLOW_MODE: Connection to all URL filter servers are down and ALLOW MODE is OFF

  This LOG_ERR type message is displayed when all UFSs are down and the system enters allow mode.

  Note Whenever the system goes into allow mode (all filter servers are down), a periodic keepalive timer will be triggered which will try to bring up a server by opening a TCP connection.

- "%URLF-5-SERVER_UP: Connection to an URL filter server 10.92.0.9 is made, the system is returning from ALLOW MODE"

  This LOG_NOTICE-type message is displayed when the UFSs are detected as being up and the system is returning from allow-mode.

- "%URLF-4-URL_TOO_LONG:URL too long (more than 3072 bytes), possibly a fake packet?"

  This LOG_WARNING-type message is displayed when the URL in a look-up request is too long; any URL longer than 3K will be dropped.

- "%URLF-4-MAX_REQ: The number of pending request exceeds the maximum limit <1000>"

  This LOG_WARNING-type message is displayed when the number of pending requests in the system exceeds the maximum limit and all further requests are dropped.

To display these alert messages, use the **ip urlfilter alert** command.

This feature introduces the following syslog messages:

- "%URLF-6-SITE_ALLOWED: Client 10.0.0.2:12543 accessed server 10.76.82.21:8080"

  This message is logged for each request whose destination IP address is found in the cache. It includes the source IP address, source port number, destination IP address, and destination port number. The URL is not logged because the IP address of the request is found in the cache, so parsing the request and extracting the URL is a waste of time.

- "%URLF-4-SITE-BLOCKED: Access denied for the site 'www.sports.com'; client 10.54.192.6:34557 server 172.24.50.12:80"

  This message is logged when a request finds a match against one of the blocked domains in the exclusive-domain list or the corresponding entry in the IP cache.

- "%URLF-6-URL_ALLOWED: Access allowed for URL http://www.n2h2.com/; client 10.54.192.6:54123 server 192.168.0.1:80"

  This message is logged for each URL request that is allowed by a UFS. It includes the allowed URL, source IP address, source port number, destination IP address, and destination port number. Longer URLs will be truncated to 300 bytes and logged.

- "%URLF-6-URL_BLOCKED: Access denied URL http://www.google.com; client 10.54.192.6:54678 server 172.19.14.2:80"

  This message is logged for each URL request that is blocked by a UFS. It includes the blocked URL, source IP address, source port number, destination IP address, and destination port number. Longer URLs will be truncated to 300 bytes and then logged.

To display these syslog messages, use the **ip urlfilter audit-trail** command.

Verifying Firewall and N2H2 URL Filtering

To verify that the Firewall N2H2 Support feature is working, perform any of the following optional steps:

SUMMARY STEPS

1. **enable**
2. **show ip urlfilter cache**
3. **show ip urlfilter config**

DETAILED STEPS

Command or Action	Purpose
enable **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
show ip urlfilter cache **Example:** `Router# show ip urlfilter cache`	Displays the destination IP addresses that are cached into the cache table.
show ip urlfilter config **Example:** `Router# show ip urlfilter config`	Displays the size of the cache, the maximum number of outstanding requests, the allow mode state, and the list of configured N2H2 servers.
show ip urlfilter statistics **Example:** `Router# show ip urlfilter statistics`	Displays information such as the number of requests that are sent to the N2H2 server, the number of responses received from the N2H2 server, the number pending requests in the system, the number of failed requests, the number of blocked URLs.

Maintaining the Cache Table

To clear the cache table of a specified or all IP addresses, perform the following optional steps:

SUMMARY STEPS

1. **enable**

2. **clear ip urlfilter cache**

DETAILED STEPS

Command or Action	Purpose
enable **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
clear ip urlfilter cache {*ip-address* | **all**} **Example:** `Router# clear ip urlfilter cache all`	Clears the cache table.

Monitoring the URL Filter Subsytems

To monitor the URL filter subsystems, perform the following optional steps:

SUMMARY STEPS

1. **enable**
2. **debug ip urlfilter** {**function-trace** | **detailed** | **events**}

DETAILED STEPS

Command or Action	Purpose
enable **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
debug ip urlfilter {**function-trace** | **detailed** | **events**} **Example:** `Router# debug ip urlfilter detailed`	Enables debugging information of URL filter subsystems. • **function-trace**—Prints a sequence of important functions that are called when configuring URL filtering. • **detailed**—Prints detailed information about various activities that occur during URL filtering. • **events**—Prints various events such as queue event, timer event, and socket event.

Configuration Examples for Firewall and Webserver

This section provides the following comprehensive configuration example:

- URL Filter Client (Firewall) Configuration Example, page 568

URL Filter Client (Firewall) Configuration Example

The following example shows how to configure the Cisco IOS firewall (also known as the URL filter client [UFC]) for N2H2 URL filtering:

```
Topology:

End User------LAN-----Fa0/0 -- Firewall -- S2/0----- Internet ---- Web Server
               |               Router
               |
N2H2           |
Server --------+


Router Configuration:

Example 1:

hostname fw9-7200b
!
!-----------------------------------------------------------------
! The following commands define the inspection rule "myfw," allowing
! the specified protocols to be inspected. Note that the "urlfilter"
! keyword entered for HTTP protocol enables URL filtering on HTTP
! traffic that are bound to this inspection.
!-----------------------------------------------------------------
!
ip inspect name myfw http urlfilter
ip inspect name myfw ftp
ip inspect name myfw smtp
ip inspect name myfw h323
!
!-------------------------------------------------------------------------
! The following command sets the URL filtering cache table size to 12000.
!-------------------------------------------------------------------------
ip urlfilter cache 12000
!
!-------------------------------------------------------------------
! The following commands configure three exclusive domains--
! two partial domains and one complete domain.
!-------------------------------------------------------------------
ip urlfilter exclusive-domain permit .weapons.com
ip urlfilter exclusive-domain deny .nbc.com
ip urlfilter exclusive-domain permit www.cisco.com
!
!-----------------------------------------------------------------
! The following two commands enable URL filtering Audit Trail and
! Alert messages.
!-----------------------------------------------------------------
ip urlfilter audit-trail
ip urlfilter alert
!
!-------------------------------------------------------------
! The command configures the N2H2 URL filtering server installed
! on 192.168.3.1.
!-------------------------------------------------------------
ip urlfilter server vendor n2h2 192.168.3.1
!
!-----------------------------------------------------------------------
! Create Access Control List 102:
! ACL 102 denies all IP protocol traffic except for ICMP traffic.
! This means that only the return traffic for protocols defined in the
! inspection rule and the ICMP traffic is allowed access through the
```

```
! interface where this rule is applied.
!
! Note that ACL is given here for an example; it is not relevant
! to the URL filtering. The URL filtering will work without ACL also.
!--------------------------------------------------------------------
!
access-list 102 permit icmp any any
access-list 102 deny tcp any any
access-list 102 deny udp any any
access-list 102 deny ip any any
!
!
interface FastEthernet0/0
ip address 192.168.3.254 255.255.255.0
ip nat inside
no ip route-cache
no ip mroute-cache
!
interface Ethernet1/0
no ip route-cache
no ip mroute-cache
shutdown
duplex half
!
interface Ethernet1/1
no ip address
no ip mroute-cache
shutdown
duplex half
!
!--------------------------------------------------------------------
! The ACL and CBAC inspection rules are applied to the Serial2/0 interface.
!  In this example, the ACL is applied IN, meaning that it applies to traffic
!  inbound from the internet. The CBAC inspection rule myfw is applied OUT,
!  meaning that CBAC inspects the traffic that goes out through the interface
!  and controls return traffic to the router for an existing connection.
!--------------------------------------------------------------------------
interface Serial2/0
ip address 10.6.9.7 255.255.0.0
ip access-group 102 in
ip nat outside
ip inspect myfw out
no ip directed-broadcast
no ip mroute-cache
!
ip nat inside source static 192.168.3.1 10.6.243.1
ip nat inside source static 192.168.3.2 10.6.243.2
ip nat inside source static 192.168.3.3 10.6.243.3
ip classless
ip route 192.168.0.30 255.255.255.255 10.6.0.1
!
!
line con 0
exec-timeout 0 0
stopbits 1
line aux 0
stopbits 1
line vty 0 4
password letmein
login
!
end
```

```
Example 2:

! In the above example, the CBAC can also be configured on the inbound
! FastEthernet0/0 interface as IN, in which case the CBAC inspects all
! the traffic that comes in on FastEthernet0/0 and controls return traffic
! that leaves out of this interface for an existing connection.

interface FastEthernet0/0
ip address 192.168.3.254 255.255.255.0
ip access-group 102 out
ip nat inside
ip inspect myfw in
no ip route-cache
no ip mroute-cache
!
!
hostname fw9-7200b
!
logging buffered 64000 debugging
enable secret 5 $1$qMOf$umPb75mb3sV27JpNbW//7.
!
clock timezone PST -8
clock summer-time PDT recurring
ip subnet-zero
ip cef
no ip domain lookup
!
ip inspect name test http urlfilter
ip urlfilter cache 5
ip urlfilter exclusive-domain permit .weapons.com
ip urlfilter exclusive-domain deny .nbc.com
ip urlfilter exclusive-domain permit www.cisco.com
ip urlfilter audit-trail
ip urlfilter alert
ip urlfilter server vendor n2h2 192.168.3.1
ip audit notify log
ip audit po max-events 100
ip port-map http port 8080
!
no voice hpi capture buffer
no voice hpi capture destination
!
mta receive maximum-recipients 0
!
interface FastEthernet0/0
 ip address 192.168.3.254 255.255.255.0
 ip access-group 101 out
 ip nat inside
 ip inspect test in
 no ip route-cache
 no ip mroute-cache
!
interface Ethernet1/0
 ip address 10.6.9.7 255.255.0.0
 ip nat outside
 no ip route-cache
 no ip mroute-cache
 duplex half
!
interface Ethernet1/1
 no ip address
 no ip mroute-cache
 shutdown
 duplex half
```

```
!
interface Ethernet1/2
 no ip address
 no ip mroute-cache
 shutdown
 duplex half
!
interface Ethernet1/3
 no ip address
 no ip mroute-cache
 shutdown
 duplex half
!
interface Serial2/0
 no ip address
 no ip mroute-cache
 shutdown
 dsu bandwidth 44210
 framing c-bit
 cablelength 10
 serial restart_delay 0
 fair-queue
!
ip nat pool devtest 10.6.243.21 10.6.243.220 netmask 255.255.0.0
ip nat inside source list 1 pool devtest
ip nat inside source static 192.168.3.1 10.6.243.1
ip nat inside source static 192.168.3.2 10.6.243.2
ip nat inside source static 192.168.3.3 10.6.243.3
ip classless
ip route 192.168.0.30 255.255.255.255 10.6.0.1
no ip http server
no ip http secure-server
!
ip pim bidir-enable
!
!
access-list 101 deny   tcp any any
access-list 101 deny   udp any any
access-list 101 permit ip any any
access-list 102 deny   tcp any any
access-list 102 deny   udp any any
access-list 102 permit ip any any
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
!
call rsvp-sync
!
!
mgcp profile default
!
dial-peer cor custom
!
!
!gatekeeper
 shutdown
!
!
line con 0
 exec-timeout 0 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
```

```
 password letmein
 login
!
exception core-file sisu-devtest/coredump/fw9-7200b.core
exception dump 192.168.0.1
no scheduler max-task-time
!
end
```

Additional References

For additional information related to the Firewall N2H2 Support feature, refer to the following references:

- Related Documents, page 572
- Standards, page 572
- MIBs, page 572
- RFCs, page 573
- Technical Assistance, page 573

Related Documents

Related Topic	Document Title
Websense URL filtering information	*Firewall Websense URL Filtering*, Cisco IOS Release 12.2(15)T feature module
Additional Cisco IOS firewall configuration tasks and information	The part "Traffic Filtering and Firewalls" in the *Cisco IOS Security Configuration Guide*, Release 12.2
Additional Cisco IOS firewall commands	The part "Traffic Filtering and Firewalls" in the *Cisco IOS Security Command Reference*, Release 12.2
Cisco IOS firewall configuration	The chapter "Cisco IOS Firewall Overview" in the *Cisco IOS Security Configuration Guide*, Release 12.2

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

RFCs[1]	Title
RFC 1945	*Hypertext Transfer Protocol — HTTP/1.0*
RFC 2616	*Hypertext Transfer Protocol — HTTP/1.1*

1. Not all supported RFCs are listed.

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

New Commands

- **clear ip urlfilter cache**
- **debug ip urlfilter**
- **ip urlfilter alert**
- **ip urlfilter allowmode**
- **ip urlfilter audit-trail**

- **ip urlfilter cache**
- **ip urlfilter exclusive-domain**
- **ip urlfilter max-request**
- **ip urlfilter max-resp-pak**
- **ip urlfilter server vendor**
- **ip urlfilter urlf-server-log**
- **show ip urlfilter cache**
- **show ip urlfilter config**
- **show ip urlfilter statistics**

Modified Command

- **ip inspect name**

Glossary

ACL—Access Control List.

CSIS—Cisco Secure Integrated Software. CSIS is a content-based firewall that currently inspects application data, checks for protocol conformance, extracts the relevant port information to create dynamic access list entries that successfully allows return traffic, and closes the ports at the end of the session.

ICMP—Internet Control Message Protocol. ICMP is a network layer Internet protocol that reports errors and provides other information relevant to IP packet processing. ICMP is documented in RFC 792.

UFC—URL filter client. UFC is a separate process that accepts URLs from CSIS, forwards the URL to the Websense server, and process the replies from the vendor server (Websense or N2H2).

UFS—URL filter server. UFS is a generic name given to the vendor server (Websense or N2H2), which processes URLs and decides whether to allow or deny web traffic based on a given policy.

Note Refer to the *Internetworking Terms and Acronyms* for terms not included in this glossary.

Firewall Stateful Inspection of ICMP

The Firewall Stateful Inspection of ICMP feature addresses the limitation of qualifying Internet Control Management Protocol (ICMP) messages into either a malicious or benign category by allowing the Cisco IOS firewall to use stateful inspection to "trust" ICMP messages that are generated within a private network and to permit the associated ICMP replies. Thus, network administrators can debug network issues by using ICMP without concern that possible intruders may enter the network.

Feature Specifications for the Firewall Stateful Inspection of ICMP feature

Feature History	
Release	**Modification**
12.2(11)YU	This feature was introduced.
12.2(15)T	This feature was integrated into Cisco IOS Release 12.2(15)T.
Supported Platforms	
For platforms supported in Cisco IOS Release 12.2(11)YU and 12.2(15)T, consult Cisco Feature Navigator.	

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Restrictions for Firewall Stateful Inspection of ICMP, page 578
- Information About Firewall Stateful Inspection of ICMP, page 578
- How to Use Firewall Stateful Inspection of ICMP, page 579
- Configuration Examples for Stateful Inspection of ICMP, page 581
- Additional References, page 583
- Command Reference, page 585
- Glossary, page 586

Restrictions for Firewall Stateful Inspection of ICMP

- To enable this feature, your Cisco IOS image must contain the Cisco IOS firewall.
- This feature does not work for the User Datagram Protocol (UDP) traceroute, in which UDP datagrams are sent instead of ICMP packets. The UDP traceroute is typically the default for UNIX systems. To use ICMP inspection with a UNIX host, use the "I" option with the traceroute command. This functionality will cause the UNIX host to generate ICMP traceroute packets, which will be inspected by the Cisco IOS firewall ICMP.

Information About Firewall Stateful Inspection of ICMP

The following section provides information about the Cisco IOS Firewall Stateful Inspection of ICMP feature:

- Feature Design of Firewall Stateful Inspection of ICMP, page 578

Feature Design of Firewall Stateful Inspection of ICMP

ICMP is used to report errors and information about a network. It is a useful tool for network administrators who are trying to debug network connectivity issues. Unfortunately, intruders can also use ICMP to discover the topology of a private network. To guard against a potential intruder, ICMP messages can be blocked from entering a private network; however, a network administrator may then be unable to debug the network. Although a Cisco IOS router can be configured using access lists to selectively allow certain ICMP messages through the router, the network administrator must still guess which messages are potentially malicious and which messages are benign. With the introduction of this feature, a user can now configure a Cisco IOS firewall for stateful inspection to "trust" that the ICMP messages are generated within the private network and to permit the associated ICMP replies.

Note Access lists can still be used to allow unsolicited error messages along with Cisco IOS firewall inspection. Access lists complement Cisco IOS firewall ICMP inspection.

Stateful inspection of ICMP packets is limited to the most common types of ICMP messages that are useful to network administrators who are trying to debug their networks. That is, ICMP messages that do not provide a valuable tool for the internal network administrator will not be allowed. For the Cisco IOS firewall-supported ICMP message request types, see Table 29.

Table 29 ICMP Packet Types Supported by CBAC

ICMP Packet Type	Name	Description
0	Echo Reply	Reply to Echo Request (Type 8)
3	Destination Unreachable	Possible reply to any request **Note** This packet is included because it is a possible response to any ICMP packet request.
8	Echo Request	Ping or traceroute request
11	Time Exceeded	Reply to any request if the time to live (TTL) packet is 0

Table 29 ICMP Packet Types Supported by CBAC

ICMP Packet Type	Name	Description
13	Timestamp Request	Request
14	Timestamp Reply	Reply to Timestamp Request (type 13)

Note ICMP packet types 0 and 8 are used for pinging: the source sends out an Echo Request packet, and the destination responds with an Echo Reply packet.

Packet types 0, 8, and 11 are used for ICMP traceroute: Echo Request packets are sent out starting with a TTL packet of 1, and the TTL is incremented for each hop. The intermediate hops respond to the Echo Request packet with a Time Exceeded packet; the final destination responds with an Echo Reply packet.

How to Use Firewall Stateful Inspection of ICMP

This section contains the following procedures:

- Configuring Firewall Stateful Inspection for ICMP, page 579
- Verifying Firewall and ICMP Session Information, page 580
- Monitoring Firewall and ICMP Session Information, page 581

Configuring Firewall Stateful Inspection for ICMP

To enable the Cisco IOS Firewall to start inspection ICMP messages, perform the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip inspect name** *inspection-name* **icmp** [**alert** {**on** | **off**}] [**audit-trail** {**on** |**off**}] [**timeout** *seconds*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip inspect name** *inspection-name* **icmp** [**alert** {**on** | **off**}] [**audit-trail** {**on** |**off**}] [**timeout** *seconds*] **Example:** `Router(config)# ip inspect name test icmp alert on audit-trail on timeout 30`	Turns on inspection for ICMP. • **alert**—Alert messages are generated. This function is **on** by default. • **audit-trail**—Audit trail messages are generated. This function is **off** by default. • **timeout**—Overrides the global channel inactivity timeout value. The default value of the *seconds* argument is 10.

Verifying Firewall and ICMP Session Information

To display active ICMP session and IP access list information, perform the following optional steps:

SUMMARY STEPS

1. **enable**
2. **show ip inspect session** [**detail**]
3. **show ip access-list**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip inspect session** [**detail**] **Example:** `Router# show ip inspect session`	(Optional) Displays existing sessions that are currently being tracked and inspected by the Cisco IOS firewall. • The optional **detail** keyword causes additional details about these sessions to be shown.
Step 3	**show ip access-list** **Example:** `Router# show ip access-list`	(Optional) Displays the contents of all current IP access lists. For a sample output example, see the section "ICMP Session Verification Example."

Monitoring Firewall and ICMP Session Information

To monitor debugging messages related to ICMP inspection, perform the following optional steps:

SUMMARY STEPS

1. **enable**
2. **debug ip inspect icmp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug ip inspect icmp** **Example:** `Router# debug ip inspect icmp`	(Optional) Displays the operations of the ICMP inspection engine for debugging purposes. For an example of sample output, see the command **debug ip inspect** in the Command Reference section.

Configuration Examples for Stateful Inspection of ICMP

This section provides the following configuration examples:

- Firewall Stateful Inspection for ICMP Configuration Example, page 582
- ICMP Session Verification Example, page 582

Firewall Stateful Inspection for ICMP Configuration Example

The default ICMP timeout is deliberately short (10 seconds) due to the security hole that is opened by allowing ICMP packets with a wild-carded source address back into the inside network. The timeout will occur 10 seconds after the last outgoing packet from the originating host. For example, if you send a set of 10 ping packets spaced 1 second apart, the timeout will expire in 20 seconds or 10 seconds after the last outgoing packet. However, the timeout is not extended for return packets. If a return packet is not seen within the timeout window, the hole will be closed and the return packet will not be allowed in. Although the default timeout can be made longer if desired, it is recommended that this value be kept relatively short.

The following example shows how to configure a firewall for stateful inspection of ICMP packets:

```
no service pad
service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname UUT
!
ip subnet-zero
no ip domain lookup
!
ip inspect audit-trail
ip inspect name test icmp alert on audit-trail on timeout 30
!
interface Ethernet0
ip address 192.168.10.2 255.255.255.0
ip inspect test in
!
interface Ethernet1
ip address 192.168.20.2 255.255.255.0
ip access-group 101 in
!
ip classless
ip route 0.0.0.0 0.0.0.0 192.168.20.3
no ip http server
!
access-list 101 deny ip any any
!
line con 0
exec-timeout 0 0
!
end
```

ICMP Session Verification Example

The following example is sample output from the **show ip access-list** command. In this example, Access Control Lists (ACLs) are created for an ICMP session on which only ping packets were issued from the host.

```
Router# show ip access-list 101

Extended IP access list 101
     permit icmp any host 192.168.133.3 time-exceeded
     permit icmp any host 192.168.133.3 unreachable
     permit icmp any host 192.168.133.3 timestamp-reply
     permit icmp any host 192.168.133.3 echo-reply (4 matches)
```

Additional References

For additional information related to Firewall Stateful Inspection of ICMP, refer to the following references:

- Related Documents, page 583
- Standards, page 583
- MIBs, page 584
- RFCs, page 584
- Technical Assistance, page 585

Related Documents

Related Topic	Document Title
CBAC information and configuration tasks	The chapter "Configuring Context-based Access Control" in the *Cisco IOS Security Configuration Guide*, Release 12.2
Additional CBAC commands	The chapter "Context-based Access Control Commands" in the *Cisco IOS Security Command Reference*, Release 12.2

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

RFCs[1]	Title
RFC 792	*Internet Control Message Protocol*
RFC 950	*Internet Standard Subnetting Procedure*
RFC 1700	*Assigned Numbers*

1. Not all supported RFCs are listed.

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **debug ip inspect**
- **ip inspect name**

Glossary

ACL—access control list. An ACL is a list kept by routers to control access to or from the router for a number of services (for example, to prevent packets with a certain IP address from leaving a particular interface on the router).

CBAC—Context-Based Access Control. CBAC is the name given to the Cisco IOS Firewall subsystem.

firewall—A firewall is a networking device that controls access to the network assets of your organization. Firewalls are positioned at the entrance points into your network. If your network has multiple entrance points, you must position a firewall at each point to provide effective network access control.

The most basic function of a firewall is to monitor and filter traffic. Firewalls can be simple or elaborate, depending on your network requirements. Simple firewalls are usually easier to configure and manage. However, you might require the flexibility of a more elaborate firewall.

ICMP—Internet Control Message Protocol. An ICMP is a network layer Internet protocol that reports errors and provides other information relevant to IP packet processing.

RPC—remote-procedure call. A RPC is the technological foundation of client or server computing. RPCs are procedure calls that are built or specified by clients and are executed on servers, with the results returned over the network to the clients.

RTSP—Real Time Streaming Protocol. RTSP enables the controlled delivery of real-time data, such as audio and video. Sources of data can include both live data feeds, such as live audio and video, and stored content, such as prerecorded events. RTSP is designed to work with established protocols, such as RTP and HTTP.

SIP—Session Initiation Protocol. SIP is a protocol developed by the IETF MUSIC Working Group as an alternative to H.323. SIP features are compliant with IETF RFC 2543, published in March 1999. SIP equips platforms to signal the setup of voice and multimedia calls over IP networks.

SMTP—simple mail transfer protocol. SMTP is an Internet protocol providing e-mail services.

UDP—User Datagram Protocol. A UDP is a connectionless transport layer protocol in the TCP/IP protocol stack. UDP is a simple protocol that exchanges datagrams without acknowledgments or guaranteed delivery, requiring that error processing and retransmission be handled by other protocols. UDP is defined in RFC 768.

Note Refer to the *Internetworking Terms and Acronyms* for terms not included in this glossary.

Firewall Support for SIP

The Firewall Support for SIP feature integrates Cisco IOS firewalls, Voice over IP (VoIP) protocol, and Session Initiation Protocol (SIP) within a Cisco IOS-based platform, enabling better network convergence.

Note Some Cisco IOS versions earlier than 12.2(11)YU and 12.2(15)T may accept the configuration commands for SIP that are shown in this document; however, those earlier versions will not function properly.

Feature Specifications for Firewall Support for SIP

Feature History	
Release	**Modification**
12.2(11)YU	This feature was introduced.
12.2(15)T	This feature was integrated into Cisco IOS Release 12.2(15)T.
Supported Platforms	
For platforms supported in Cisco IOS Releases 12.2(11)YU and 12.2(15)T, consult Cisco Feature Navigator.	

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Restrictions for Firewall Support for SIP, page 588
- Information About Firewall Support for SIP, page 588
- How to Configure Your Firewall for SIP, page 594
- Configuration Examples for Firewall SIP Support, page 597
- Additional References, page 597
- Command Reference, page 599

Restrictions for Firewall Support for SIP

DNS Name Resolution

Although SIP methods can have Domain Name System (DNS) names instead of raw IP addresses, this feature currently does not support DNS names.

SIP UDP Support Only

This feature supports only the SIP User Datagram Protocol (UDP) format for signaling; the TCP format is not supported.

SIP Abbreviated Header

This feature does not support the compact form of SIP header fields.

Earlier Versions of Cisco IOS

Some Cisco IOS versions earlier than 12.2(11)YU and 12.2(15)T may accept the configuration commands for SIP that are shown in this document; however, those earlier versions will not function properly.

Information About Firewall Support for SIP

To configure the Cisco IOS Firewall Support for SIP feature, you must understand the following concepts:

- Firewall and SIP Overviews, page 588
- Firewall for SIP Functionality Description, page 591
- SIP Message Treatment by the Firewall, page 592
- Call Database, page 593

Firewall and SIP Overviews

This section contains the following concepts:

- Cisco IOS Firewall, page 588
- SIP (Session Initiation Protocol), page 589

Cisco IOS Firewall

The Cisco IOS firewall extends the concept of static access control lists (ACLs) by introducing dynamic ACL entries that open on the basis of the necessary application ports on a specific application and close these ports at the end of the application session. The Cisco IOS firewall achieves this functionality by inspecting the application data, checking for conformance of the application protocol, extracting the relevant port information to create the dynamic ACL entries, and closing these ports at the end of the session. The Cisco IOS firewall is designed to easily allow a new application inspection whenever support is needed.

SIP (Session Initiation Protocol)

SIP is an ASCII-based, application-layer control protocol that can be used to establish, maintain, and terminate calls between two or more endpoints. Like other VoIP protocols, SIP is designed to address the functions of signaling and session management within a packet telephony network. Signaling allows call information to be carried across network boundaries. Session management provides the ability to control the attributes of an end-to-end call.

SIP Messages

SIP has two types of messages—requests and responses—that have the following generic structure:

```
generic-message   =   Request-Line | Status-Line
                      * ( general-header | request-header
                      | response-header | entity-header )
                      CRLF
                      [ message-body]
```

Note Any of these message components may contain embedded IP addresses.

Table 30 identifies the six available SIP request messages.

Table 30 SIP Request Messages

SIP Message	Purpose
ACK	Confirms receipt of a final response to INVITE
BYE	Is sent by either side to end the call
CANCEL	Is sent to end a call that has not yet been connected
INVITE	Is a request from a User Agent Client (UAC) to initiate a session
OPTIONS	Are sent to query capabilities of the user agents and network servers
REGISTER	Is sent by the client to register the address with a SIP proxy

Table 31 identifies the available SIP response methods.

Table 31 SIP Response Messages

SIP Message	Purpose
1xx Informational	• 100 = Trying • 180 = Ringing • 181 = Call Is Being Forwarded • 182 = Queued • 183 = Session Progress
2xx Successful	• 200 = OK

Table 31 SIP Response Messages (continued)

SIP Message	Purpose
3xx Redirection	• 300 = Multiple Choices • 301 = Moved Permanently • 302 = Moved Temporarily • 303 = See Other • 305 = Use Proxy • 380 = Alternative Service
4xx Request Failure	• 400 = Bad Request • 401 = Unauthorized • 402 = Payment Required • 403 = Forbidden • 404 = Not Found • 405 = Method Not Allowed • 406 = Not Acceptable • 407 = Proxy Authentication Required • 408 = Request Timeout • 409 = Conflict • 410 = Gone • 411 = Length Required • 413 = Request Entity Too Large • 414 = Request URI Too Large • 415 = Unsupported Media Type • 420 = Bad Extension • 480 = Temporarily Not Available • 481 = Call Leg/Transaction Does Not Exist
4xx Request Failure (continued)	• 482 = Loop Detected • 483 = Too Many Hops • 484 = Address Incomplete • 485 = Ambiguous • 486 - Busy Here

Table 31 SIP Response Messages (continued)

SIP Message	Purpose
5xx Server Failure	• 500 = Internal Server Error • 501 = Not Implemented • 502 = Bad Gateway • 503 = Service Unavailable • 504 = Gateway Timeout • 505 = SIP Version Not Supported
6xx Global Failure	• 600 = Busy Anywhere • 603 = Decline • 604 = Does Not Exist Anywhere • 606 = Not Acceptable

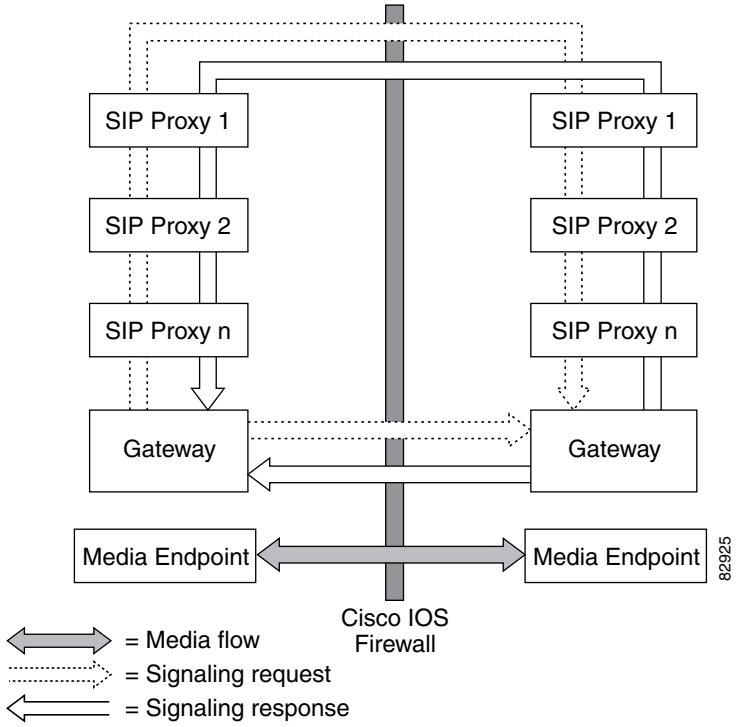
Firewall for SIP Functionality Description

The Firewall for SIP Support feature allows SIP signaling requests to traverse directly between gateways or through a series of proxies to the destination gateway or phone. After the initial request, if the Record-Route header field is not used, subsequent requests can traverse directly to the destination gateway address as specified in the Contact header field. Thus, the Cisco IOS firewall is aware of all surrounding proxies and gateways and allows the following functionality:

- SIP signaling responses can travel the same path as SIP signaling requests.
- Subsequent signaling requests can travel directly to the endpoint (destination gateway).
- Media endpoints can exchange data between each other.

See Figure 28 for a sample topology that displays these functionalities.

Figure 28 Cisco IOS Firewall for SIP Awareness Sample Topology



SIP Message Treatment by the Firewall

See Table 32 for information on the treatment of SIP methods by the Cisco IOS firewall.

Table 32 Treatment of SIP Methods by the Cisco IOS Firewall

SIP Message	Purpose
200 OK	Signifies the end of the call creation phase. The packet is checked for validity against the call database, and the contact information of the server is taken from it. Temporary call-flow–based openings in the firewall are created for allowing the BYE message, which can be initiated from the inside or outside.
200 OK for BYE	Signifies the graceful termination of the call and is in response to the BYE message. The same action as the CANCEL message is taken.
ACK	Signifies that the message is passed after checking for validity.
BYE	Signifies the intent to terminate the call. The database state is updated and temporary openings in the firewall are created for response to the BYE message.
CANCEL	Signifies abnormal data termination. The signaling sessions, media sessions, pregenerated temporary openings in the firewall, and the call database entry for the call are removed.

Table 32 Treatment of SIP Methods by the Cisco IOS Firewall (continued)

SIP Message	Purpose
INVITE	Occurs typically at the start of the call. The firewall will create a database entry upon receipt of this method and fill the database with relevant information extracted from this message. Temporary openings in the firewall will allow for a series of responses to the INVITE request. The temporary openings will be call-flow sensitive and will allow for responses for a fixed amount of time (t = 30 secs).
NO MATCH	Signifies a signaling message that is not present in the database.
Other Methods	Signifies that the message is passed if the call ID is present in the call database.
REGISTER	Results in the creation of an entry in the call database. Time-based, flow-control ACL firewall openings will allow for the response to the REGISTER and subsequent INVITE messages.
SESSION PROGRESS	Contains a response to the INVITE message, and it is a packet during the call creation phase. The packet is checked against the call database for validity of call ID and the media ports; the server proxy information is gathered from the packet. Media channels should be created in this phase.

Call Database

A call database, which contains the details of a call leg, is maintained for all call flows. A call database is created and maintained because there can be numerous signaling sessions for each call. Table 33 identifies the information available in the call database.

Table 33 Call Database Information

Type	Purpose
call_int_over	Checks to see whether or not call initialization is over, and if so, checks to see of the call is in the teardown phase
C con ip & C con port	Signifies the IP address and port in the contact field of the initiator; for example, “Contact:<sip:1111@172.16.0.3:5060;user=phone>”
C media ip & C media port	Signifies the IP address in the media field of the initiator; for example, “c=IN IP4 172.16.0.3”
C media port	Signifies the port in the media field of the initiator; for example, “m=audio 20758 RTP/AVP 0”
C src ip & C src port	Signifies the actual IP address and port of the initiator
C via ip & C via port	Signifies the IP address and port in the via field of the initiator (the first via line); for example, “Via: SIP/2.0/UDP 172.16.0.3:5060”
current sip state	Is the current state of the call (which helps to avoid retransmission)
from/to/callid	Is extracted from the “INVITE” SIP request message to identify the call
media header	Keeps the list of media sessions for the call

Table 33 Call Database Information (continued)

Type	Purpose
media opened	Signifies multiple messages that may have media information, so you need to check to see whether or not the media has been opened for the call
prev sip state	Signifies the previous state of the call (which helps to avoid retransmission)
S con ip & S con port	Signifies the IP address and port in the contact field for the responder
S media ip	Signifies the IP address in the media field for the responder
S media port	Signifies the port in the media field for the responder
S src ip & S src port	Signifies the actual IP address and port of the responder
S via ip & S via port	Signifies the IP address and port in the via field for the responder
signal header	Keeps the list of signaling sessions for the call
sip_proxy_traversed	Makes the firewall topologically aware of whether the call has traversed through proxies

How to Configure Your Firewall for SIP

To configure a Cisco IOS Firewall for SIP support, perform the following tasks:

- Configuring Firewall for SIP Support, page 594 (required)
- Verifying Firewall for SIP Support, page 595 (optional)
- Monitoring Firewall for SIP Support, page 596 (optional)

Configuring Firewall for SIP Support

To enable a firewall to support SIP, use the following commands.

Prerequisite

Before you configure Cisco IOS firewall support for SIP on your router, you first need to configure access lists, whose purpose normally is to block SIP traffic from unprotected networks for which the firewall will create temporary openings for specific traffic. For information about configuring access lists and the **access-list** command, see the chapter "Configuring IPSec Network Security" in the *Cisco IOS Security Configuration Guide*, Release 12.2, and the *Cisco IOS Command Reference*, Release 12.2 T, respectively.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip inspect name** *inspection-name* **sip** [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout** *seconds*]
4. **interface** *type number*
5. **ip inspect** *inspection-name* {**in** | **out**}

6. Repeat Steps 3 through 5 (Optional)

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip inspect name** *inspection-name* **sip** [**alert** {**on** | **off**}] [**audit-trail** {**on** |**off**}] [**timeout** *seconds*] **Example:** `Router(config)# ip inspect name voip sip`	Turns on inspection for SIP. • **alert**—Alert messages are generated. This function is **on** by default. • **audit-trail**—Audit trail messages are generated. This function is **off** by default. • **timeout**—Overrides the global channel inactivity timeout value.
Step 4	**interface** *type number* **Example:** `Router(config)# interface FastEthernet 0/0`	Configures an interface type and enters interface configuration mode.
Step 5	**ip inspect** *inspection-name* {**in** | **out**} **Example:** `Router(config-if)# ip inspect voip in`	Applies inspection configurations to an interface and for a particular traffic direction.
Step 6	If SIP calls are coming from other interfaces, repeat Steps 3 through 5 and apply SIP inspections for the calls that are coming from those interfaces.	**Note** The inspection of protocols other than SIP may not be desirable for traffic that comes from external networks, so it may be necessary to configure an additional inspection rule specifying only SIP.

Verifying Firewall for SIP Support

To verify Cisco IOS firewall session information, perform the following optional steps:

SUMMARY STEPS

1. **enable**
2. **show ip inspect name** *inspection-name*
3. **show ip inspect session** [**detail**]
4. **show ip access-list**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip inspect name** *inspection-name* **Example:** `Router# show ip inspect name voip`	(Optional) Displays the configured inspection rule.
Step 3	**show ip inspect session** [**detail**] **Example:** `Router# show ip inspect session`	(Optional) Displays existing sessions that are currently being tracked and inspected by the Cisco IOS firewall. • The optional **detail** keyword causes additional details about these sessions to be shown.
Step 4	**show ip access-list** **Example:** `Router# show ip access-list`	(Optional) Displays the contents of all current IP access lists.

Monitoring Firewall for SIP Support

To monitor firewall events, perform the following optional steps:

SUMMARY STEPS

1. **enable**
2. **debug ip inspect sip**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug ip inspect sip** **Example:** `Router# debug ip inspect sip`	(Optional) Displays the operations of the SIP inspection engine for debugging purposes.

Configuration Examples for Firewall SIP Support

This section provides the following configuration example:

- Firewall and SIP Configuration Example, page 597

Firewall and SIP Configuration Example

The following example shows how to allow outside initiated calls and internal calls. For outside initiated calls, an ACL needs to be punched to allow for the traffic from the initial signaling packet from outside. Subsequent signaling and media channels will be allowed by the inspection module.

```
ip inspect name voip sip
interface FastEthernet0/0
 ip inspect voip in
!
!
interface FastEthernet0/1
 ip inspect voip in
 ip access-group 100 in
!
!
access-list 100 permit udp host <gw ip> any eq 5060
access-list 100 permit udp host <proxy ip> any eq 5060
access-list deny ip any any
```

Additional References

For additional information related to Firewall Support for SIP, refer to the following references:

- Related Documents, page 597
- Standards, page 598
- MIBs, page 598
- RFCs, page 598
- Technical Assistance, page 599

Related Documents

Related Topic	Document Title
Cisco IOS firewall information and configuration tasks	The chapter "Configuring Context-Based Access Control" in the *Cisco IOS Security Configuration Guide*, Release 12.2
Cisco IOS firewall commands	The chapter "Context-Based Access Control Commands" in the *Cisco IOS Security Command Reference*, Release 12.2
SIP information and configuration tasks	The chapter "Configuring Session Initiation Protocol for Voice over IP" in the *Cisco IOS Voice, Video, and Fax Configuration Guide*, Release 12.2 and

Related Topic	Document Title
Additional SIP Information	*Guide to Cisco Systems' VoIP Infrastructure Solution for SIP*
Access lists and the **access-list** command	The chapter "Configuring IPSec Network Security" in the *Cisco IOS Security Configuration Guide*, Release 12.2, and the *Cisco IOS Command Reference*, Release 12.2, respectively.

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

RFCs[1]	Title
RFC 2543	*SIP: Session Initiation Protocol*

1. Not all supported RFCs are listed.

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **debug ip inspect**
- **ip inspect name**

Firewall Websense URL Filtering

The Firewall Websense Url Filtering feature enables your Cisco IOS firewall (also known as Cisco Secure Integrated Software [CSIS]) to interact with the Websense URL filtering software, thereby allowing you to prevent users from accessing specified websites on the basis of some policy. The Cisco IOS firewall works with the Websense server to know whether a particular URL should be allowed or denied (blocked).

Feature Specifications for the Firewall Websense URL Filtering feature

Feature History	
Release	**Modification**
12.2(11)YU	This feature was introduced.
12.2(15)T	This feature was integrated into Cisco IOS Release 12.2(15)T.
Supported Platforms	
For platforms supported in Cisco IOS Release 12.2(11)YU and 12.2(15)T, consult Cisco Feature Navigator.	

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Restrictions for Firewall Websense URL Filtering, page 602
- Information About Firewall Websense URL Filtering, page 602
- How to Configure Websense URL Filtering, page 605
- Configuration Examples for the Firewall and Webserver, page 612
- Additional References, page 614
- Command Reference, page 615
- Glossary, page 617

Restrictions for Firewall Websense URL Filtering

Websense Server Requirement

To enable this feature, you must have *at least* one Websense server; however, two or more Websense servers are preferred. Although there is no limit to the number of Websense servers you may have, and you can configure as many servers as you wish, only one server will be active at any given time—the primary server. URL look-up requests will be sent only to the primary server.

URL Filtering Support Restriction

This feature supports only one active URL filtering scheme at a time. (Before enabling Websense URL filtering, you should always ensure that there is not another URL filtering scheme configured, such as N2H2.)

Username Restriction

This feature does not pass the username and group information to the Websense server. However, the Websense server can work for user-based policies because it has another mechanism for getting the username to correspond to an IP address.

Information About Firewall Websense URL Filtering

To configure the Firewall Websense URL Filtering feature, you should understand the following concepts:

- Benefits of Firewall Websense URL Filtering, page 602
- Feature Design of Firewall WEBSENSE Url Filtering, page 604
- Supported Websense Server Features on a Cisco IOS Firewall, page 605

Benefits of Firewall Websense URL Filtering

The Cisco IOS Firewall Websense URL Filtering feature provides an Internet management application that allows you to control web traffic for a given host or user on the basis of a specified security policy. In addition, the following functions are available in this feature:

Primary and Secondary Servers

When users configure multiple Websense servers, the firewall will use only one server at a time—the primary server; all other servers are called secondary servers. When the primary server becomes unavailable for any reason, it becomes a secondary server and one of the secondary servers becomes the primary server.

A firewall marks a primary server as down when sending a request to or receiving a response from the server fails. When a primary server goes down, the system will go to the beginning of the configured servers list and try to activate the first server on the list. If the first server on the list is unavailable, it will try the second server on the list; the system will keep trying to activate a server until it is successful or until it reaches the end of the server list. If the system reaches the end of the server list, it will set a flag indicating that all of the servers are down, and it will enter allow mode.

When all the servers are down and the system is in allow mode, a periodic event that occurs for each minute will trace through the server list, trying to bring up a server by opening a TCP connection. If the TCP connection is successfully opened, the server is considered to be up, and the system will return to operational mode.

IP Cache Table

This function provides an IP cache table that contains the IP addresses of web servers whose underlying URLs can be accessed by all users and hosts.

The caching algorithm involves three parameters—the maximum number of IP addresses that can be cached, an idle time, and an absolute time. The algorithm also involves two timers—idle timer and absolute timer. The idle timer is a small periodic timer (1 minute) that checks to see whether the number of cached IP addresses in the cache table exceeds 80 percent of the maximum limit. If the cached IP addresses have exceeded 80 percent, it will start removing idle entries; if it has not exceeded 80 percent, it will quit and wait for the next cycle. The absolute timer is a large periodic timer (1 hour) that is used to remove all of the elapsed entries. (The age of an elapsed entry is greater than the absolute time.) An elapsed entry will also be removed during cache lookup.

The idle time value is fixed at 10 minutes. The absolute time value is taken from the Websense look-up response, which is often greater than 15 hours. The absolute value for cache entry made out of exclusive-domains is 12 hours. The maximum number of cache entries is configurable.

To configure cache table parameters, use the **ip urlfilter cache** command.

Packet Buffering

This function allows you to increase the maximum number of HTTP responses that a Cisco IOS firewall can hold. If the HTTP responses arrive prior to a Websense server reply, this buffering scheme allows your firewall to store a maximum of 200 HTTP responses. (After 200 responses have been reached, the firewall will drop further responses.) The responses will remain in the buffer until an allow or deny message is received from Websense: if the status indicates that the URL is allowed, the firewall will release the HTTP responses in the buffer to the browser of the end user; if the status indicates that the URL is blocked, the firewall will discard the HTTP responses in the buffer and close the connection to both ends. This function prevents numerous HTTP responses from overwhelming your system.

To configure the maximum number of HTTP responses for your firewall, use the **ip urlfilter max-resp-pak** command.

Exclusive Domains

This function provides a configurable list of domain names so that the Cisco IOS firewall does not have to send a lookup request to the Websense server for the HTTP traffic that is destined for one of the domains in the exclusive list. Thus, the Websense server does not have to deal with look-up requests for HTTP traffic that is destined for a host that has already been marked as "allowed."

Flexibility when entering domain names is also provided; that is, the user can enter the complete domain name or a partial domain name. If the user adds a complete domain name, such as "www.cisco.com," to the exclusive domain list, all HTTP traffic whose URLs are destined for this domain (such as www.cisco.com/news and www.cisco.com/index) will be excluded from the Websense URL filtering policies, and based on the configuration, the URLs will be permitted or blocked (denied).

If the user adds only a partial domain name to the exclusive domain list, such as ".cisco.com," all URLs whose domain names end with this partial domain name (such as www.cisco.com/products and www.cisco.com/eng) will be excluded from the Websense URL filtering policies, and based on the configuration, the URLs will be permitted or blocked (denied).

To configure an exclusive domain list, use the **ip urlfilter exclusive-domain** command.

Allow Mode

The system will go into allow mode when connections to all the Websense servers are down. The system will return to normal mode when a connection to at least one web Websense server is up. Allow mode directs your system to forward or drop all packets on the basis of the configurable allow mode setting. By default, allow mode is off, so all HTTP requests are forbidden if all Websense servers are down.

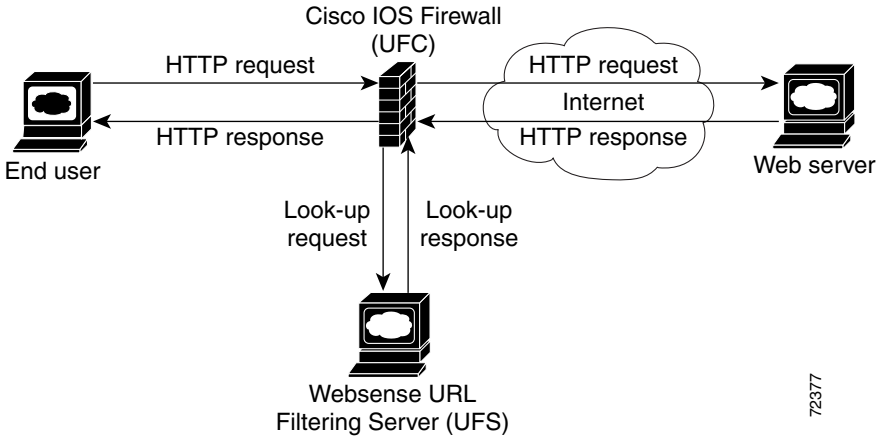
To configure allow mode for your system, use the **ip urlfilter allowmode** command.

Feature Design of Firewall WEBSENSE Url Filtering

Note This feature assumes that the Websense server will be part of a protected network and that requests from the Cisco IOS firewall will not travel over any unprotected network to reach the Websense server.

Figure 29 and the corresponding steps explain a sample URL filtering network topology.

Figure 29 Firewall WEBSENSE URL Filtering Sample Topology



1. The end user browses a page on the web server, and the browser sends an HTTP request.
2. After the Cisco IOS firewall receives this request, it forwards the request to the web server while simultaneously extracting the URL and sending a look-up request to the Websense server.
3. After the Websense server receives the look-up request, it checks its database to see whether it should permit or deny the URL; it returns a permit or deny status via a look-up response to the Cisco IOS firewall.
4. After the Cisco IOS firewall receives this look-up response, it performs one of the following functions:

- If the look-up response permits the URL, it sends the HTTP response to the end user.
- If the look-up response denies the URL, the Websense server redirects the user to its own internal web server, which displays a message that describes the category under which the URL is blocked; thereafter, the connection is reset to both ends.

Supported Websense Server Features on a Cisco IOS Firewall

The Cisco IOS firewall supports all of the filtering and user authentication methods that are supported by the Websense server.

The following filtering methods are supported:

- Global filtering, which is applied to all users, groups, and IP addresses
- User- or group-based filtering, which is applied to a specific user or group
- Keyword-based filtering, which is applied on the basis of specific keywords (for example, a user can configure a policy for which all URLs with the keyword “dog” will be denied)
- Category-based filtering, which is applied on the basis of specific categories
- Customized filtering, which allows the user to apply a policy for customized URLs

The NT LAN Manager (NTLM) and Lightweight Directory Access Protocol (LDAP) user authentication methods are supported in this feature. Websense uses these methods to authenticate the user when the firewall does not pass the authenticated username along with the look-up request.

When the username is not passed along with the look-up request, the Websense server retrieves the username through one of the following methods:

- Manual authentication—Websense redirects the user to its own internal web server, which displays a challenge or response for the username and password. (This process is similar to when a user is blocked, but in this process, an authentication message is displayed instead of a blocked message.) Thereafter, Websense checks the NTLM or LDAP directory service to see if the username and password are a match. If there is a match, Websense associates the username with the source IP address and policies can be created for that username.
- Transparent ID (XID)—Websense has an agent that automatically associates users, when they log onto a Windows network, to their IP addresses. Unlike manual authentication, this method does not require an additional logon by the user. However, this method can be used only for Windows.

Note Although Websense also supports user authentication via TACACS or RADIUS, this feature currently does not support these protocols for user authentication.

How to Configure Websense URL Filtering

To configure your Cisco IOS firewall to interact with at least one Websense server to provide URL filtering, configure the following procedures:

- Configuring Firewall WEBSENSE URL Filtering, page 605 (required)
- Verifying Cisco IOS Firewall and Websense URL Filtering, page 610 (optional)
- Maintaining the Cache Table, page 610 (optional)
- Monitoring the URL Filter Subsytems, page 611 (optional)

Configuring Firewall WEBSENSE URL Filtering

Websense is a third-party filtering software that can filter HTTP requests on the basis of the following policies: destination hostname, destination IP address, keywords, and username. The software maintains a URL database of more than 20 million sites organized into more than 60 categories and subcategories.

Prerequisites

Before enabling Websense URL filtering, you should always ensure that there is not another URL filtering scheme configured, such as N2H2. If you try to enter a new filtering scheme when one already exists, the new scheme will be ignored, and the system will display an error message that says, "different URL filtering scheme cannot co-exist."

Restrictions

Enabling HTTP inspection (via the **ip inspect name** command) triggers the Java applet scanner, which is CPU intensive. The only way to stop the Java applet scanner is to specify the **java-list** *access-list* keyword and argument and configure a standard access list to allow any traffic. Configuring URL filtering without enabling the **java-list** *access-list* keyword and argument will severely impact performance.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip inspect name** *inspection-name* **http** [**urlfilter**] [**java-list** *access-list*] [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout** *seconds*]
4. **ip inspect** *inspection-name* {**in** | **out**}
5. **interface** *type slot/port*
6. **ip urlfilter server vendor** {**websense** | **n2h2**} *ip-address* [**port** *port-number*] [**timeout** *seconds*] [**retransmit** *number*]
7. **ip urlfilter alert**
8. **ip urlfilter audit-trail**
9. **ip urlfilter urlf-server-log**
10. **ip urlfilter exclusive-domain** {**permit** | **deny**} *domain-name*
11. **ip urlfilter cache** *number*
12. **ip urlfilter allowmode** [**on** | **off**]
13. **ip urlfilter max-resp-pak** *number*
14. **ip urlfilter max-request** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip inspect name** *inspection-name* **http** [**urlfilter**] [**java-list** *access-list*] [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout** *seconds*] **Example:** `Router(config)# ip inspect name fw_urlf http urlfilter java-list 51 timeout 30`	Turns on HTTP inspection. The **urlfilter** keyword associates URL filtering with HTTP inspection. **Note** You may configure two or more inspections in a router, but URL filtering will work only with the inspections in which the **urlfilter** keyword is enabled. **Note** Enabling HTTP inspection with or without any options triggers the Java applet scanner, which is CPU intensive. The only way to stop the Java applet scanner is to specify the **java-list** *access-list* keyword and argument. Configuring URL filtering without enabling the **java-list** *access-list* keyword and argument will severely impact performance.
Step 4	**ip inspect** *inspection-name* {**in** | **out**} **Example:** `Router(config)# ip inspect fw_urlf in`	Applies a set of inspection rules to an interface. • The **in** keyword applies the inspection rules to inbound traffic.
Step 5	**interface** *type slot*/*port* **Example:** `Router(config)# interface ethernet 1/0`	Configures an interface type and enters interface configuration mode.
Step 6	**ip urlfilter server vendor** {**websense** | **n2h2**} *ip-address* [**port** *port-number*] [**timeout** *seconds*] [**retransmit** *number*] **Example:** `Router(config)# ip urlfilter server vendor websense 10.201.6.202`	Configures a Websense server to interact with the firewall to filter HTTP requests on the basis of a specified policy. • *ip-address*—IP address of the vendor server. • **port** *port-number*—Port number that the vendor server listens on. The default port number is 15868. • **timeout** *seconds*—Length of time that the firewall will wait for a response from the vendor server. The default timeout is 5 minutes. • **retransmit** *number*—Number of times the firewall will retransmit the request when a response does not arrive. The default value is 2 times.

	Command or Action	Purpose
Step 7	**ip urlfilter alert** **Example:** `Router(config)# ip urlfilter alert`	(Optional) Enables the system alert, which displays system messages such as a server entering allow mode or going down. • The system alert is enabled by default.
Step 8	**ip urlfilter audit-trail** **Example:** `Router(config)# ip urlfilter audit-trail`	(Optional) Enables the logging of messages into the syslog server of router. This function is disabled by default.
Step 9	**ip urlfilter urlf-server-log** **Example:** `Router(config)# ip urlfilter urlf-server-log`	(Optional) Enables the logging of system messages on the URL filtering server (the Websense server). • This function is disabled by default.
Step 10	**ip urlfilter exclusive-domain** {**permit** | **deny**} *domain-name* **Example:** `Router(config)# ip urlfilter exclusive-domain permit www.cisco.com`	(Optional) Adds a domain name to or from the exclusive domain list so that the firewall does not have to send look-up requests to the Websense server. • **permit**—Permits all traffic destined for the specified domain name. • **deny**—Denies all traffic destined for the specified domain name. • *domain-name*—Domain name that is added or removed from the exclusive domain list.
Step 11	**ip urlfilter cache** *number* **Example:** `Router(config)# ip urlfilter cache 4500`	(Optional) Configures cache table parameters. • *number*—Maximum number of destination IP addresses that can be cached into the cache table; the default is 5000.
Step 12	**ip urlfilter allowmode** [**on** | **off**] **Example:** `Router(config)# ip urlfilter allowmode on`	(Optional) Turns on the default mode of the filtering systems. • **on**—Allows HTTP requests to pass to the end user if all Websense servers are down. • **off**—Blocks all HTTP requests if all Websense servers are down; **off** is the default setting.
Step 13	**ip urlfilter max-resp-pak** *number* **Example:** `Router(config)# ip urlfilter max-resp-pak 150`	(Optional) Configures the maximum number of HTTP responses that the firewall can keep in its packet buffer. • The default and absolute maximum value is 200.
Step 14	**ip urlfilter max-request** *number* **Example:** `Router(config)# ip urlfilter maxrequest 500`	(Optional) Sets the maximum number of outstanding requests that can exist at any given time. • The default value is 1000.

Troubleshooting Tips

This feature introduces the following alert messages:

- "%URLF-3-SERVER_DOWN: Connection to the URL filter server 10.92.0.9 is down"

  This level three LOG_ERR-type message is displayed when a configured UFS goes down. When this happens, the firewall will mark the configured server as secondary and try to bring up one of the other secondary servers and mark that server as the primary server. If there is no other server configured, the firewall will enter allow mode and display the "URLF-3-ALLOW_MODE" message.

- %URLF-3-ALLOW_MODE: Connection to all URL filter servers are down and ALLOW MODE is OFF

  This LOG_ERR type message is displayed when all UFSs are down and the system enters allow mode.

  Note Whenever the system goes into allow mode (all filter servers are down), a periodic keepalive timer will be triggered that will try to bring up a server by opening a TCP connection.

- "%URLF-5-SERVER_UP: Connection to an URL filter server 10.92.0.9 is made; the system is returning from ALLOW MODE"

  This LOG_NOTICE-type message is displayed when the UFSs are detected as being up and the system is returning from allow mode.

- "%URLF-4-URL_TOO_LONG:URL too long (more than 3072 bytes), possibly a fake packet?"

  This LOG_WARNING-type message is displayed when the URL in a look-up request is too long; any URL longer than 3K will be dropped.

- "%URLF-4-MAX_REQ: The number of pending request exceeds the maximum limit <1000>"

  This LOG_WARNING-type message is displayed when the number of pending requests in the system exceeds the maximum limit and all further requests are dropped.

To display these alert messages, use the **ip urlfilter alert** command.

This feature introduces the following syslog messages:

- "%URLF-6-SITE_ALLOWED: Client 10.0.0.2:12543 accessed server 10.76.82.21:8080"

  This message is logged for each request whose destination IP address is found in the cache. It includes the source IP address, source port number, destination IP address, and destination port number. The URL is not logged because the IP address of the request is found in the cache, so parsing the request and extracting the URL is a waste of time.

- "%URLF-4-SITE-BLOCKED: Access denied for the site 'www.sports.com'; client 10.54.192.6:34557 server 172.24.50.12:80"

  This message is logged when a request finds a match against one of the blocked domains in the exclusive-domain list or the corresponding entry in the IP cache.

- "%URLF-6-URL_ALLOWED: Access allowed for URL http://www.websense.com/; client 10.54.192.6:54123 server 192.168.0.1:80"

  This message is logged for each URL request that is allowed by a UFS. It includes the allowed URL, source IP address, source port number, destination IP address, and destination port number. Longer URLs will be truncated to 300 bytes and then logged.

- "%URLF-6-URL_BLOCKED: Access denied URL http://www.google.com; client 12.54.192.6:54678 server 64.192.14.2:80"

  This message is logged for each URL request that is blocked by a UFS. It includes the blocked URL, source IP address, source port number, destination IP address, and destination port number. Longer URLs will be truncated to 300 bytes and then logged.

To display these syslog messages, use the **ip urlfilter audit-trail** command.

Verifying Cisco IOS Firewall and Websense URL Filtering

To verify that the Firewall WEBSENSE URL Filtering feature is working, perform any of the following optional steps:

SUMMARY STEPS

1. **enable**
2. **show ip urlfilter cache**
3. **show ip urlfilter config**
4. **show ip urlfilter statistics**

DETAILED STEPS

Command or Action	Purpose
enable **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
show ip urlfilter cache **Example:** `Router# show ip urlfilter cache`	Displays the destination IP addresses that are cached into the cache table.
show ip urlfilter config **Example:** `Router# show ip urlfilter config`	Displays the size of the cache, the maximum number of outstanding requests, the allow mode state, and the list of configured Websense servers.
show ip urlfilter statistics **Example:** `Router# show ip urlfilter statistics`	Displays information such as the number of requests that are sent to the Websense server, the number of responses received from the Websense server, the number of pending requests in the system, the number of failed requests, the number of blocked URLs.

Maintaining the Cache Table

To clear the cache table of a specified address or of all IP addresses, perform the following optional steps:

SUMMARY STEPS

1. **enable**
2. **clear ip urlfilter cache**

DETAILED STEPS

Command or Action	Purpose
enable **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
clear ip urlfilter cache {*ip-address* | **all**} **Example:** `Router# clear ip urlfilter cache all`	Clears the cache table.

Monitoring the URL Filter Subsytems

To monitor the URL filter subsystems, perform the following optional steps:

SUMMARY STEPS

1. **enable**
2. **debug ip urlfilter** {**func-trace** | **detailed** | **events**}

DETAILED STEPS

Command or Action	Purpose
enable **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
debug ip urlfilter {**func-trace** | **detailed** | **events**} **Example:** `Router# debug ip urlfilter detailed`	Enables debugging information of the URL filter subsystems. • **func-trace**—Prints a sequence of important functions that are called when configuring URL filtering. • **detailed**—Prints detailed information about various activities that occur during URL filtering. • **events**—Prints various events, such as queue event, timer event, and socket event.

Configuration Examples for the Firewall and Webserver

This section provides the following comprehensive configuration example:

- URL Filter Client (Firewall) Configuration Example, page 612

URL Filter Client (Firewall) Configuration Example

The following example shows how to configure the Cisco IOS firewall (also known as the URL filter client [UFC]) for Websense URL filtering:

```
hostname fw9-7200b
!
logging buffered 64000 debugging
enable secret 5 $1$qMOf$umPb75mb3sV27JpNbW//7.
!
clock timezone PST -8
clock summer-time PDT recurring
ip subnet-zero
ip cef
no ip domain lookup
!
ip inspect name test http urlfilter
ip urlfilter cache 5
ip urlfilter exclusive-domain permit .weapons.com
ip urlfilter exclusive-domain deny .nbc.com
ip urlfilter exclusive-domain permit www.cisco.com
ip urlfilter audit-trail
ip urlfilter alert
ip urlfilter server vendor websense 192.168.3.1
ip audit notify log
ip audit po max-events 100
ip port-map http port 8080
!
no voice hpi capture buffer
no voice hpi capture destination
!
mta receive maximum-recipients 0
!
interface FastEthernet0/0
 ip address 192.168.3.254 255.255.255.0
 ip access-group 101 out
 ip nat inside
 ip inspect test in
 no ip route-cache
 no ip mroute-cache
!
interface Ethernet1/0
 ip address 10.6.9.7 255.255.0.0
 ip nat outside
 no ip route-cache
 no ip mroute-cache
 duplex half
!
interface Ethernet1/1
 no ip address
 no ip mroute-cache
 shutdown
 duplex half
!
interface Ethernet1/2
```

```
 no ip address
 no ip mroute-cache
 shutdown
 duplex half
!
interface Ethernet1/3
 no ip address
 no ip mroute-cache
 shutdown
 duplex half
!
interface Serial2/0
 no ip address
 no ip mroute-cache
 shutdown
 dsu bandwidth 44210
 framing c-bit
 cablelength 10
 serial restart_delay 0
 fair-queue
!
ip nat pool devtest 10.6.243.21 10.6.243.220 netmask 255.255.0.0
ip nat inside source list 1 pool devtest
ip nat inside source static 192.168.3.1 10.6.243.1
ip nat inside source static 192.168.3.2 10.6.243.2
ip nat inside source static 192.168.3.3 10.6.243.3
ip classless
ip route 192.168.0.30 255.255.255.255 10.6.0.1
no ip http server
no ip http secure-server
!
ip pim bidir-enable
!
!
access-list 101 deny   tcp any any
access-list 101 deny   udp any any
access-list 101 permit ip any any
access-list 102 deny   tcp any any
access-list 102 deny   udp any any
access-list 102 permit ip any any
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
!
call rsvp-sync
!
!
mgcp profile default
!
dial-peer cor custom
!!
!
gatekeeper
 shutdown
!
!
line con 0
 exec-timeout 0 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 password letmein
 login
```

```
!
exception core-file sisu-devtest/coredump/fw9-7200b.core
exception dump 192.168.0.1
no scheduler max-task-time
!
end
```

Additional References

For additional information related to the Firewall Websense URL Filtering feature, refer to the following references:

- Related Documents, page 614
- Standards, page 614
- MIBs, page 614
- RFCs, page 615
- Technical Assistance, page 615

Related Documents

Related Topic	Document Title
N2H2 URL filtering	*Firewall N2H2 Support*, Cisco IOS Release 12.2(15)T feature module
Additional CBAC configuration tasks and information	The part "Traffic Filtering and Firewalls" in the *Cisco IOS Security Configuration Guide*, Release 12.2
Additional CBAC commands	The part "Traffic Filtering and Firewalls" in the *Cisco IOS Security Command Reference*, Release 12.2 T

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

RFCs[1]	Title
RFC 1945	*Hypertext Transfer Protocol — HTTP/1.0*
RFC 2616	*Hypertext Transfer Protocol — HTTP/1.1*

1. Not all supported RFCs are listed.

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

New Commands

- **clear ip urlfilter cache**
- **debug ip urlfilter**
- **ip urlfilter alert**
- **ip urlfilter allowmode**
- **ip urlfilter audit-trail**
- **ip urlfilter cache**

- **ip urlfilter exclusive-domain**
- **ip urlfilter max-request**
- **ip urlfilter max-resp-pak**
- **ip urlfilter server vendor**
- **ip urlfilter urlf-server-log**
- **show ip urlfilter cache**
- **show ip urlfilter config**
- **show ip urlfilter statistics**

Modified Commands

- **ip inspect name**

Glossary

CSIS—Cisco Secure Integrated Software. CSIS is a content-based firewall that currently inspects application data, checks for protocol conformance, extracts the relevant port information to create dynamic access list entries that successfully allow return traffic, and closes the ports at the end of the session.

UFC—URL filter client. UFC is a separate process that accepts URLs from CSIS, forwards the URL to the Websense server, and processes the replies from the vendor server (Websense or N2H2).

UFS—URL filter server. UFS is a generic name given to the vendor server (Websense or N2H2), which processes URLs and decides whether to allow or deny web traffic on the basis of a given policy.

Note Refer to the *Internetworking Terms and Acronyms* for terms not included in this glossary.

Firewall Support of Skinny Client Control Protocol (SCCP)

The Firewall Support of Skinny Client Control Protocol (SCCP) feature enables Context-Based Access Control (CBAC) inspection to support the Voice over IP (VoIP) protocol, Skinny Client Control Protocol (SCCP). That is, CBAC inspects Skinny control packets that are exchanged between a Skinny client and the Call Manager (CM); CBAC then configures the router (also known as the Cisco IOS Firewall) to enable the Skinny data channels to traverse through the router.

Feature Specifications for the Firewall Support of Skinny Client Control Protocol (SCCP) Feature

Feature History	
Release	**Modification**
12.3(1)	This feature was introduced.
Supported Platforms	
For platforms supported in Cisco IOS Release 12.3(1), consult Cisco Feature Navigator.	

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for Firewall Support of Skinny Client Control Protocol (SCCP), page 620
- Restrictions for Firewall Support of Skinny Client Control Protocol (SCCP), page 620
- Information About Firewall Support of Skinny Client Control Protocol (SCCP), page 620
- How to Configure Your Firewall for Skinny Support, page 622
- Configuration Examples for Firewall Skinny Support, page 626
- Additional References, page 628
- Command Reference, page 629

Prerequisites for Firewall Support of Skinny Client Control Protocol (SCCP)

The Skinny inspection module is part of the inspection subsystem; thus, your router must be running an image that has firewall support.

Restrictions for Firewall Support of Skinny Client Control Protocol (SCCP)

This feature has the following restrictions:

- Skinny inspection will inspect only the SCCP sessions that have been established after the firewall is configured with Skinny inspection. That is, any SCCP sessions that were established through the firewall before the Skinny inspection was configured will not be inspected.
- This feature does not support Music on Hold (MOH) when a device other than the CM is the music server. (This feature does support MOH when the CM is the music server.)
- This feature does not address either the multicast functionality of SCCP or the functionality of multiple active calls on a single Skinny client.

This feature does not support the following Skinny and firewall configurations:

- The firewall and CM cannot be in the same router. Skinny inspection does not support this configuration because the current firewall implementation does not inspect sessions that start or terminate at the router. Thus, Skinny inspection will work only with an external CM.
- The CM and the Skinny client cannot be on three different networks that are separated at the firewall. The current firewall implementation does not inspect sessions that have devices residing on more than two distinct networks that are segregated at the firewall. That is, if there are more than two interfaces at the firewall, session inspection is not supported.

Information About Firewall Support of Skinny Client Control Protocol (SCCP)

To configure the Firewall Support of SCCP feature, you must understand the following concepts:

- Context-Based Access Control Overview, page 620
- Skinny Overview, page 621
- CBAC and Skinny Functionality Overview, page 621

Context-Based Access Control Overview

CBAC extends the concept of static access control lists (ACLs) by introducing dynamic ACL entries that open the necessary application ports on the basis of a specific application and close these ports at the end of the application session. CBAC achieves this functionality by inspecting the application data,

checking for conformance of the application protocol, extracting the relevant port information to create the dynamic ACL entries, and closing these ports at the end of the session. CBAC is designed to easily allow a new application inspection whenever support is needed.

Skinny Overview

Skinny enables voice communication between two Skinny clients through the use of a CM. Typically, the CM provides service to the Skinny clients on TCP Port 2000. Initially, a Skinny client connects to the CM by establishing a TCP connection; the client will also establish a TCP connection with a secondary CM, if available. After the TCP connection is established, the client will register with the primary CM, which will be used as the controlling CM until it reboots or there is a keepalive failure. Thus, the Skinny TCP connection between the client and the CM exists forever and is used to establish calls coming to or from the client. If a TCP connection failure is detected, the secondary CM is used. All data channels established with the previous CM remain active and will be closed after the end parties hang up the call.

Table 34 lists the set of messages that are necessary for the data sessions to open and close. Skinny inspection will examine the data sessions that are deemed for opening and closing the access list pin holes.

Table 34 Skinny Data Session Messages

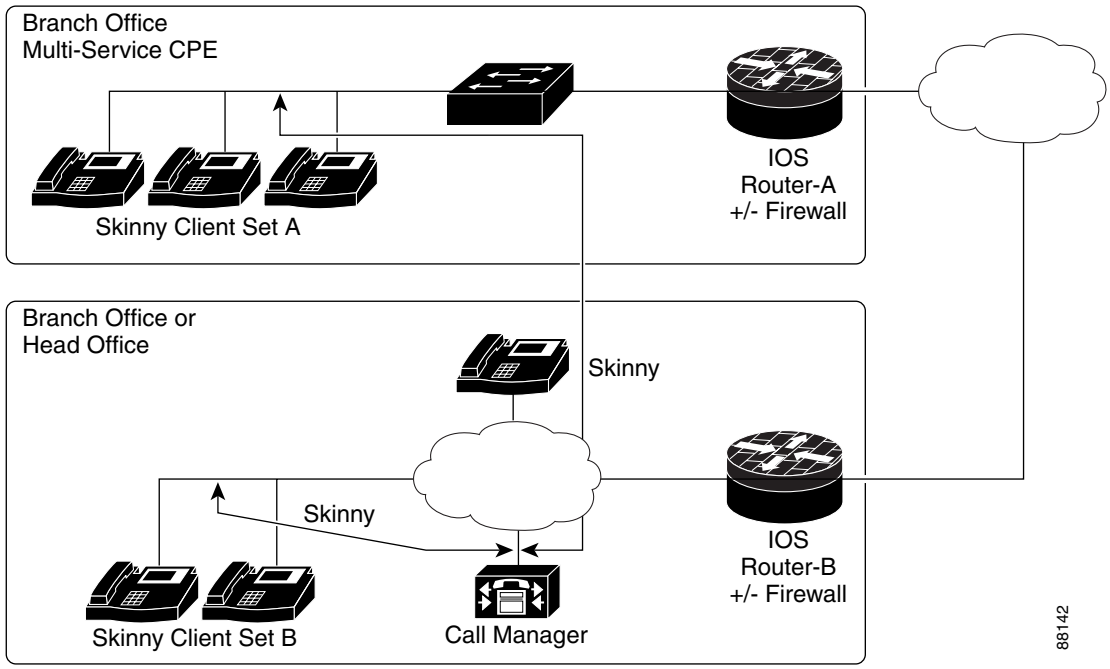
Skinny Inspection Message	Description
StationOpenReceiveChannelAckMessage	Contains the IP address and port information of the Skinny client sending this message. This message also contains the status of whether or not the client is willing to receive the voice traffic.
StationStartMediaTransmissionMessage	Contains the IP address and port information of the remote Skinny client.
StationCloseReceiveChannelMessage	CM instructs the Skinny client (on the basis of the information in this message) to close the receiving channel.
StationStopMediaTransmissionMessage	CM instructs the Skinny client (on the basis of the information in this message) to stop transmitting voice traffic.
StationStopSessionTransmissionMessage	CM instructs the Skinny client (on the basis of the information in this message) to end an indicated session.

CBAC and Skinny Functionality Overview

Figure 30 depicts typical deployment solutions that are supported by CBAC inspection for Skinny. According to Figure 1, a firewall with Skinny inspection can be configured on Cisco IOS Router A, Cisco IOS Router B, or both routers, thereby addressing the following three scenarios:

- A Cisco IOS router with a firewall on the customer premises equipment (CPE) side, supporting Skinny VoIP phone
- A Cisco IOS router with a firewall on the CM side
- A Cisco IOS router with a firewall at both ends of the connection

Figure 30 CBAC Inspection for Skinny Sample Topology



How to Configure Your Firewall for Skinny Support

To configure a Cisco IOS Firewall for SCCP support, perform the following tasks:

- Configuring Basic Skinny CBAC Inspection, page 622
- Setting Skinny CBAC Session Timeouts, page 623
- Configuring Port to Application Mapping, page 624
- Verifying Cisco IOS Firewall for Skinny Support, page 625
- Monitoring Cisco IOS Firewall for Skinny Support, page 626

Configuring Basic Skinny CBAC Inspection

Perform the following required steps to configure a basic Skinny CBAC configuration:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip inspect name** *inspection-name protocol* [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout** *seconds*]
4. **ip inspect name** *inspection-name protocol* [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout** *seconds*] (Optional. Required if the TFTP server is outside the firewall.)
5. **interface** *type number*

6. **ip access-group** {*access-list-number*} {**in** | **out**}
7. **ip inspect** *inspection-name* {**in** | **out**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip inspect name** *inspection-name protocol* [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout** *seconds*] **Example:** `Router(config)# ip inspect name firewall skinny`	Enables CBAC Skinny inspections.
Step 4	**ip inspect name** *inspection-name protocol* [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout** *seconds*] **Example:** `Router(config)# ip inspect name firewall tftp`	(Optional. Required if the TFTP server is outside the firewall.) Defines a set of inspection rules.
Step 5	**interface** *type number* **Example:** `Router(config)# interface FastEthernet 0/0`	Configures an interface type and enters interface configuration mode.
Step 6	**ip access-group** {*access-list-number*} {**in** | **out**} **Example:** `Router(config-if)# ip access-group 100 in`	Control access to an interface. Number of the access list that is blocking incoming traffic.
Step 7	**ip inspect** *inspection-name* {**in** | **out**} **Example:** `Router(config-if)# ip inspect firewall out`	Applies a set of inspection rules to an interface.

Setting Skinny CBAC Session Timeouts

Session timeouts are triggered when traffic is not seen on a particular session for a configured amount of time. (This value is configured via the **ip inspect name** command.) After the inactivity timeout is triggered, the firewall will clean up the session and deallocate all of the session data.

You must set the inactivity timeout value for Skinny to a greater value than the keepalive timeout value that is configured between the CM and Skinny clients. Otherwise, the Skinny connection may become inaccessible for inspection because the firewall might delete the session-related information due to inactivity.

After the inactivity timeout is triggered, the inspection module will send reset (RST packets) to both ends of the connection. Any data channels that are associated with the control channel will not be closed. After both end parties hang up, there will not be any traffic on the data channels and the connection will eventually timeout.

Note If the inactivity timeout of the control channel that is connected to the primary CM is less than the keepalive timeout that is sent by the CM to the Skinny client, the firewall will set the inactivity timeout to three times the keepalive timeout. If a timeout is not configured, the default value of 3600 seconds will be used.

Configuring Port to Application Mapping

By default, the Skinny inspection will inspect SCCP messages to or from the CM on TCP port 2000. If you prefer to configure the CM to use a different port, the port to application mapping (PAM) feature should be used to specify the desired port to the Cisco IOS firewall. Thus, the firewall will inspect the SCCP messages in the desired port and in port 2000. To configure the CM to use a different port via PAM, use the **ip port-map** command.

Prerequisites

Before you can configure PAM, you must first configure the steps in the section, "Configuring Basic Skinny CBAC Inspection."

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip port map** *appl_name* **port** *port_num* [**list** *acl_num*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip port map** *appl_name* **port** *port_num* [**list** *acl_num*] **Example:** `Router(config)# ip port map skinny port 2100`	(Optional) Creates a port to address mapping for SCCP. This command allows you to indicate additional ports that need to be monitored for SCCP.

Verifying Cisco IOS Firewall for Skinny Support

To display active Skinny session information, perform the following optional steps:

SUMMARY STEPS

1. **enable**
2. **show ip inspect** {**name** *inspection-name* | **config** | **interfaces** | **session** [**detail**] | **all**}
3. **show ip access-list**
4. **show ip port-map** [*appl_name* | **port** *port_num*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip inspect** {**name** *inspection-name*	**config**	**interfaces**	**session** [**detail**]	**all**} **Example:** `Router# show ip inspect session detail`	(Optional) Displays existing sessions that are currently being tracked and inspected by CBAC. The optional **detail** keyword causes additional details about these sessions to be shown.

	Command or Action	Purpose
Step 3	**show ip access-list** **Example:** `Router# show ip access-list`	(Optional) Displays the contents of all current IP access lists, which includes the dynamic access lists created by Skinny inspection.
Step 4	**show ip port-map** [*appl_name* | **port** *port_num*] **Example:** `Router# show ip port-map skinny`	(Optional) Displays information about the active port to application mappings on the router. Use this command to view Skinny port map information. • *appl_name*—Displays Skinny-specific PAM information. (You must specify the *skinny* argument.)

Monitoring Cisco IOS Firewall for Skinny Support

To monitor debugging messages related to Skinny inspection, perform the following optional steps:

SUMMARY STEPS

1. **enable**
2. **debug ip inspect** {*sccp* | **detailed**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug ip inspect** {*sccp* | **detailed**} **Example:** `Router# debug ip inspect sccp`	(Optional) Displays and logs the debugging messages related to SCCP inspection.

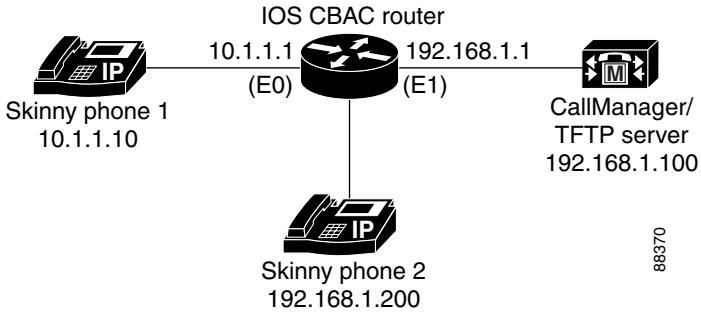
Configuration Examples for Firewall Skinny Support

This section provides the following configuration example:

- Firewall and Skinny Configuration Example, page 627

Firewall and Skinny Configuration Example

Figure 31 Skinny and CBAC Configuration



The following is an example of how to configure a Cisco IOS firewall for Skinny support and includes PAM (see Figure 31):

```
!  Define the name of the router as "CBAC-Firewall."
!
host CBAC-Firewall
!
!  Create a DHCP server process to offer out 10.1.1.x addresses on the
!  inside network. Option 150 is used by Cisco IP phones as where to
!  look for their configuration file. A default router is required so that all
!  the IP phones can talk to networks other than just to the local 10.1.1.x.
!
ip dhcp pool localnetwork
   network 10.1.1.0 255.255.255.0
   option 150 ip 192.168.1.100
   default-router 10.1.1.1
!
!  Prevent the DHCP server process from assigning 10.1.1.1 -.9 as an IP
!  address on the local network. This is done to hold the addresses .2 - .9 as static-
!  defined addresses.
!
ip dhcp excluded-address 10.1.1.1 10.1.1.9
!
!  Define firewall rules to all Skinny traffic in/out along with TFTP
!  services.
!
ip inspect name fwout tftp
ip inspect name fwout skinny
!
!  Prevent any traffic from coming in.
!
access-list 100 deny ip any any
!
interface ethernet 1
  ip access-group 100 in
  ip inspect firewall out


If the CallManager is requiring Skinny registration to happen on port tcp/2100, you will
still need the above configuration plus the following additional step.

ip port map skinny port 2100
```

Additional References

The following sections provide additional references related to the Firewall Support of Skinny Client Control Protocol (SCCP) feature:

- Related Documents, page 628
- Standards, page 628
- MIBs, page 628
- RFCs, page 629
- Technical Assistance, page 629

Related Documents

Related Topic	Document Title
Additional CBAC information and configuration tasks	The chapter "Configuring Context-based Access Control" in the *Cisco IOS Security Configuration Guide*, Release 12.3
CBAC commands	*Cisco IOS Security Command Reference*, Release 12.3
PAM information and configuration tasks	The chapter "Configuring Port to Application Mapping" in the *Cisco IOS Security Configuration Guide*, Release 12.3

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs[1]	Title
None	—

1. Not all supported RFCs are listed.

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **debug ip inspect**
- **ip inspect name**
- **ip port-map**

Granular Protocol Inspection

The Granular Protocol Inspection feature adds flexibility to the Cisco IOS Firewall by allowing it to perform a higher degree of inspection of TCP and User Data Protocol (UDP) traffic for most RFC 1700 application types.

Feature History for Granular Protocol Inspection

Release	Modification
12.3(14)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for Granular Inspection Protocol, page 631
- Restrictions for Granular Inspection Protocol, page 632
- Information About Granular Protocol Inspection, page 632
- How to Configure Granular Protocol Inspection, page 633
- Configuration Examples for Granular Protocol Inspection, page 636
- Additional References, page 638
- Command Reference, page 639
- Glossary, page 640

Prerequisites for Granular Inspection Protocol

- Cisco IOS Firewall software must be installed in your network.
- Access control lists (ACLs) must be applied to specified interfaces to enable the existing firewall software to function properly.

Restrictions for Granular Inspection Protocol

Port ranges cannot be specified directly in the **ip inspect name** command; use the port-to-application mapping (PAM) table.

Information About Granular Protocol Inspection

To use the Granular Protocol Inspection feature, you need to understand the following concepts:

- Cisco IOS Firewall, page 632
- Granular Protocol Inspection, page 632
- Benefits, page 633

Cisco IOS Firewall

The Cisco IOS Firewall is a security-specific option that provides inspection firewall functionality and intrusion detection for every network perimeter. By delivering state-of-the-art security features such as stateful, application-based filtering; dynamic per-user authentication and authorization; and URL filtering, the Cisco IOS Firewall adds greater depth and flexibility to existing Cisco IOS security solutions including authentication, encryption, and failover.

A firewall is a physical software or hardware barrier between one part of an internal network used to control access to and from external networks. This barrier is unique because it allows predefined traffic to pass through the firewall while being monitored for protocol anomalies. The difficult part is determining the criteria by which the packets are granted or denied access through the device.

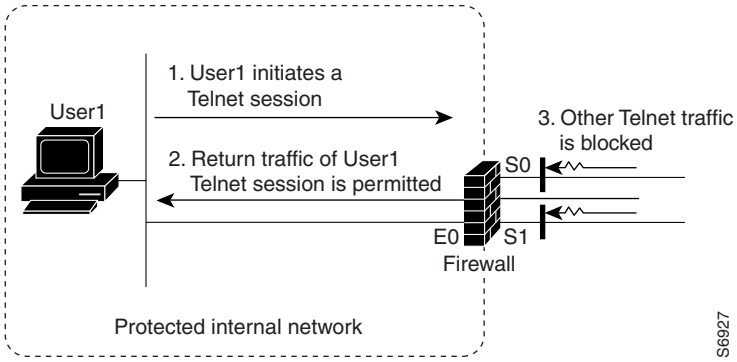
As mentioned, a firewall blocks traffic and permits other types of traffic to traverse. Firewalls are not just access control lists (ACLs); rather, they are a stateful inspection application.

Granular Protocol Inspection

The Cisco IOS Firewall performs inspections for TCP and UDP traffic. For example, TCP inspections include Telnet traffic (port 23, by default) as well as all other applications on TCP such as Hypertext Transfer Protocol (HTTP), e-mail, instant message (IM) chatter, and so on. Therefore, there is no easy way to inspect Telnet traffic alone and deny all other TCP traffic.

The Granular Protocol Inspection feature allows you to specify TCP or UDP ports using the PAM table. As a result, the Cisco IOS Firewall can restrict traffic inspections to specific applications, thereby permitting a higher degree of granularity in selecting which protocols are to be permitted and denied as shown in Figure 32.

Figure 32 Sample Topology



Benefits

The Granular Protocol Inspection feature provides the following benefits:

- Greater flexibility by allowing more granularity in the selection of protocols to be inspected
- Ease of use by providing for group inspection of multiple ports into a single, user-defined application keyword
- Enhanced functionality with the addition of more well-known ports, user-defined applications, and user-defined port ranges
- Improved performance and reduced CPU load resulting from focused inspection selections

How to Configure Granular Protocol Inspection

This section contains the following procedures:

- Defining Applications, page 633 (required)
- Setting Up Inspection Rules, page 634 (required)
- Verifying the Configuration, page 635 (optional)

Defining Applications

Perform the following task to define your applications in the PAM table by using the **ip port-map** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip port-map** *appl-name* **port** [**tcp** | **udp**] [*port_num* | **from** *begin_port_num* **to** *end_port_num*] [**list** *acl-num*] [**description** *description_string*]
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip port-map** *appl-name* **port** [**tcp** | **udp**] [*port_num* | **from** *begin_port_num* **to** *end_port_num*] [**list** *acl-num*] [**description** *description_string*] **Example:** `Router(config)# ip port-map user-10 port udp from 3400 to 3433 list 22 description "test application"`	Establishes PAM entries. **Note** When defining a user application in the PAM table, you must enter the prefix **user-**; otherwise, the following error message appears: "Unable to add port-map entry. Names for user-defined applications must start with 'user-'." **Note** Write the text string in the following format: "C *description_string* C," where "C" is a delimiting character.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Setting Up Inspection Rules

Perform the following task to set up your inspection rules by using the **ip inspect name** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip inspect name** *inspection-name protocol* [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout** *seconds*]
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip inspect name** *inspection-name protocol* [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout** *seconds*] **Example:** `Router(config)# ip inspect name abc user-10`	Defines inspection rules. **Note** Replace the *protocol* argument with the application (PAM entry) that you just defined in the previous step. In this example, it is *user-10*.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Verifying the Configuration

Perform the following task to verify your applications and inspection rules.

SUMMARY STEPS

1. **enable**
2. **show ip port-map** [*appl-name* | **port** *port-num* [**detail**]]
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip port-map** [*appl-name* | **port** *port-num* [**detail**]] **Example:** `Router# show ip port-map port 70 detail`	Establishes PAM entries.
Step 3	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Configuration Examples for Granular Protocol Inspection

This section contains the following configuration examples:

- Defining an Application for the PAM Table: Example, page 636
- Setting Up an Inspection Rule: Example, page 636
- Verifying the Configuration: Example, page 638

Defining an Application for the PAM Table: Example

In the following example from the **ip port-map** command, a user-defined application named user-10 is defined in the PAM table for five ports using the TCP protocol. Standard access list 77 is applied to define host-specific port mapping and “TEST STRING” is the description.

```
Router# configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)# ip port-map user-10 port tcp 4000 5000 6000 7000 8000 list 77 description
"TEST STRING"

Router(config)# end
```

Setting Up an Inspection Rule: Example

The following example from the **ip inspect name** command, lists user-10 as an application with the description “TEST STRING.”

```
Router# configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)# ip inspect name abc ?
```

```
  bootpc        Bootstrap Protocol Client
  bootps        Bootstrap Protocol Server
  cisco-fna     Cisco FNATIVE
  cisco-sys     Cisco SYSMAINT
  cisco-tna     Cisco TNATIVE
  cuseeme       CUSeeMe Protocol
  echo          Echo port
  esmtp         Extended SMTP
  finger        Finger
  fragment      IP fragment inspection
  ftp           File Transfer Protocol
  gopher        Gopher
  gtpv0         GPRS Tunneling Protocol Version 0
  gtpv1         GPRS Tunneling Protocol Version 1
  h323          H.323 Protocol (e.g, MS NetMeeting, Intel Video Phone)
  http          HTTP Protocol
  icmp          ICMP Protocol
  imap          IMAP Protocol
  imap3         Interactive Mail Access Protocol 3
  kerberos      Kerberos
  ldap          Lightweight Directory Access Protocol
  netbios-dgm   NETBIOS Datagram Service
  netshow       Microsoft NetShow Protocol
  nntp          Network News Transport Protocol
  parameter     Specify inspection parameters
  pop3          POP3 Protocol
  pwdgen        Password  Generator Protocol
  rcmd          R commands (r-exec, r-login, r-sh)
  realaudio     Real Audio Protocol
  rpc           Remote Prodedure Call Protocol
  rtsp          Real Time Streaming Protocol
  secure-http   Secure Hypertext Transfer Protocol
  sip           SIP Protocol
  skinny        Skinny Client Control Protocol
  smtp          Simple Mail Transfer Protocol
  snmp          Simple Network Management Protocol
  snmptrap      SNMP Trap
  sqlnet        SQL Net Protocol
  sqlsrv        SQL Service
  streamworks   StreamWorks Protocol
  tacacs        Login Host Protocol (TACACS)
  tacacs-ds     TACACS-Database Service
  tcp           Transmission Control Protocol
  telnet        Telnet
  tftp          TFTP Protocol
  udp           User Datagram Protocol
  vdolive       VDOLive Protocol
  user-10       TEST STRING<----- !user-defined application!
```

In the following example from the **ip inspect name** command, an inspection rule is established for user-10:

```
Router# configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)# ip inspect name abc user-10

Router(config)# end
```

Verifying the Configuration: Example

The following example verifies your port-map configuration:

```
Router# show running-config  | include port-map

ip port-map user-10 port tcp 4000 5000 6000 7000 8000 list 77 description "TEST STRING"
```

The following example verifies your inspection rule configuration:

```
Router# show running-config | include inspect

ip inspect name abc user-10
```

The following example displays information about the user-defined application called user-10.

```
Router# show ip port-map user-10

Host specific:    user-10              tcp port 4000...8000    in list 77    user defined
```

The following example displays detailed information about the user-defined application called user-10.

```
Router# show ip port-map user-10 detail

IP port-map entry for application 'user-10':
     tcp 4000...8000                  list 77 "TEST STRING"                  user defined
```

Additional References

The following sections provide references related to the Granular Protocol Inspection feature.

Related Documents

Related Topic	Document Title
Security commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*, Release 12.3 T
Security features including firewalls and authentication	*Cisco IOS Security Configuration Guide*, Release 12.3

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Command Reference

The following modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **ip inspect name**
- **ip port-map**
- **show ip port-map**

Glossary

CBAC—Context-Based Access Control. A protocol that provides internal users with secure access control for each application and for all traffic across network perimeters. CBAC enhances security by scrutinizing both source and destination addresses and by tracking each application's connection status.

firewall—A router or access server, or several routers or access servers, designated as a buffer between any connected public networks and a private network. A firewall router uses access lists and other methods to ensure the security of the private network.

granular—Degree of componentization. Small, fine-grained components provide greater flexibility in assembling the right combination of functionality, but can be difficult to manage.

inspection rule—A rule that specifies what IP traffic (which application-layer protocols) will be inspected by CBAC at an interface.

PAM—port-to-application mapping. A flexible, per-application port mapping capability that allows the Cisco IOS Firewall to support applications running on nonstandard ports. This feature allows network administrators to customize access control for specific applications and services, in order to meet their distinct network needs.

traffic inspection—A way that CBAC inspects traffic that travels through the firewall to discover and manage state information for TCP and UDP sessions. This state information is used to create temporary openings in the firewall's access lists to allow return traffic and additional data connections for permissible sessions (sessions that originated from within the protected internal network).

UDP—User Data Protocol. A connectionless service—there are no actual sessions, so the software approximates sessions by examining the information in the packet and determining if the packet is similar to other UDP packets (for example, similar source/destination addresses and port numbers) and if the packet was detected soon after another similar UDP packet. "Soon" means within the configurable UDP idle timeout period.

Note Refer to *Internetworking Terms and Acronyms* for terms not included in this glossary.

HTTP Inspection Engine

The HTTP Inspection Engine feature allows users to configure their Cisco IOS Firewall to detect and prohibit HTTP connections—such as tunneling over port 80, unauthorized request methods, and non-HTTP compliant file transfers—that are not authorized within the scope of the security policy configuration. Tunneling unauthorized protocols through port 80 and over HTTP exposes a network to significant security risks.

The Cisco IOS Firewall can now be configured with a security policy that adheres to the following tasks:

- Allowing specific traffic targeted for port 80 to traverse the firewall. The traffic is inspected for protocol conformance and for the types of HTTP commands that are allowed or disallowed.
- Denying specific traffic targeted for port 80 that does not comply to HTTP traffic standards. The firewall is enabled to drop the packet, reset the connection, and send a syslog message, as appropriate.

Feature History for HTTP Inspection Engine

Release	Modification
12.3(14)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Restrictions for HTTP Inspection Engine, page 642
- Information About HTTP Inspection Engine, page 642
- How to Define and Apply an HTTP Application Policy to a Firewall for Inspection, page 642
- Configuration Examples for Setting Up an HTTP Inspection Engine, page 649
- Additional References, page 650
- Command Reference, page 651

Restrictions for HTTP Inspection Engine

The Cisco 831 router with 48M RAM does not have enough memory to support this feature.

Information About HTTP Inspection Engine

Before configuring an application firewall to detect and police specific traffic targeted for port 80, you should understand the following concepts:

- What Is a Security Policy?, page 642
- Cisco IOS HTTP Application Policy Overview, page 642

What Is a Security Policy?

The application firewall uses a security policy, which consists of a collection of static signatures, to detect security violations. A static signature is a collection of parameters that specify protocol conditions that must be met before an action is taken. (For example, a signature may specify that an HTTP data stream containing the POST method must reset the connection.) These protocol conditions and reactions are defined by the end user via the command-line interface (CLI) to form a security policy.

Cisco IOS HTTP Application Policy Overview

HTTP uses port 80 to transport Internet web services, which are commonly used on the network and rarely challenged with regards to their legitimacy and conformance to standards. Because port 80 traffic is typically allowed through the network without being challenged, many application developers are leveraging HTTP traffic as an alternative transport protocol in which to enable their application to travel through or even bypass the firewall.

Most firewalls provide only packet filtering capabilities that simply permit or deny port 80 traffic without inspecting the data stream; the Cisco IOS application firewall for HTTP performs packet inspection as follows:

- Detects HTTP connections that are not authorized within the scope of the security policy configuration.
- Detects users who are tunneling applications through port 80.

If the packet is not in compliance with the HTTP protocol, it will be dropped, the connection will be reset, and a syslog message will be generated, as appropriate.

How to Define and Apply an HTTP Application Policy to a Firewall for Inspection

This section contains the following procedures:

- Defining an HTTP Application Policy, page 643
- Applying an HTTP Application Policy to a Firewall for Inspection, page 646

Defining an HTTP Application Policy

Use this task to create an HTTP application firewall policy.

Restrictions

Although application firewall policies are defined in global configuration mode, only one global policy for a given protocol is allowed per interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **appfw policy-name** *policy-name*
4. **application** *protocol*
5. **strict-http action** {**reset** | **allow**} [**alarm**]
6. **content-length** {**min** *bytes* **max** *bytes* | **min** *bytes* | **max** *bytes*} **action** {**reset** | **allow**} [**alarm**]
7. **content-type-verification** [**match-req-resp**] **action** {**reset** | **allow**} [**alarm**]
8. **max-header-length** {**request** *bytes* **response** *bytes*} **action** {**reset** | **allow**} [**alarm**]
9. **max-uri-length** *bytes* **action** {**reset** | **allow**} [**alarm**]
10. **request-method** {**rfc** *rfc-method* | **extension** *extension-method*} **action** {**reset** | **allow**} [**alarm**]
11. **port-misuse** {**p2p** | **tunneling** | **im** | **default**} **action** {**reset** | **allow**} [**alarm**
12. **transfer-encoding type** {**chunked** | **compress** | **deflate** | **gzip** | **identity** | **default**} **action** {**reset** | **allow**} [**alarm**]
13. **timeout** *seconds*
14. **audit-trail** {**on** | **off**}
15. **exit**
16. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**appfw policy-name** *policy-name* **Example:** `Router(config)# appfw policy-name mypolicy`	Defines an application firewall policy and puts the router in application firewall policy configuration mode.
Step 4	**application** *protocol* **Example:** `Router(cfg-appfw-policy)# application http`	Allows you to configure inspection parameters for a given protocol. Currently, only HTTP traffic can be inspected. • *protocol* —Specify the **http** keyword. This command puts you in appfw-policy-*protocol* configuration mode, where "*protocol*" is dependent upon the specified protocol. Because only HTTP can be specified, the configuration mode is appfw-policy-http.
Step 5	**strict-http action** {**reset** | **allow**} [**alarm**] **Example:** `Router(cfg-appfw-policy-http)# strict-http action allow alarm`	(Optional) Allows HTTP messages to pass through the firewall or resets the TCP connection when HTTP noncompliant traffic is detected.
Step 6	**content-length** {**min** *bytes* **max** *bytes* | **min** *bytes* | **max** *bytes*} **action** {**reset** | **allow**} [**alarm**] **Example:** `Router(cfg-appfw-policy-http)# content-length max 1 action allow alarm`	(Optional) Permits or denies HTTP traffic through the firewall on the basis of message size. • **min** | **max** *bytes*—Minimum or maximum content length, in bytes, allowed per message. Number of bytes range: 0 to 65535.
Step 7	**content-type-verification** [**match-req-resp**] **action** {**reset** | **allow**} [**alarm**] **Example:** `Router(cfg-appfw-policy-http)# content-type-verification match-req-resp action allow alarm`	(Optional) Permits or denies HTTP traffic through the firewall on the basis of content message type.
Step 8	**max-header-length** {**request** *bytes* **response** *bytes*} **action** {**reset** | **allow**} [**alarm**] **Example:** `Router(cfg-appfw-policy-http)# max-header-length request 1 response 1 action allow alarm`	(Optional) Permits or denies HTTP traffic on the basis of the message header length. • *bytes*—Number of bytes ranging from 0 to 65535.
Step 9	**max-uri-length** *bytes* **action** {**reset** | **allow**} [**alarm**] **Example:** `Router(cfg-appfw-policy-http)# max-uri-length 1 action allow alarm`	(Optional) Permits or denies HTTP traffic on the basis of the URI length in the request message.

	Command or Action	Purpose
Step 10	**request method** {**rfc** *rfc-method* | **extension** *extension-method*} **action** {**reset** | **allow**} [**alarm**] **Example:** `Router(cfg-appfw-policy-http)# request-method rfc default action allow alarm`	(Optional) Permits or denies HTTP traffic according to either the request methods or the extension methods. • **rfc**—Specifies that the supported methods of RFC 2616, *Hypertext Transfer Protocol—HTTP/1.1*, are to be used for traffic inspection. • *rfc-method*—Any one of the following RFC 2616 methods can be specified: **connect**, **default, delete**, **get**, **head**, **options**, **post**, **put**, **trace**. • **extension**—Specifies that the extension methods are to be used for traffic inspection. • *extension-method*—Any one of the following extension methods can be specified: **copy**, **default, edit**, **getattribute**, **getproperties**, **index**, **lock**, **mkdir**, **move**, **revadd**, **revlabel**, **revlog**, **save**, **setattribute**, **startrev**, **stoprev**, **unedit**, **unlock**.
Step 11	**port-misuse** {**p2p** | **tunneling** | **im** | **default**} **action** {**reset** | **allow**} [**alarm**] **Example:** `Router(cfg-appfw-policy-http)# port-misuse default action allow alarm`	(Optional) Permits or denies HTTP traffic through the firewall on the basis of specified applications in the HTTP message. • **p2p**—Peer-to-peer protocol applications subject to inspection: Kazaa and Gnutella. • **tunneling**—Tunneling applications subject to inspection: HTTPPort/HTTPHost, GNU Httptunnel, GotoMyPC, Firethru, Http-tunnel.com Client • **im**—Instant messaging protocol applications subject to inspection: Yahoo Messenger. • **default**—All applications are subject to inspection.
Step 12	**transfer-encoding type** {**chunked** | **compress** | **deflate** | **gzip** | **identity** | **default**} **action** {**reset** | **allow**} [**alarm**] **Example:** `Router(cfg-appfw-policy-http)# transfer-encoding type default action allow alarm`	(Optional) Permits or denies HTTP traffic according to the specified transfer-encoding of the message. • **chunked**—Encoding format (specified in RFC 2616, *Hypertext Transfer Protocol—HTTP/1*) in which the body of the message is transferred in a series of chunks; each chunk contains its own size indicator. • **compress**—Encoding format produced by the UNIX "compress" utility. • **deflate**—"ZLIB" format defined in RFC 1950, *ZLIB Compressed Data Format Specification version 3.3*, combined with the "deflate" compression mechanism described in RFC 1951, *DEFLATE Compressed Data Format Specification version 1.3*. • **gzip**—Encoding format produced by the "gzip" (GNU zip) program. • **identity**—Default encoding, which indicates that no encoding has been performed. • **default**—All of the transfer encoding types.

	Command or Action	Purpose
Step 13	**timeout** *seconds* **Example:** `Router(cfg-appfw-policy-http)# timeout 60`	(Optional) Overrides the global TCP idle timeout value for HTTP traffic. **Note** If this command is not issued, the default value specified via the **ip inspect tcp idle-time** command will be used.
Step 14	**audit-trail** {**on** | **off**} **Example:** `Router(cfg-appfw-policy-http)# audit-trail on`	(Optional) Turns audit trail messages on or off. **Note** If this command is not issued, the default value specified via the **ip inspect audit-trail** command will be used.
Step 15	**exit** **Example:** `Router(cfg-appfw-policy-http)# exit`	Exits cfg-appfw-policy-http configuration mode.
Step 16	**exit** **Example:** `Router(cfg-appfw-policy)# exit`	Exits cfg-appfw-policy configuration mode.

What to Do Next

After you have successfully defined an application policy for HTTP traffic inspection, you must apply the policy to an inspection rule. Thereafter, the inspection rule must be applied to an interface. For information on completing this task, see the section "Applying an HTTP Application Policy to a Firewall for Inspection."

Applying an HTTP Application Policy to a Firewall for Inspection

Use this task to apply an HTTP application policy to an inspection rule, followed by applying the inspection rule to an interface.

Note An application policy can coexist with other inspection protocols (for example, an HTTP policy and an FTP policy can coexist).

Prerequisites

You must have already defined an application policy (as shown in the section "Defining an HTTP Application Policy").

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip inspect name** *inspection-name* **appfw** *policy-name*
4. **ip inspect name** *inspection-name* **http** [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout** *seconds*]

5. **interface** *type number*
6. **ip inspect** *inspection-name* {**in** | **out**}
7. **exit**
8. **exit**
9. **show appfw configuration** [*name*]

   or

   show ip inspect {**name** *inspection-name* | **config** | **interfaces** | **session** [**detail**] | **statistics** | **all**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip inspect name** *inspection-name* **appfw** *policy-name* **Example:** `Router(config)# ip inspect name firewall appfw mypolicy`	Defines a set of inspection rules for the application policy. • *policy-name*—Must match the policy name specified via the **appfw policy-name** command.
Step 4	**ip inspect name** *inspection-name* **http** [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout** *seconds*] **Example:** `Router(config)# ip inspect name firewall http`	Defines a set of inspection rules that is to be applied to all HTTP traffic. • The *inspection-name* argument must match the *inspection-name* argument specified in Step 3.
Step 5	**interface** *type number* **Example:** `Router#(config)# interface FastEthernet0/0`	Configures an interface type and enters interface configuration mode.
Step 6	**ip inspect** *inspection-name* {**in** | **out**} **Example:** `Router#(config-if)# ip inspect firewall in`	Applies the inspection rules (defined in Step 3 and Step 4) to all traffic entering the specified interface. • The *inspection-name* argument must match the inspection name defined via the **ip inspect name** command.
Step 7	**exit** **Example:** `Router#(config-if)# exit`	Exits interface configuration mode.

	Command or Action	Purpose
Step 8	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 9	**show appfw configuration** [*name*] **Example:** `Router# show appfw configuration` or **show ip inspect** {**name** *inspection-name* | **config** | **interfaces** | **session** [**detail**] | **statistics** | **all**} **Example:** `Router# show ip inspect config`	(Optional) Displays application firewall policy configuration information. (Optional) Displays firewall-related configuration information.

Troubleshooting Tips

To help troubleshoot the application firewall configuration, issue the following application-firewall specific debug command: **debug appfw** {**application** *protocol* | **function-trace** | **object-creation** | **object-deletion** | **events** | **timers** | **detailed**}.

The following sample configuration shows how to configure an HTTP policy with application firewall debugging enabled:

```
Router(config)# appfw policy-name myPolicyAPPFW  FUNC:appfw_policy_find
APPFW  FUNC:appfw_policy_find -- Policy myPolicy is not found
APPFW  FUNC:appfw_policy_alloc
APPFW  FUNC:appfw_policy_alloc -- policy_alloc 0x65727278
APPFW  FUNC:appfw_policy_alloc -- Policy 0x65727278 is set to valid
APPFW  FUNC:appfw_policy_alloc -- Policy myPolicy has been created
APPFW  FUNC:appfw_policy_command -- memlock policy 0x65727278

! Debugging sample for application (HTTP) creation

Router(cfg-appfw-policy)# application httpAPPFW  FUNC:appfw_http_command
APPFW  FUNC:appfw_http_appl_find
APPFW  FUNC:appfw_http_appl_find -- Application not found
APPFW  FUNC:appfw_http_appl_alloc
APPFW  FUNC:appfw_http_appl_alloc -- appl_http 0x64D7A25C
APPFW  FUNC:appfw_http_appl_alloc -- Application HTTP parser structure 64D7A25C created

! Debugging sample for HTTP-specific application inspection
Router(cfg-appfw-policy-http)#
Router(cfg-appfw-policy-http)# strict-http action reset alarm
APPFW  FUNC:appfw_http_subcommand
APPFW  FUNC:appfw_http_subcommand -- strict-http cmd turned on

Router# debug appfw detailed

APPFW Detailed Debug debugging is on
fw7-7206a#debug appfw object-creation
APPFW Object Creations debugging is on
fw7-7206a#debug appfw object-deletion
APPFW Object Deletions debugging is on
```

Configuration Examples for Setting Up an HTTP Inspection Engine

This section contains the following configuration example:

- Setting Up and Verifying an HTTP Inspection Engine: Example, page 649

Setting Up and Verifying an HTTP Inspection Engine: Example

The following example show how to define the HTTP application firewall policy "mypolicy." This policy includes all supported HTTP policy rules. This example also includes sample output from the **show appfw configuration** and **show ip inspect config** commands, which allow you to verify the configured setting for the application policy.

```
! Define the HTTP policy.
appfw policy-name mypolicy
 application http
  strict-http action allow alarm
  content-length maximum 1 action allow alarm
  content-type-verification match-req-rsp action allow alarm
  max-header-length request 1 response 1 action allow alarm
  max-uri-length 1 action allow alarm
  port-misuse default action allow alarm
  request-method rfc put action allow alarm
  transfer-encoding type default action allow alarm
!
!
! Apply the policy to an inspection rule.
ip inspect name firewall appfw mypolicy
ip inspect name firewall http
!
!
! Apply the inspection rule to all HTTP traffic entering the FastEthernet0/0 interface.
interface FastEthernet0/0
 ip inspect firewall in
!
!
! Issue the show appfw configuration command and the show ip inspect config command after
the inspection rule "mypolicy" is applied to all incoming HTTP traffic on the
FastEthernet0/0 interface.
!
Router# show appfw configuration

Application Firewall Rule configuration
  Application Policy name mypolicy
    Application http
      strict-http action allow alarm
      content-length minimum 0 maximum 1 action allow alarm
      content-type-verification match-req-rsp action allow alarm
      max-header-length request length 1 response length 1 action allow alarm
      max-uri-length 1 action allow alarm
      port-misuse default action allow alarm
      request-method rfc put action allow alarm
      transfer-encoding default action allow alarm

Router# show ip inspect config

Session audit trail is disabled
Session alert is enabled
```

```
one-minute (sampling period) thresholds are [400:500] connections
max-incomplete sessions thresholds are [400:500]
max-incomplete tcp connections per host is 50. Block-time 0 minute.
tcp synwait-time is 30 sec -- tcp finwait-time is 5 sec
tcp idle-time is 3600 sec -- udp idle-time is 30 sec
dns-timeout is 5 sec
Inspection Rule Configuration
Inspection name firewall
http alert is on audit-trail is off timeout 3600
```

Additional References

The following sections provide references related to the HTTP Inspection Engine feature.

Related Documents

Related Topic	Document Title
Firewall commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*, Release 12.3T

Standards

Standards	Title
No new or modified standards are supported by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2616	*Hypertext Transfer Protocol -- HTTP/1.1*

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

New Commands

- **appfw policy-name**
- **application**
- **audit-trail**
- **content-length**
- **content-type-verification**
- **debug appfw**
- **max-header-length**
- **max-uri-length**
- **port-misuse**
- **request-method**
- **show appfw**
- **strict-http**
- **timeout**
- **transfer-encoding type**

Modified Command

- **ip inspect name**

Inspection of Router-Generated Traffic

The Inspection of Router-Generated Traffic feature allows Context-Based Access Control (CBAC) to inspect traffic that is originated by or destined to the router on which CBAC is configured. Previously, inspection of Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and H.323 connections initiated by or destined to the router were allowed.

Feature History for Inspection of Router-Generated Traffic

Release	Modification
12.3(14)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for Inspection of Router-Generated Traffic, page 653
- Restrictions for Inspection of Router-Generated Traffic, page 654
- Information About Inspection of Router-Generated Traffic, page 654
- How to Configure Inspection of Router-Generated Traffic, page 655
- Configuration Examples for Inspection of Router-Generated Traffic, page 660
- Additional References, page 660
- Command Reference, page 661
- Glossary, page 662

Prerequisites for Inspection of Router-Generated Traffic

- Configure CBAC.
- Configure Cisco Call Manager Express (CCME) or H.323 Gateway to configure the inspection of H.323 connections to and from the router.

Restrictions for Inspection of Router-Generated Traffic

- Inspection of router-generated traffic is supported only on the following protocols: H.323, TCP, and UDP.
- The Cisco IOS Firewall supports only Version 2 of the H.323 protocol. If CCME or the H.323 Gateway has inspection of H.323 router traffic enabled, enter the following commands so that it is configured to support only Version 2 features:

```
voice service voip
h323
session transport tcp calls-per-connection 1
h245 tunnel disable
h245 caps mode restricted
h225 timeout tcp call-idle value 0
```

Information About Inspection of Router-Generated Traffic

To configure Inspection of Router-Generated Traffic, you need to understand the following concepts:

- CBAC, page 654
- Inspection of Router-Generated Traffic Overview, page 655

CBAC

CBAC is a Cisco IOS Firewall set feature that provides network protection by using the following functions:

- Traffic Filtering
- Traffic Inspection
- Alerts and Audit Trails
- Intrusion Detection

Traffic Filtering

CBAC filters TCP and UDP packets based on application-layer protocol session information. You can configure CBAC to permit specified TCP and UDP traffic through a firewall only when the connection is initiated from within the network you want to protect. CBAC can inspect traffic for sessions that originate from either side of the firewall, and CBAC can be used for intranet, extranet, and Internet perimeters of your network.

Traffic Inspection

CBAC inspects traffic that travels through the firewall to discover and manage state information for TCP and UDP sessions. This state information is used to create temporary openings in the firewall's access lists to allow return traffic and additional data connections for permissible sessions.

Alerts and Audit Trails

CBAC generates real-time alerts and audit trails. Enhanced audit trail features use SYSLOG to track all network transactions; it records time stamps, the source host, the destination host, the ports used, and the total number of transmitted bytes, for advanced, session-based reporting. Real-time alerts send SYSLOG error messages to central management consoles upon detecting suspicious activity.

Using CBAC inspection rules, you can configure alerts and audit trail information on a per-application protocol basis. For example, if you want to generate audit trail information for HTTP traffic, you can specify that in the CBAC rule covering HTTP inspection.

Intrusion Detection

CBAC provides a limited amount of intrusion detection to protect against specific Simple Mail Transfer Protocol (SMTP) attacks. With intrusion detection, SYSLOG messages are reviewed and monitored for specific "attack signatures." Certain types of network attacks have specific characteristics, or signatures. When CBAC detects an attack, it resets the offending connections and sends SYSLOG information to the SYSLOG server.

Inspection of Router-Generated Traffic Overview

Inspection of Router-Generated Traffic enhances CBAC's functionality to inspect TCP, UDP, and H.323 connections that have a router or firewall as one of the connection endpoints. This enables CBAC to open pinholes for TCP, UDP, and H.323 control channel connections to and from the router, and to open pinholes for data and media channels negotiated over the H.323 control channels.

Inspection of TCP and UDP channels initiated from the router enables dynamic opening of pinholes on the interface access control list (ACL) to allow return traffic. You do not have to modify the ACL when a TCP connection such as Telnet is made from the router.

Inspection of local H.323 connections enables the deployment of CCME, H.323 gateway, and the Cisco IOS Firewall on the same router. This also simplifies ACL configuration on CCME's interface through which H.323 connections are made. Before this feature, in addition to configuring ACLs to allow H.323 connections on a standard port (for example, port 1720), you had to configure ACLs to allow all dynamically negotiated data and media channels. With this feature you just configure the ACLs to allow H.323 control channels on port 1720. The Cisco IOS Firewall inspects all the traffic on the control channel and opens pinholes to allow dynamically negotiated data and media channels.

To enable Inspection of Router-Generated Traffic, specify the **router-traffic** keyword in the **ip inspect name** command of the appropriate protocol.

How to Configure Inspection of Router-Generated Traffic

This section contains the following procedures:

- Configuring H.323 Inspection, page 655 (required)
- Configuring CBAC, page 656 (required)
- Verifying the CBAC Configuration, page 658 (optional)

Configuring H.323 Inspection

To configure the H.323 protocol, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **ip inspect name** *inspection-name* {**TCP | UDP | H323**} [**alert** {**on | off**}] [**audit-trail** {**on | off**}][**router-traffic**][**timeout** *seconds*]
4. **interface** *type slot/port*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip inspect name** *inspection-name* {**TCP** | **UDP** | **H323**} [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}][**router-traffic**][**timeout** *seconds*] **Example:** `Router(config)# ip inspect name test H.323 router-traffic`	Defines a set of inspection rules.
Step 4	**interface** *type slot/port* **Example:** `Router(config)# interface FE 0/0`	Configures an interface type.
Step 5	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Configuring CBAC

To configure CBAC, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**deny | permit**} *source* [*source-wildcard*] [**log**]
4. **ip inspect name** *inspection-name* {**TCP | UDP | H323**} [**alert** {**on | off**}] [**audit-trail** {**on | off**}][**router-traffic**][**timeout** *seconds*]
5. **interface** *type slot/port*

6. **ip inspect** *inspection-name* {**in** | **out**}
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] [**log**] **Example:** `Router(config)# access-list 121 permit tcp host 100.168.11.1 any eq 1720`	Defines a standard IP access list.
Step 4	**ip inspect name** *inspection-name* {**TCP** | **UDP** | **H323**} [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}][**router-traffic**][**timeout** *seconds*] **Example:** `Router(config)# ip inspect name here H323 router-traffic timeout 180`	Defines a set of inspection rules.
Step 5	**interface** *type slot/port* **Example:** `Router(config)# Serial0/3/0`	Configures an interface type.
Step 6	**ip inspect** *inspection-name* {**in** | **out**} **Example:** `Router(config-if)# ip inspect test in`	Enables the Cisco IOS Firewall on an interface.
Step 7	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Verifying the CBAC Configuration

To verify the CBAC configuration, perform the following steps.

SUMMARY STEPS

1. **show ip inspect name** *inspection-name*
2. **show ip inspect config**
3. **show ip inspect interfaces**
4. **show ip inspect session** [**detail**]
5. **show ip inspect all**

DETAILED STEPS

Step 1 **show ip inspect name** *inspection-name*

Use this command to show a particular configured inspection rule. The following example configures the inspection rule myinspectionrule. The output shows the protocols that should be inspected by CBAC and the corresponding idle timeouts for each protocol.

```
Router# show ip inspect name myinspectionrule

Inspection Rule Configuration

 Inspection name myinspectionrule
  tcp timeout 3600
  udp timeout 30
  ftp timeout 3600
```

Step 2 **show ip inspect config**

Use this command to show the CBAC configuration, including global timeouts, thresholds, and inspection rules.

```
Router# show ip inspect config

Session audit trail is disabled
one-minute (sampling period) thresholds are [400:500] connections
max-incomplete sessions thresholds are [400:500]
max-incomplete tcp connections per host is 50. Block-time 0 minute.
tcp synwait-time is 30 sec -- tcp finwait-time is 5 sec
tcp idle-time is 3600 sec -- udp idle-time is 30 sec
dns-timeout is 5 sec
 Inspection Rule Configuration
  inspection name myinspectionrule
  tcp timeout 3600
  udp timeout 30
  ftp timeout 3600
```

Step 3 **show ip inspect interfaces**

Use this command to show the interface configuration with respect to applied inspection rules and access lists.

```
Router# show ip inspect interfaces

Interface Configuration
 Interface Ethernet0
  Inbound inspection rule is myinspectionrule
```

```
    tcp timeout 3600
    udp timeout 30
    ftp timeout 3600
  Outgoing inspection rule is not set
  Inbound access list is not set
  Outgoing access list is not set
```

Step 4 show ip inspect session detail

Use this command to display existing sessions that CBAC is currently tracking and inspecting. The following sample output shows that an outgoing ACL and an inbound ACL (dynamic ACLs) have been created to allow return traffic.

```
Router# show ip inspect session detail

Established Sessions
 Session 80E87274 (192.168.1.116:32956)=>(192.168.101.115:23) tcp SIS_OPEN
   Created 00:00:08, Last heard 00:00:04
   Bytes sent (initiator:responder) [140:298] acl created 2
   Outgoing access-list 102 applied to interface FastEthernet0/0
   Inbound access-list 101 applied to interface FastEthernet0/1
```

Step 5 show ip inspect all

Use this command to show all CBAC configuration and all existing sessions that are currently being tracked and inspected by CBAC.

```
Router# show ip inspect all

Session audit trail is disabled
one-minute (sampling period) thresholds are [400:500] connections
max-incomplete sessions thresholds are [400:500]
max-incomplete tcp connections per host is 50. Block-time 0 minute.
tcp synwait-time is 30 sec -- tcp finwait-time is 5 sec
tcp idle-time is 3600 sec -- udp idle-time is 30 sec
dns-timeout is 5 sec
Inspection Rule Configuration
 Inspection name all
    tcp timeout 3600
    udp timeout 30
    ftp timeout 3600
Interface Configuration
 Interface Ethernet0
  Inbound inspection rule is all
    tcp timeout 3600
    udp timeout 30
    ftp timeout 3600
  Outgoing inspection rule is not set
  Inbound access list is not set
  Outgoing access list is not set
 Established Sessions
 Session 25A6E1C (30.0.0.1:46065)=>(40.0.0.1:21) ftp SIS_OPEN
 Session 25A34A0 (40.0.0.1:20)=>(30.0.0.1:46072) ftp-data SIS_OPEN
```

Configuration Examples for Inspection of Router-Generated Traffic

This section provides the following configuration examples:

- Configuring CBAC with Inspection of H.323 Traffic: Example, page 660

Configuring CBAC with Inspection of H.323 Traffic: Example

These commands create the ACL. In this example, TCP traffic from subnet 100.168.11.1, 192.168.11.50, and 192.168.100.1 is permitted.

```
access-list 120 permit tcp host 100.168.11.1 any eq 1720
access-list 121 permit tcp host 192.168.11.50 host 100.168.11.1 eq 1720
access-list 121 permit tcp host 192.168.100.1 host 100.168.11.1 eq 1720
```

These commands create the CBAC inspection rule LOCAL-H323, allowing inspection of the protocol traffic specified by the rule. This inspection rule sets the timeout value to 180 seconds for each protocol (except for RPC). The timeout value defines the maximum time that a connection for a given protocol can remain active without any traffic passing through the router. When these timeouts are reached, the dynamic ACLs that are inserted to permit the returning traffic are removed, and subsequent packets (possibly even valid ones) are not permitted.

```
ip inspect name LOCAL-H323 tftp timeout 180
ip inspect name LOCAL-H323 h323 router-traffic timeout 180
```

These commands apply the inspection rule and ACL. In this example, the inspection rule LOCAL-H323 is applied to traffic at interface Serial0/3/0.

```
interface Serial0/3/0
 ip address 11.168.11.2 255.255.255.0
 ip access-group 121 in
 ip access-group 120 out
 ip inspect LOCAL-H323 in
 ip inspect LOCAL-H323 out
 encapsulation frame-relay
frame-relay map ip 11.168.11.1 168 broadcast
 no frame-relay inverse-arp
 frame-relay intf-type dce
```

Additional References

The following sections provide references related to Inspection of Router-Generated Traffic.

Related Documents

Related Topic	Document Title
CBAC	*Cisco IOS Security Configuration Guide*, Release 12.3 *Cisco IOS Security Command Reference*, Release 12.3T
H.323	*Cisco IOS H.323 Configuration Guide,* Release 12.3

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following modified command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **ip inspect name**

Glossary

CBAC—Context-Based Access Control. Scrutinizes source and destination addresses to enhance security for TCP and UDP applications that use well-known ports, such as FTP and e-mail traffic.

firewall—One or more router or access servers designated as a buffer between any connected public networks and a private network. A firewall router uses access lists and other methods to ensure the security of the private network.

FTP—File Transfer Protocol. An application protocol, part of the TCP/IP protocol stack, for transferring files between network nodes.

H.323—A multimedia conferencing protocol that includes voice, video, and data conferencing for use over packet-switched networks. H.323 allows dissimilar communication devices to communicate with each other by using a standardized communication protocol.

IMAP—Internet Message Access Protocol. A method of accessing e-mail or bulletin board messages kept on a mail server that can be shared. IMAP permits client e-mail applications to access remote message stores as if they were local without actually transferring the message.

IP—Internet Protocol. Connectionless protocol at the network layer (Layer 3) of the OSI reference model. Provides features for addressing, type-of-service specification, fragmentation and reassemble, and security. IP works with TCP and is usually identified as TCP/IP.

POP—Post Office Protocol. A protocol that client e-mail applications use to retrieve mail from a mail server.

SMTP—Simple Mail Transfer Protocol. A simple ASCII protocol that describes the exchange of e-mail between two message-transfer agents using TCP/IP.

TCP—Transmission Control Protocol. A connection-oriented transport-layer protocol that provides reliable full-duplex data transmissions.

TCP/IP—Transmission Control Protocol/Internet Protocol. Common name for the suite of protocols developed by the U.S. Department of Defense in the 1970s to support the construction of worldwide internetworks. TCP and IP are the two best-known protocols in the suite.

UDP—User Datagram Protocol. A connectionless transport-layer protocol for exchanging datagrams without acknowledgments or guaranteed delivery.

VoIP—Voice over IP. Capability of carrying normal telephony-style voice over an IP network with circuit-based telephone-like functionality, reliability, and voice quality. VoIP generally refers to the Cisco standards-based (H.323 and so forth) approach to IP voice traffic.

Note Refer to *Internetworking Terms and Acronyms* for terms not included in this glossary.

Transparent Cisco IOS Firewall

The Transparent Cisco IOS Firewall feature allows users to “drop” a Cisco IOS Firewall in front of their existing network without changing the statically defined IP addresses of their network-connected devices. Thus, users can allow selected devices from a subnet to traverse the firewall while access to other devices on the same subnet is denied.

Feature History for Transparent Cisco IOS Firewall

Release	Modification
12.3(7)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Restrictions for Transparent Cisco IOS Firewall, page 664
- Information About Transparent Cisco IOS Firewall, page 664
- How to Configure a Transparent Cisco IOS Firewall, page 665
- Configuration Examples for Transparent Cisco IOS Firewall, page 671
- Additional References, page 17
- Additional References, page 679
- Command Reference, page 680

Restrictions for Transparent Cisco IOS Firewall

Layer 3 IP Packet Support Only

Only IP packets (TCP, User Datagram Protocol [UDP] , and Internet Control Message Protocol [ICMP]) are subjected to inspection by the transparent firewall. Non-IP traffic is bridged as usual without interference from the transparent firewall. However, if users wish to block non-IP traffic, the MAC access control lists (ACLs) can be applied on interfaces to filter out non-IP traffic and allow only IP traffic.

The following example shows how to configure an ACL that permits all IP packets (0x0800) into the Ethernet interface but denies all Internetwork Packet Exchange (IPX) packets (0x8137):

```
Router(config)# access-list 201 permit 0x0800
Router(config)# access-list 201 permit 0x8137
Router(config)# interface ethernet 0
Router(config-if)# bridge-group 1 input-type-list 201
```

VLAN Trunk Bridging

Bridging between VLAN trunks works only for dot1q encapsulation; Inter-Switch Link (ISL) encapsulation will not work. (However, ISL VLANs will work if subinterfaces are created and placed in a bridge group.)

Information About Transparent Cisco IOS Firewall

To use a transparent Cisco IOS Firewall in your network, you should understand the following concepts:

- Benefit of the Transparent Firewall, page 2
- Transparent Firewall Overview, page 2
- Layer 2 and Layer 3 Firewalls Configured on the Same Router, page 3

Benefit of the Transparent Firewall

Added Security with Minimum Configuration

Users can simply drop a transparent Cisco IOS Firewall into an existing network without having to reconfigure their statically defined devices. Thus, the tedious and costly overhead that is required to renumber devices on the trusted network is eliminated.

Transparent Firewall Overview

A typical Cisco IOS Firewall is a Layer 3 device with trusted and untrusted interfaces on different IP subnets. A Layer 3 firewall works well with Cisco IOS devices that function as routers with preexisting subnet separations. However, when a Layer 3 firewall is placed in an existing network, the network IP addresses must be reconfigured to accommodate the firewall.

A transparent Cisco IOS firewall acts as a Layer 2 transparent bridge with context-based access control (CBAC) and ACLs configured on the bridged interface. Because the Layer 2 firewall intercepts packets at Layer 2 and is “transparent” to the existing network, Layer 3 firewall limitations are not applicable.

Transparent Bridging Overview

If configured for bridging, a Cisco IOS device can bridge any number of interfaces. The device can complete basic bridging tasks such as learning MAC addresses on ports to restrict collision domains and running Spanning Tree Protocol (STP) to prevent looping in the topology.

Within bridging, a user can configure Integrated Routed Bridging (IRB), which allows a device to bridge on some interfaces while a Layer 3 Bridged Virtual Interface (BVI) is presented for routing. The bridge can determine whether the packet is to be bridged or routed on the basis of the destination of the Layer 2 or Layer 3 IP address in the packet. Configured with an IP address, the BVI can manage the device even if there is no interface configured for routing.

Layer 2 and Layer 3 Firewalls Configured on the Same Router

A transparent firewall supports a BVI for routing, so a packet that comes in on a bridged interface can be bridged or routed out of the BVI. This functionality allows a Layer 2 (transparent) firewall and a Layer 3 firewall to be configured on the same router: The transparent firewall operates on the bridged packets while the "normal" firewall operates on the routed packets. For example, if you have six interfaces on your router and two of them are in a bridge group, you can simultaneously configure and run normal inspection on the remaining four interfaces.

How to Configure a Transparent Cisco IOS Firewall

You configure a transparent firewall just as you would configure a Layer 3 firewall (via the **ip inspect** command, which can be configured on any of the bridged interfaces for the transparent firewall). Also, you configure transparent bridging for a firewall just as you would for any other Cisco IOS device.

This section contains the following procedures:

- Configuring a Bridge Group, page 3 (required)
- Configuring Inspection and ACLs, page 6 (required)
- Forwarding DHCP Traffic, page 8 (optional)
- Monitoring Transparent Firewall Events, page 8 (optional)

Configuring a Bridge Group

Use this task to configure a bridge group and to associate interfaces or subinterfaces in the configured bridge group.

BVI Configuration Requirements

- If a BVI is not configured, you must disable IP routing (via the **no ip routing** command) for the bridging operation to take effect.
- If configured, a BVI must be configured with an IP address in the same subnet.
- You *must* configure a BVI if more than two interfaces are placed in a bridge group.

Restrictions

- If more than two interfaces are assigned to a bridge group, any routers that are acting as first-hop gateways to hosts that are in the bridged network (the bridge group) must allow ICMP time-to-live (TTL) exceeded messages to pass.
- Spanning Tree Bridge Protocol Data Units (BPDU) and packets that are to be routed out of the bridge, if IRB is configured, are not inspected.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bridge** *bridge-group* **protocol** {**dec** | **ibm** | **ieee** | **vlan-bridge**}
4. **interface** *type number*
5. **bridge-group** *bridge-group*
6. **exit**
7. **bridge irb**
8. **bridge** *bridge-group* **route** *protocol*
9. **interface** *type number*
10. **ip address** *ip-address mask*
11. **no shutdown**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**bridge** *bridge-group* **protocol** {**dec** | **ibm** | **ieee** | **vlan-bridge**} **Example:** `Router(config)# bridge 1 protocol ieee`	Defines the type of Spanning Tree Protocol (STP).
Step 4	**interface** *type number* **Example:** `Router(config)# interface Ethernet0`	Configures an interface type and enters interface configuration mode.

	Command or Action	Purpose
Step 5	**bridge-group** *bridge-group* **Example:** `Router(config-if)# bridge-group 1`	Assigns each network interface to a bridge group. **Note** Complete Step 4 and Step 5 for each interface you want to assign to a bridge group. **Note** You can also assign subinterfaces to a bridge group to control bridging between VLANs.
Step 6	**exit**	Exits interface configuration mode.
Step 7	**bridge irb** **Example:** `Router(config)# bridge irb`	Enables the Cisco IOS software to route a given protocol between routed interfaces and bridge groups or to route a given protocol between bridge groups. **Note** Step 7 through Step 11 are necessary only if you want to configure a BVI.
Step 8	**bridge** *bridge-group* **route** *protocol* **Example:** `Router(config)# bridge 1 route ip`	Enables the routing of a specified protocol in a specified bridge group.
Step 9	**interface** *type number* **Example:** `Router(config)# interface BVI1`	Configures a BVI and enters interface configuration mode.
Step 10	**ip address** *ip-address mask* **Example:** `Router(config-if) ip address 10.1.1.1 255.255.255.0`	Sets a primary IP address for an interface.
Step 11	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Restarts a disabled interface.

Examples

The following example shows how to configure interfaces “ethernet0” and “ethernet1” in a bridge group. These interfaces are associated with the BVI interface “BVI1,” which can be reached from any host on either of the interfaces via the IP address 10.1.1.1.

```
Router(config)# bridge 1 protocol ieee
Router(config)# interface ethernet0
Router(config-if)# bridge-group 1
Router(config-if)# interface ethernet1
Router(config-if)# bridge-group 1
Router(config-if)# exit
! Configure the BVI.
Router(config)# bridge irb
Router(config)# bridge 1 route ip
Router(config)# interface BVI1
Router(config-if)# ip address 10.1.1.1 255.255.255.0
Router(config-if)# no shutdown
```

Troubleshooting Tips

To display the status of each bridge group, use the **show bridge-group** command or to display entries in the bridge table, use the **show bridge** command.

What to Do Next

After you have configured the bridge group, you must configure an inspection rule and at least one IP ACL. To complete this task, refer to the following section, "Configuring Inspection and ACLs."

Note If inspection is not configured on any interface in the bridge group, IP ACLs on bridged interfaces will not be active.

Configuring Inspection and ACLs

Use this task to configure an inspection rule and apply it on the appropriate interface. Also, use this task to configure at least one ACL and apply it on one or more of the interfaces that you configured in the bridge group.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip inspect name** *protocol* [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout** *seconds*]
4. **interface** *type number*
5. **ip inspect** *inspection-name* {**in** | **out**}
6. **exit**
7. **access-list** *access-list-number* {**permit** | **deny**} {*type-code wild-mask* | *address mask*}
8. **interface** *type number*
9. **ip access-group** {*access-list-number* | *access-list-name*} {**in** | **out**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip inspect name** *protocol* [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout** *seconds*] **Example:** `Router(config)# ip inspect name test tcp`	Defines a set of inspection rules.
Step 4	**interface** *type number* **Example:** `Router(config)# interface Ethernet0`	Configures an interface type and enters interface configuration mode.
Step 5	**ip inspect** *inspection-name* {**in** | **out**} **Example:** `Router(config-if)# ip inspect test in`	Applies a set of inspection rules to an interface.
Step 6	**exit**	Exits interface configuration mode.
Step 7	**access-list** *access-list-number* {**permit** | **deny**} {*type-code wild-mask* | *address mask*} **Example:** `Router(config)# access-list 156 permit 10.1.1.0 0.0.0.255 any`	Configures the ACL. **Note** Repeat this step for each ACL that you want to configure.
Step 8	**interface** *type number* **Example:** `Router(config)# interface Ethernet0`	Configures an interface type and enters interface configuration mode. **Note** Repeat Steps 8 and 9 for each ACL that you want to apply to inbound packets from a specific interface.
Step 9	**ip access-group** {*access-list-number* | *access-list-name*} {**in** | **out**} **Example:** `Router(config-if) ip access-group 156 in`	Controls access to an interface.

Examples

The following example shows how to configure an inspection rule on interface "ethernet0," which is the inside interface. Policies can be specified via ACL 156 or 101; for example, ACL 156 can be used to specify that rlogin and rsh are not allowed for the internal users, and ACL 101 can be used to specify that an external host requires connectivity to a particular host in the internal domain.

```
Router(config)# ip inspect name test tcp
Router(config)# interface ethernet0
Router(config-if)# ip inspect test in
Router(config-if)# exit
!
! Configure the ACLs.
Router(config)# access-list 101 deny ip any any
Router(config)# access-list 156 permit 10.1.1.0 0.0.0.255 any
Router(config)# access-list 156 deny ip any any
```

```
Router(config)# interface ethernet0
Router(config-if) ip access-group 156 in
Router(config)# interface ethernet1
Router(config-if) ip access-group 101 in
```

Forwarding DHCP Traffic

Use this task to enable a transparent firewall to forward DHCP packets across the bridge without inspection; that is, the **ip inspect L2-transparent dhcp-passthrough** command overrides the ACL for DHCP packets, so DHCP packets will be forwarded even if the ACL is configured to deny all IP packets. Thus, clients on one side of the bridge can get an IP address from a DHCP server on the opposite side of the bridge.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip inspect L2-transparent dhcp-passthrough**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip inspect L2-transparent dhcp-passthrough** **Example:** `Router#(config) ip inspect L2-transparent dhcp-passthrough`	Allows a transparent firewall to forward DHCP passthrough traffic.

Monitoring Transparent Firewall Events

Use either of these optional steps to monitor the activity of the transparent firewall.

SUMMARY STEPS

1. **enable**
2. **debug ip inspect L2-transparent** {**packet** | **dhcp-passthrough**}
3. **show ip inspect** {**name** *inspection-name* | **config** | **interfaces** | **session** [**detail**] | **all**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug ip inspect L2-transparent** {**packet** | **dhcp-passthrough**} **Example:** `Router# debug ip inspect L2-transparent dhcp-passthrough`	Enables debugging messages for transparent firewall events. • **packet**—Displays messages for all debug packets that are inspected by the transparent firewall. • **dhcp-passthrough—**Displays debug messages only for DHCP pass-through traffic that the transparent firewall forwards across the bridge.
Step 3	**show ip inspect** {**name** *inspection-name* | **config** | **interfaces** | **session** [**detail**] | **all**} **Example:** `Router# show ip inspect all`	Displays Cisco IOS Firewall configuration and session information. • If the transparent firewall is configured, use the **all** keyword to display the bridging interface in the interface configuration section of the output.

Examples

The following sample output is a portion of the **show ip inspect all** command that shows the bridging interface:

```
Router# show ip inspect all
.
.
.
Interface Configuration
! Below is the bridging interface.
Interface Ethernet1
Inbound inspection rule is test
tcp alert is on audit-trail is off timeout 3600
ftp alert is on audit-trail is off timeout 3600
Outgoing inspection rule is not set
Inbound access list is not set
Outgoing access list is 156
.
.
.
```

Configuration Examples for Transparent Cisco IOS Firewall

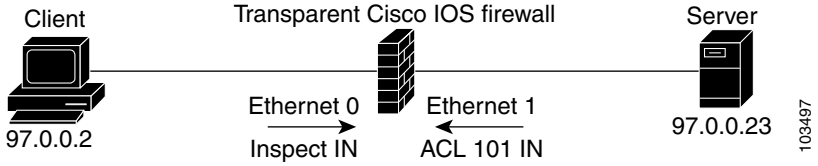
This section contains the following configuration examples:

- Comprehensive Transparent Firewall Configuration: Example, page 10
- Monitoring Telnet Connections via debug and show Output: Examples, page 12
- Configuring and Verifying DHCP Pass-Through Traffic: Examples, page 15

Comprehensive Transparent Firewall Configuration: Example

The following example and sample topology (see Figure 1) illustrate how to configure and debug a transparent Cisco IOS Firewall configuration between a client, a firewall, and a server. This example also includes **show** command output for additional configuration verification. After you have configured a transparent firewall, you can Telnet from the client to the server through the firewall. (See the section "Monitoring Telnet Connections via debug and show Output: Examples."

Figure 33 Sample Topology for Transparent Firewall Configuration




```
! Enable debug commands.
Router# debug ip inspect L2-transparent packet
INSPECT L2 firewall debugging is on
Router# debug ip inspect object-creation
INSPECT Object Creations debugging is on
Router# debug ip inspect object-deletion
INSPECT Object Deletions debugging is on
! Start the transparent firewall configuration process
Router# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
! Configure bridging
Router(config)# bridge 1 protocol ieee
Router(config)# bridge irb
Router(config)# bridge 1 route ip
Router(config)# interface bvi1
*Mar  1 00:06:42.511:%LINK-3-UPDOWN:Interface BVI1, changed state to down.
Router(config-if)# ip address 209.165.200.225 255.255.255.254
! Configure inspection
Router(config)# ip inspect name test tcp
! Following debugs show the memory allocated for CBAC rules.
*Mar  1 00:07:21.127:CBAC OBJ_CREATE:create irc 817F04F0 (test)
*Mar  1 00:07:21.127:CBAC OBJ_CREATE:create irt 818AED20 Protocol:tcp Inactivity time:0
test
Router(config)# ip inspect name test icmp
Router(config)#
*Mar  1 00:07:39.211:CBAC OBJ_CREATE:create irt 818AEDCC Protocol:icmp Inactivity time:0
! Configure Bridging on ethernet0 interface
Router(config)# interface ethernet0
Router(config-if)# bridge-group 1
*Mar  1 00:07:49.071:%LINK-3-UPDOWN:Interface BVI1, changed state to up
*Mar  1 00:07:50.071:%LINEPROTO-5-UPDOWN:Line protocol on Interface BVI1, changed state to
up
! Configure inspection on ethernet0 interface
Router(config-if)# ip inspect test in
Router(config-if)#
*Mar  1 00:07:57.543:CBAC OBJ_CREATE:create idbsb 8189CBFC (Ethernet0)

! Incremented the number of bridging interfaces configured for inspection
*Mar  1 00:07:57.543:L2FW:Incrementing L2FW i/f count
Router(config-if)# interface ethernet1
```

```
! Configure bridging and ACL on interface ethernet1
Router(config-if)# bridge-group 1
Router(config-if)# ip access-group 101 in
*Mar  1 00:08:26.711:%LINEPROTO-5-UPDOWN:Line protocol on Interface Ethernet1, changed
state to up
Router(config-if)# end
Router(config)# end
!
! Issue the show running-config command to verify the complete transparent firewall
! configuration.
Router# show running-config
Building configuration...

Current configuration :1126 bytes
!
version 12.3
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Firewall
!
logging buffered 12000 debugging
no logging console
!
no aaa new-model
ip subnet-zero
no ip domain lookup
!
!
ip inspect name test tcp
ip inspect name test icmp
ip audit notify log
ip audit po max-events 100
no ftp-server write-enable
!
!
!
no crypto isakmp enable
!
!
bridge irb
!
!
interface Ethernet0
 no ip address
 no ip proxy-arp
 ip inspect test in
 bridge-group 1
 hold-queue 100 out
!
interface Ethernet1
 no ip address
 ip access-group 101 in
 no ip unreachables
 no ip proxy-arp
 duplex auto
 bridge-group 1
!
interface BVI1
 ip address 209.165.200.225 255.255.255.254
!
```

```
ip classless
ip route 9.1.0.0 255.255.0.0 9.4.0.1
no ip http server
no ip http secure-server
!
!
ip access-list log-update threshold 1
access-list 101 permit icmp any any log
access-list 101 deny   ip any any log
!
control-plane
!
bridge 1 protocol ieee
bridge 1 route ip
!
line con 0
 no modem enable
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
scheduler max-task-time 5000
!
end
!
! Issue show brige commands to check the tables.
Router# show bridge

Total of 300 station blocks, 300 free
Codes:P - permanent, S - self

Bridge Group 1:

! The bridge table is empty because no traffic has been seen
!
Router# show bridge group

Bridge Group 1 is running the IEEE compatible Spanning Tree protocol

Port 2 (Ethernet0) of bridge group 1 is forwarding
Port 3 (Ethernet1) of bridge group 1 is forwarding
! Note that the interfaces are in a "forwarding" state. The interfaces move from
! a listening state to a learning state and finally to a forwarding state. It takes
! approximately 30 seconds to move to a forwarding after "bridge-group 1" is configured.
```

Monitoring Telnet Connections via debug and show Output: Examples

The following examples shows how to monitor established Telnet connections from the client to the server through the firewall (see Figure 1) and from the server to the client. In these example, the **debug ip inspect L2-transparent packet** command has been issued to generate the debug messages. Relevant **show** commands are also issued for additional verification.

- Telnet Connection from the Client (97.0.0.2) to the Server (97.0.0.23), page 13
- Telnet Connection from the Server (97.0.0.23) to the Client (97.0.0.2), page 15

Telnet Connection from the Client (97.0.0.2) to the Server (97.0.0.23)

The following example is output from the initial Telnet connection between the client and the server. A subsequent connection is established to highlight differences in the debug output. Explanations are given inline.

```
! A packet is received by the firewall in the flood path because the bridge-table is
! initially empty. However, the client seems to have the server's mac-address in its ARP
! cache, so the bridge floods the packet and it appears in the firewall's "flood" path.
*Mar  1 00:17:32.119:L2FW:insp_l2_flood:input is Ethernet0 output is Ethernet1
! Source and destination IP addresses and the L4 protocol of the packet
*Mar  1 00:17:32.123:L2FW*:Src 97.0.0.2 dst 97.0.0.23 protocol tcp
! ACL processing status. An ACL is not configured in this direction; that is, from the
! client to the server.
*Mar  1 00:17:32.123:L2FW:Input ACL not configured or the ACL is bypassed
*Mar  1 00:17:32.123:L2FW:Output ACL is not configured or ACL is bypassed
! If there are exactly two interfaces in the bridge-group and the packet is in flood path,
! the firewall invokes inspection directly, skipping the Unicast flood algorithm. If there
! are more than 2 interfaces, the firewall "drops" the packet and issues the algorithm.
*Mar  1 00:17:32.123:L2FW:FLOOD number of i/fs in bridge-group is exactly 2. Calling
Inspection
! The packet is being inspected.
*Mar  1 00:17:32.123:L2FW:insp_l2_inspection:input is Ethernet0 output is Ethernet1
*Mar  1 00:17:32.123:L2FW*:Src 97.0.0.2 dst 97.0.0.23 protocol tcp
*Mar  1 00:17:32.123:L2FW:Input ACL not configured or the ACL is bypassed
*Mar  1 00:17:32.123:L2FW:Output ACL is not configured or ACL is bypassed

! Memory is allocated for the transparent firewall attributes in the session structure
*Mar  1 00:17:32.123:L2FW:allocating L2 extension for sis
! CBAC-related debug messages: The packet has been passed to the existing CBAC code.
*Mar  1 00:17:32.123:CBAC Pak 814635DC sis 816C9C24 initiator_addr (97.0.0.2:11016)
responder_addr (97.0.0.23:23)
initiator_alt_addr (97.0.0.2:11016) responder_alt_addr (97.0.0.23:23)
! CBAC session structure has been allocated
*Mar  1 00:17:32.127:CBAC OBJ_CREATE:create sis 816C9C24
*Mar  1 00:17:32.127:CBAC OBJ-CREATE:sid 816D69D8 acl 101 Prot:tcp
*Mar  1 00:17:32.127: Src 97.0.0.23 Port [23:23]
*Mar  1 00:17:32.127: Dst 97.0.0.2 Port [11016:11016]
! The Layer 2 header length is being computed for caching the L2 header, which will be
! used if a TCP RST should be sent in the future to tear down the connection.
*Mar  1 00:17:32.127:L2FW:L2 header length(initiator->responder) is 14
! Checks to see if the header is 802.3, SNAP, SAP. (This header is 802.3.)
*Mar  1 00:17:32.127:L2FW:info_start is NULL for init->rsp
*Mar  1 00:17:32.127:CBAC OBJ_CREATE:create host entry 816D4018 addr 97.0.0.23 bucket 118
! CBAC has indicated that the packet should be passed
*Mar  1 00:17:32.127:L2FW:insp_inspection returned FALSE. PASS

! The next packet in the flow has arrived on the interrupt path. This packet is from the
! server (ethernet1) to the client (ethernet0).
*Mar  1 00:17:32.131:L2FW*:insp_l2_fast_inspection:pak 812C9084, input-interface
Ethernet1, output-interface Ethernet0
*Mar  1 00:17:32.131:L2FW*:Src 97.0.0.23 dst 97.0.0.2 protocol tcp
*Mar  1 00:17:32.131:L2FW:Input ACL not configured or the ACL is bypassed
*Mar  1 00:17:32.131:L2FW:Output ACL is not configured or ACL is bypassed
! The Layer 2 header length is computed and will be cached
*Mar  1 00:17:32.131:L2FW:L2 header length is 14 (rsp->init)
*Mar  1 00:17:32.131:L2FW:info_start is NULL rsp->init
! CBAC has indicated that the packet should be forwarded
*Mar  1 00:17:32.131:L2FW*:insp_l2_fast_inspection returning INSP_L2_OK
! A new packet has arrived from the client. The following trace repeats for each packet
received by the firewall
*Mar  1 00:17:32.135:L2FW*:insp_l2_fast_inspection:pak 81462FB4, input-interface
Ethernet0, output-interface Ethernet1
```

```
*Mar  1 00:17:32.135:L2FW*:Src 97.0.0.2 dst 97.0.0.23 protocol tcp
*Mar  1 00:17:32.135:L2FW:Input ACL not configured or the ACL is bypassed
*Mar  1 00:17:32.135:L2FW:Output ACL is not configured or ACL is bypassed
*Mar  1 00:17:32.135:L2FW*:insp_l2_fast_inspection returning INSP_L2_OK
       …<more packets >…
! The host entry for the server is deleted.
*Mar  1 00:17:32.263:CBAC OBJ_DELETE:delete host entry 816D4018 addr 97.0.0.23

! Issue the show ip inspect command to verify that a CBAC session has been established
Router# show ip inspect session detailed

Established Sessions
 Session 816C9C24 (97.0.0.2:11016)=>(97.0.0.23:23) tcp SIS_OPEN
  Created 00:00:28, Last heard 00:00:09
  Bytes sent (initiator:responder) [38:75]
  In  SID 97.0.0.23[23:23]=>97.0.0.2[11016:11016] on ACL 101 (12 matches)
Router#
!
! Issue the show bridge command to verify that entries for the client and server have been
! created in the bridge-table.
Router# show bridge

Total of 300 station blocks, 298 free
Codes:P - permanent, S - self

Bridge Group 1:

    Address       Action   Interface       Age   RX count   TX count
0008.a3b6.b603   forward   Ethernet0         2         14         12
0007.0d97.308f   forward   Ethernet1         2         12         13
Router#
!
! Close the TCP connection (by typing exit at the client).
*Mar  1 00:21:26.259:CBAC OBJ_DELETE:delete sis 816C9C24
*Mar  1 00:21:26.259:CBAC OBJ-DELETE:sid 816D69D8 on acl 101 Prot:tcp
*Mar  1 00:21:26.259: Src 97.0.0.23 Port [23:23]
*Mar  1 00:21:26.259: Dst 97.0.0.2 Port [11016:11016]
! The data structures pertaining to the Layer 2 firewall have been deleted from the
! session. The session has also been deleted.
*Mar  1 00:21:26.259:L2FW:Deleting L2FW data structures
```

A New Telnet Connection from the Client (97.0.0.2) to the Server (97.0.0.23)

```
! The initial SYN packet from the client has arrived in the interrupt path. Note that the
! corresponding packet from the previous telnet session came in on the flood path because
! the bridge-table was empty so the bridge was forced to flood the packet. Since the
! bridge-table is now populated, the packet does not not to be flooded. This is the only
! difference between the previous telnet session and this session. Subsequent packets will
! follow the same path (and generate the same debugs) as the previous session.
*Mar  1 00:23:31.883:L2FW*:insp_l2_fast_inspection:pak 81465190, input-interface
Ethernet0, output-interface Ethernet1
*Mar  1 00:23:31.883:L2FW*:Src 97.0.0.2 dst 97.0.0.23 protocol tcp
*Mar  1 00:23:31.883:L2FW:Input ACL not configured or the ACL is bypassed
*Mar  1 00:23:31.883:L2FW:Output ACL is not configured or ACL is bypassed
! CBAC has indicated that the packet should be punted to the process path since memory
! allocation and the control-plane is involved
*Mar  1 00:23:31.883:L2FW*:insp_l2_fast_inspection returning INSP_L2_PUNT

! After being punted from the interrupt path, the packet has arrived at the process level
! for inspection. Moving forward, the debug messages are similar to the flood case in the
! previous session.
*Mar  1 00:23:31.883:L2FW:insp_l2_inspection:input is Ethernet0 output is Ethernet1
*Mar  1 00:23:31.883:L2FW*:Src 97.0.0.2 dst 97.0.0.23 protocol tcp
*Mar  1 00:23:31.883:L2FW:Input ACL not configured or the ACL is bypassed
```

```
*Mar  1 00:23:31.883:L2FW:Output ACL is not configured or ACL is bypassed
*Mar  1 00:23:31.887:L2FW:allocating L2 extension for sis
*Mar  1 00:23:31.887:CBAC Pak 81465190 sis 816C9C24 initiator_addr (97.0.0.2:11017)
responder_addr (97.0.0.23:23)
initiator_alt_addr (97.0.0.2:11017) responder_alt_addr (97.0.0.23:23)
*Mar  1 00:23:31.887:CBAC OBJ_CREATE:create sis 816C9C24
*Mar  1 00:23:31.887:CBAC OBJ-CREATE:sid 816D69D8 acl 101 Prot:tcp
*Mar  1 00:23:31.887: Src 97.0.0.23 Port [23:23]
*Mar  1 00:23:31.887: Dst 97.0.0.2 Port [11017:11017]
*Mar  1 00:23:31.887:L2FW:L2 header length(initiator->responder) is 14
*Mar  1 00:23:31.887:L2FW:info_start is NULL for init->rsp
*Mar  1 00:23:31.887:CBAC OBJ_CREATE:create host entry 816D4018 addr 97.0.0.23 bucket 118
! CBAC has indicated that the packet should be Passed
*Mar  1 00:23:31.891:L2FW:insp_inspection returned FALSE. PASS
!
! Issue the show ip inspect command to verify the newly created inspect session
Router# show ip inspect session details
Established Sessions
 Session 816C9C24 (97.0.0.2:11017)=>(97.0.0.23:23) tcp SIS_OPEN
  Created 00:00:52, Last heard 00:00:37
  Bytes sent (initiator:responder) [38:75]
  In  SID 97.0.0.23[23:23]=>97.0.0.2[11017:11017] on ACL 101 (10 matches)
Router#
```

Telnet Connection from the Server (97.0.0.23) to the Client (97.0.0.2)

The following sample output is from a Telnet connection that was initiated from the server to the client. This connection will not go through because “ACL 101” is configured to allow only ICMP packets and deny all other packets. Note that inspection is not configured from the server to the client. This example is shown to display the debug messages that are associated with dropped packets.

```
! The first packet from the server comes in on ethernet1 interface
*Mar  1 00:26:12.367:L2FW*:insp_l2_fast_inspection:pak 815C89FC, input-interface
Ethernet1, output-interface Ethernet0
*Mar  1 00:26:12.367:L2FW*:Src 97.0.0.23 dst 97.0.0.2 protocol tcp
! This packet is punted up since ACL 101 is configured for logging. Logging happens in the
process path. If logging was not configured, the packet would have been dropped instead of
being punted to process level
*Mar  1 00:26:12.367:L2FW:Packet punted up by Input ACL for logging
! The packet arrives at process level
*Mar  1 00:26:12.367:L2FW:insp_l2_inspection:input is Ethernet1 output is Ethernet0
*Mar  1 00:26:12.371:L2FW*:Src 97.0.0.23 dst 97.0.0.2 protocol tcp
! The ACL log is generated
*Mar  1 00:26:12.371:%SEC-6-IPACCESSLOGP:list 101 denied tcp 97.0.0.23(11045) ->
97.0.0.2(23), 1 packet
! The packet is dropped by the ACL
*Mar  1 00:26:12.371:L2FW:Packet processed and dropped by Input ACL
! The packet is dropped by the ACL and is therefore NOT sent to CBAC for inspection
*Mar  1 00:26:12.371:L2FW:Packet is dropped in insp_l2_inspection
```

Configuring and Verifying DHCP Pass-Through Traffic: Examples

The following examples show how to verify (via debug messages) DHCP pass-through that has been allowed and traffic that has not been allowed.

- Allowing DHCP Pass-Through Traffic: Example, page 16
- Denying DHCP Pass-Through Traffic: Example, page 16

Allowing DHCP Pass-Through Traffic: Example

In this example, the static IP address of the client is removed and the address is acquired via DHCP using the **ip address dhcp** command on the interface that is connected to the transparent firewall.

```
Router# show debug
ARP:
  ARP packet debugging is on
L2 Inspection:
  INSPECT L2 firewall debugging is on
  INSPECT L2 firewall DHCP debugging is on
Router#
Router#
! Configure DHCP passthrough
Router(config)# ip insp L2-transparent dhcp-passthrough
! The DHCP discover broadcast packet arrives from the client. Since this packet is a
! broadcast (255.255.255.255), it arrives in the flood path
*Mar  1 00:35:01.299:L2FW:insp_l2_flood:input is Ethernet0 output is Ethernet1
*Mar  1 00:35:01.299:L2FW*:Src 0.0.0.0 dst 255.255.255.255 protocol udp
*Mar  1 00:35:01.299:L2FW:udp ports src 68 dst 67
*Mar  1 00:35:01.299:L2FW:src 0.0.0.0 dst 255.255.255.255
! The DHCP pass through flag is checked and the packet is allowed
*Mar  1 00:35:01.299:L2FW:DHCP packet seen. Pass-through flag allows the packet
! The packet is a broadcast packet and therefore not sent to CBAC
*Mar  1 00:35:01.299:L2FW*:Packet is broadcast or multicast.PASS
! The DHCP server 97.0.0.23 responds to the client's request
*Mar  1 00:35:01.303:L2FW:insp_l2_flood:input is Ethernet1 output is Ethernet0
*Mar  1 00:35:01.303:L2FW*:Src 97.0.0.23 dst 255.255.255.255 protocol udp
*Mar  1 00:35:01.307:L2FW:udp ports src 67 dst 68
*Mar  1 00:35:01.307:L2FW:src 97.0.0.23 dst 255.255.255.255
*Mar  1 00:35:01.307:L2FW:DHCP packet seen. Pass-through flag allows the packet
*Mar  1 00:35:01.307:L2FW*:Packet is broadcast or multicast.PASS
*Mar  1 00:35:01.311:L2FW:insp_l2_flood:input is Ethernet0 output is Ethernet1
*Mar  1 00:35:01.311:L2FW*:Src 0.0.0.0 dst 255.255.255.255 protocol udp
*Mar  1 00:35:01.311:L2FW:udp ports src 68 dst 67
*Mar  1 00:35:01.311:L2FW:src 0.0.0.0 dst 255.255.255.255
*Mar  1 00:35:01.315:L2FW:DHCP packet seen. Pass-through flag allows the packet
*Mar  1 00:35:01.315:L2FW*:Packet is broadcast or multicast.PASS
*Mar  1 00:35:01.315:L2FW:insp_l2_flood:input is Ethernet1 output is Ethernet0
*Mar  1 00:35:01.323:L2FW*:Src 97.0.0.23 dst 255.255.255.255 protocol udp
*Mar  1 00:35:01.323:L2FW:udp ports src 67 dst 68
*Mar  1 00:35:01.323:L2FW:src 97.0.0.23 dst 255.255.255.255
*Mar  1 00:35:01.323:L2FW:DHCP packet seen. Pass-through flag allows the packet
*Mar  1 00:35:01.323:L2FW*:Packet is broadcast or multicast.PASS
! The client has an IP address (97.0.0.5) and has issued a G-ARP to let everyone know it's
address
*Mar  1 00:35:01.327:IP ARP:rcvd rep src 97.0.0.5 0008.a3b6.b603, dst 97.0.0.5 BVI1
Router#
```

Denying DHCP Pass-Through Traffic: Example

In this example, DHCP pass-through traffic is not allowed (via the **no ip inspect L2-transparent dhcp-passthrough** command). The client is denied when it attempts to acquire a DHCP address from the server.

```
! Deny DHCP pass-through traffic
Router(config)# no ip inspect L2-transparent dhcp-passthrough
! The DHCP discover broadcast packet arrives from the client
*Mar  1 00:36:40.003:L2FW:insp_l2_flood:input is Ethernet0 output is Ethernet1
*Mar  1 00:36:40.003:L2FW*:Src 0.0.0.0 dst 255.255.255.255 protocol udp
*Mar  1 00:36:40.003:L2FW:udp ports src 68 dst 67
```

```
*Mar  1 00:36:40.007:L2FW:src 0.0.0.0 dst 255.255.255.255
! The pass-through flag is checked
*Mar  1 00:36:40.007:L2FW:DHCP packet seen. Pass-through flag denies the packet
! The packet is dropped because the flag does not allow DHCP passthrough traffic. Thus,
! the client cannot acquire an address, and it times out
*Mar  1 00:36:40.007:L2FW:FLOOD Dropping the packet after ACL check.
Router#
```

Additional References

The following sections provide references related to Transparent Cisco IOS Firewall.

Related Documents

Related Topic	Document Title
Cisco IOS Firewall commands	*Cisco IOS Security Command Reference*, Release 12.3 T
Additional Cisco IOS Firewall configuration information	The section "Traffic Filtering and Firewalls" of the *Cisco IOS Security Configuration Guide*
Bridging commands	*Cisco IOS Bridging and IBM Networking Command Reference, Volume 1 of 2: Bridging*, Release 12.3 T
Additional bridging configuration information	The section "Bridging" of the *Cisco IOS Bridging and IBM Networking Configuration Guide*
DHCP configuration information	The chapter "Configuring DHCP" in the *Cisco IOS IP Configuration Guide*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **ip inspect L2-transparent dhcp-passthrough**
- **debug ip inspect L2-transparent**

Virtual Fragmentation Reassembly

Currently, the Cisco IOS Firewall—specifically context-based access control (CBAC) and the intrusion detection system (IDS)—cannot identify the contents of the IP fragments nor can it gather port information from the fragment. These inabilities allow the fragments to pass through the network without being examined or without dynamic access control list (ACL) creation.

Virtual fragmentation reassembly (VFR) enables the Cisco IOS Firewall to create the appropriate dynamic ACLs, thereby, protecting the network from various fragmentation attacks.

Feature History for Virtual Fragmentation Reassembly

Release	Modification
12.3(8)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Restrictions for Virtual Fragmentation Reassembly, page 682
- Information About Virtual Fragmentation Reassembly, page 682
- How to Use Virtual Fragmentation Reassembly, page 683
- Configuration Examples for Fragmentation Reassembly, page 684
- Additional References, page 686
- Command Reference, page 687

Restrictions for Virtual Fragmentation Reassembly

Performance Impact

VFR will cause a performance impact on the basis of functions such as packet copying, fragment validation, and fragment reorder. This performance impact will vary depending on the number of concurrent IP datagram that are being reassembled.

VFR Configuration Restriction

VFR should not be enabled on a router that is placed on an asymmetric path. The reassembly process requires all of the fragments within an IP datagram. Routers placed in the asymmetric path may not receive all of the fragments, so the fragment reassembly will fail.

SIP and RTSP Limitation

The Session Initiation Protocol (SIP) and the Real-Time Streaming Protocol (RTSP) do not have the ability to parse port information across noncontiguous buffers. Thus, virtual fragmentation reassembly may fail. (If the application fails, the session will be blocked.)

Information About Virtual Fragmentation Reassembly

To use fragmentation support for Cisco IOS Firewall, you should understand the following concept:

- Detected Fragment Attacks, page 682
- Automatically Enabling or Disabling VFR, page 683

Detected Fragment Attacks

VFR is responsible for detecting and preventing the following types of fragment attacks:

- Tiny Fragment Attack—In this type of attack, the attacker makes the fragment size small enough to force Layer 4 (TCP and User Datagram Protocol (UDP)) header fields into the second fragment. Thus, the ACL rules that have been configured for those fields will not match.

  VFR drops all tiny fragments, and an alert message such as follows is logged to the syslog server: “VFR-3-TINY_FRAGMENTS.”

- Overlapping Fragment Attack—In this type of attack, the attacker can overwrite the fragment offset in the noninitial IP fragment packets. When the firewall reassembles the IP fragments, it might create wrong IP packets, causing the memory to overflow or your system to crash.

  VFR drops all fragments within a fragment chain if an overlap fragment is detected, and an alert message such as follows is logged to the syslog server: “VFR-3-OVERLAP_FRAGMENT.”

- Buffer Overflow Attack—In this type of denial-of-service (DoS) attack, the attacker can continuously send a large number of incomplete IP fragments, causing the firewall to lose time and memory while trying to reassemble the fake packets.

  To avoid buffer overflow and control memory usage, configure a maximum threshold for the number of IP datagrams that are being reassembled and the number of fragments per datagram. (Both of these parameters can be specified via the **ip virtual-reassembly** command.)

When the maximum number of datagrams that can be reassembled at any given time is reached, all subsequent fragments are dropped, and an alert message such as the following is logged to the syslog server: "VFR-4_FRAG_TABLE_OVERFLOW."

When the maximum number of fragments per datagram is reached, subsequent fragments will be dropped, and an alert message such as the following is logged to the syslog server: "VFR-4_TOO_MANY_FRAGMENTS."

In addition to configuring the maximum threshold values, each IP datagram is associated with a managed timer. If the IP datagram does not receive all of the fragments within the specified time, the timer will expire and the IP datagram (and all of its fragments) will be dropped.

Automatically Enabling or Disabling VFR

VFR is designed to work with any feature that requires fragment reassembly (such as Cisco IOS Firewall and NAT). Currently, NAT enables and disables VFR internally; that is, when NAT is enabled on an interface, VFR is automatically enabled on that interface.

If more than one feature attempts to automatically enable VFR on an interface, VFR will maintain a reference count to keep track of the number of features that have enabled VFR. When the reference count is reduced to zero, VFR is automatically disabled.

How to Use Virtual Fragmentation Reassembly

This section contains the following procedures:

- Configuring VFR, page 683

Configuring VFR

Use this task to enable VFR on an interface, specify maximum threshold values to combat buffer overflow and control memory usage, and verify any VFR configurations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface type** *type number*
4. **ip virtual-reassembly** [**max-reassemblies** *number*] [**max-fragments** *number*] [**timeout** *seconds*] [**drop-fragments**]
5. **exit**
6. **exit**
7. **show ip virtual-reassembly** [**interface** *type*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet1/1`	Configures an interface type and enters interface configuration mode.
Step 4	**ip virtual-reassembly** [**max-reassemblies** *number*] [**max-fragments** *number*] [**timeout** *seconds*] [**drop-fragments**] **Example:** `Router(config-if)# ip virtual-reassembly max-reassemblies 64 max-fragments 16 timeout 5`	Enables VFR on an interface.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 7	**show ip virtual-reassembly** [**interface** *type*] **Example:** `Router# show ip virtual-reassembly ethernet1/1`	Displays the configuration and statistical information of the VFR. If an interface is not specified, VFR information is shown for all configured interfaces.

Troubleshooting Tips

To view debugging messages related to the VFR subsystem, use the **debug ip virtual-reassembly** command.

Configuration Examples for Fragmentation Reassembly

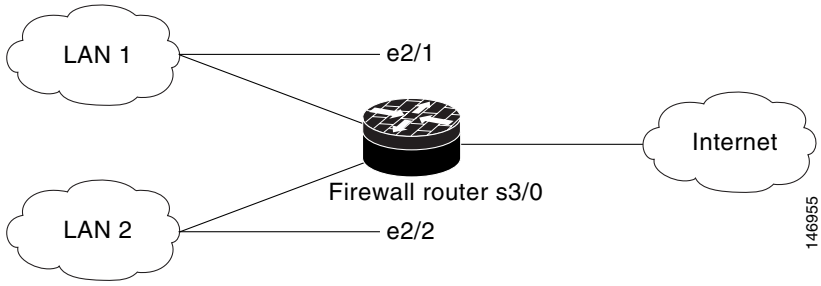
This section contains the following configuration example:

• Configuring VFR and a Cisco IOS Firewall: Example, page 685

Configuring VFR and a Cisco IOS Firewall: Example

The following example shows a typical scenario where the Virtual Fragment Reassembly module is enabled on interfaces ethernet2/1, ethernet2/2, and serial3/0 to facilitate the firewall that is enabled in the outbound direction on interface serial3/0. In this example, the firewall rules that specify the list of LAN1 and LAN2 originating protocols (FTP, HTTP and SMTP) are to be inspected.

Figure 34 VFR and Cisco IOS Firewall Sample Topology




```
!
ip inspect name INTERNET-FW ftp
ip inspect name INTERNET-FW http
ip inspect name INTERNET-FW smtp
!
!
interface Loopback0
 ip address 1.1.1.1 255.255.255.255
!
interface Ethernet2/0
 ip address 9.4.21.9 255.255.0.0
 no ip proxy-arp
 no ip mroute-cache
 duplex half
 no cdp enable
!
interface Ethernet2/1
 description LAN1
 ip address 14.0.0.2 255.255.255.0
 ip virtual-reassembly
 duplex half
!
interface Ethernet2/2
 description LAN2
 ip address 15.0.0.2 255.255.255.0
 ip virtual-reassembly
 duplex half
!
interface Ethernet2/3
 no ip address
 no ip mroute-cache
 shutdown
 duplex half
!
interface Serial3/0
 description Internet
 ip unnumbered Loopback0
 encapsulation ppp
 ip access-group 102 in
 ip inspect INTERNET-FW out
 ip virtual-reassembly
```

```
 serial restart-delay 0
!
ip classless
ip route 0.0.0.0 0.0.0.0 s3/0
!
!
! Access Control Rule that drops all internet originated traffic.
!
access-list 102 deny   ip any any
!
!
!
!
control-plane
!
no call rsvp-sync
!
!
!
dial-peer cor custom
!
!
!
!
gatekeeper
 shutdown
!
!
line con 0
 exec-timeout 0 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 password lab
 login
!
!
end
```

Additional References

The following sections provide references related to virtual fragmentation reassembly.

Related Documents

Related Topic	Document Title
Dynamic IDS	*Cisco IOS Intrusion Prevention System*, Cisco IOS Release 12.3(8)T feature module
CBAC	The chapter "Configuring Context-Based Access Control" in the *Cisco IOS Security Configuration Guide*

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 791	*Internet Protocol*
RFC 1858	*Security Considerations for IP Fragment Filtering*

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **debug ip virtual-reassembly**
- **ip virtual-reassembly**
- **show ip virtual-reassembly**

Glossary

fragment—Part of an IP datagram that is fragmented into multiple pieces. Each piece is called a fragment or an IP fragment.

fragmentation—Process of breaking down an IP datagram into smaller packets (fragments) that are transmitted over different types of network media.

initial fragment— First fragment within a fragment set. This fragment should have a Layer 4 header and should have an offset of zero.

noninitial fragment—All fragments within a fragment set, except the initial fragment.

VRF Aware Cisco IOS Firewall

VRF Aware Cisco IOS Firewall applies Cisco IOS Firewall functionality to VRF (Virtual Routing and Forwarding) interfaces when the firewall is configured on a service provider (SP) or large enterprise edge router. SPs can provide managed services to small and medium business markets.

The VRF Aware Cisco IOS Firewall supports VRF-aware URL filtering and VRF-lite (also known as Multi-VRF CE).

Feature History for VRF Aware Cisco IOS Firewall

Release	Modification
12.3(14)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for VRF Aware Cisco IOS Firewall, page 690
- Restrictions for VRF Aware Cisco IOS Firewall, page 690
- Information About VRF Aware Cisco IOS Firewall, page 690
- How to Configure VRF Aware Cisco IOS Firewall, page 699
- Configuration Examples for VRF Aware Cisco IOS Firewall, page 703
- Additional References, page 712
- Command Reference, page 714
- Glossary, page 715

Prerequisites for VRF Aware Cisco IOS Firewall

- Understand Cisco IOS firewalls.
- Configure VRFs.
- Verify that the VRFs are operational.

Restrictions for VRF Aware Cisco IOS Firewall

- VRF Aware Cisco IOS Firewall is not supported on Multiprotocol Label Switching (MPLS) interfaces.
- If two VPN networks have overlapping addresses, VRF-aware network address translation (NAT) is required for them to support VRF-aware Firewalls.
- When crypto tunnels belonging to multiple VPNs terminate on a single interface, you cannot apply per-VRF firewall policies.

Information About VRF Aware Cisco IOS Firewall

To configure VRF Aware Cisco IOS Firewall, you need to understand the following concepts:

- Cisco IOS Firewall, page 690
- VRF, page 691
- VRF-lite, page 692
- Per-VRF URL Filtering, page 693
- Alerts and Audit Trails, page 693
- MPLS VPN, page 693
- VRF-aware NAT, page 693
- VRF-aware IPSec, page 694
- VRF Aware Cisco IOS Firewall Deployment, page 695

Cisco IOS Firewall

The Cisco IOS Firewall provides robust, integrated firewall and intrusion detection functionality for every perimeter of the network. Available for a wide range of Cisco IOS software-based routers, the Cisco IOS Firewall offers sophisticated security and policy enforcement for connections within an organization (intranet) and between partner networks (extranets), as well as for securing Internet connectivity for remote and branch offices.

The Cisco IOS Firewall enhances existing Cisco IOS security capabilities such as authentication, encryption, and failover, with state-of-the-art security features such as stateful, application-based filtering (context-based access control), defense against network attacks, per-user authentication and authorization, and real-time alerts.

The Cisco IOS Firewall is configurable via Cisco ConfigMaker software, an easy-to-use Microsoft Windows 95, Windows 98, NT 4.0 based software tool.

The Cisco IOS Firewall provides great value in addition to these benefits:

- Flexibility—Provides multiprotocol routing, perimeter security, intrusion detection, VPN functionality, and dynamic per-user authentication and authorization.
- Scalable deployment—Scales to meet any network's bandwidth and performance requirements.
- Investment protection—Leverages existing multiprotocol router investment.
- VPN support—Provides a complete VPN solution based on Cisco IOS IPSec and other CISCO IOS software-based technologies, including L2TP tunneling and quality of service (QoS).

The VRF Aware Cisco IOS Firewall is different from the non-VRF Aware Firewall because it does the following:

- Allows users to configure a per-VRF Firewall. The firewall inspects IP packets that are sent and received within a VRF.
- Allows SPs to deploy the firewall on the provider edge (PE) router.
- Supports overlapping IP address space, thereby allowing traffic from nonintersecting VRFs to have the same IP address.
- Supports per-VRF (not global) firewall command parameters and Denial-of-Service (DoS) parameters so that the VRF-aware Firewall can run as multiple instances (with VRF instances) allocated to various Virtual Private Network (VPN) customers.
- Performs per-VRF URL filtering.
- Generates VRF-specific syslog messages that can be seen only by a particular VPN. These alert and audit-trail messages allow network administrators to manage the firewall; that is, they can adjust firewall parameters, detect malicious sources and attacks, add security policies, and so forth. The vrf name is tagged to syslog messages being logged to the syslog server.

Both VFR Aware and non-VFR Aware Firewalls now allow you to limit the number of firewall sessions. Otherwise, it would be difficult for VRFs to share router resources because one VRF may consume a maximum amount of resources, leaving few resources for other VRFs. That would cause the denial of service to other VRFs. To limit the number of sessions, enter the **ip inspect name** command.

VRF

VPN Routing and Forwarding (VRF) is an IOS route table instance for connecting a set of sites to a VPN service. A VRF contains a template of a VPN Routing/Forwarding table in a PE router.

The overlapping addresses, usually resulting from the use of private IP addresses in customer networks, are one of the major obstacles to successful deployment of peer-to-peer VPN implementation. The MPLS VPN technology provides a solution to this dilemma.

Each VPN has its own routing and forwarding table in the router, so any customer or site that belongs to a VPN is provided access only to the set of routes contained within that table. Any PE router in the MPLS VPN network therefore contains a number of per-VPN routing tables and a global routing table that is used to reach other routers in the service provider network. Effectively, a number of virtual routers are created in a single physical router.

VRF-lite

VRF-lite is a feature that enables a service provider to support two or more VPNs, where IP addresses can be overlapped among the VPNs. VRF-lite uses input interfaces to distinguish routes for different VPNs and forms virtual packet-forwarding tables by associating one or more Layer 3 interfaces with each VRF. Interfaces in a VRF can be physical, such as Ethernet ports, or logical, such as VLAN switched virtual interfaces (SVIs). However, a Layer 3 interface cannot belong to more than one VRF at a time.

Note VRF-lite interfaces must be Layer 3 interfaces.

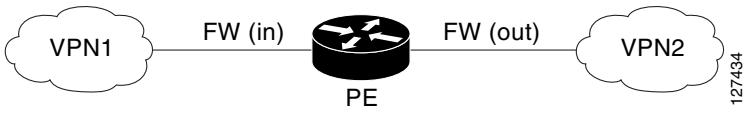
VRF-lite includes these devices:

- Customer edge (CE) devices provide customer access to the service provider network over a data link to one or more provider edge (PE) routers. The CE device advertises the site's local routes to the PE router and learns the remote VPN routes from it. A Catalyst 4500 switch can be a CE.
- Provider edge (PE) routers exchange routing information with CE devices by using static routing or a routing protocol such as BGP, RIPv1, or RIPv2.
- Provider routers (or core routers) are any routers in the service provider network that do not attach to CE devices.
- The PE is only required to maintain VPN routes for those VPNs to which it is directly attached, eliminating the need for the PE to maintain all of the service provider VPN routes. Each PE router maintains a VRF for each of its directly connected sites. Multiple interfaces on a PE router can be associated with a single VRF if all of these sites participate in the same VPN. Each VPN is mapped to a specified VRF. After learning local VPN routes from CEs, a PE router exchanges VPN routing information with other PE routers by using internal BGP (IBPG).

With VRF-lite, multiple customers can share one CE, and only one physical link is used between the CE and the PE. The shared CE maintains separate VRF tables for each customer, and switches or routes packets for each customer based on its own routing table. VRF-lite extends limited PE functionality to a CE device, giving it the ability to maintain separate VRF tables to extend the privacy and security of a VPN to the branch office.

In a VRF-to-VRF situation, if firewall policies are applied on both inbound and outbound interfaces as shown in Figure 35, the firewall on the inbound interface takes precedence over the firewall on the outbound interface. If the incoming packets do not match against the firewall rules (that is, the inspection protocols) configured on the inbound interface, the firewall rule on the outbound interface is applied to the packet.

Figure 35 Firewall in a VRF-to-VRF Scenario



Per-VRF URL Filtering

The VRF Aware firewall supports per-VRF URL filtering. Each VPN can have its own URL filter server. The URL filter server typically is placed in the Shared Service segment of the corresponding VPN. (Each VPN has a VLAN segment in the Shared Service network.) It can also be placed at the customer's site.

Alerts and Audit Trails

CBAC generates real-time alerts and audit trails based on events tracked by the firewall. Enhanced audit trail features use SYSLOG to track all network transactions; recording time stamps, the source host, the destination host, ports used, and the total number of transmitted bytes, for advanced, session-based reporting. Real-time alerts send SYSLOG error messages to central management consoles upon detecting suspicious activity. Using CBAC inspection rules, you can configure alerts and audit trail information on a per-application protocol basis. For example, if you want to generate audit trail information for HTTP traffic, you can specify that in the CBAC rule covering HTTP inspection.

MPLS VPN

When used with MPLS, the VPN feature allows several sites to interconnect transparently through a service provider's network. One service provider network can support several IP VPNs. Each VPN appears to its users as a private network, separate from all other networks. Within a VPN, each site can send IP packets to any other site in the same VPN.

Each VPN is associated with one or more VPN VRF instances. A VRF consists of an IP routing table, a derived Cisco express forwarding (CEF) table, and a set of interfaces that use the forwarding table.

The router maintains a separate routing and CEF table for each VRF. This prevents information from being sent outside the VPN and allows the same subnet to be used in several VPNs without causing duplicate IP address problems.

The router using Multiprotocol BGP (MP-BGP) distributes the VPN routing information using the MP-BGP extended communities.

VRF-aware NAT

Network Address Translation (NAT) allows a single device, such as a router, to act as an agent between the Internet (or public network) and a local (or private) network. Although NAT systems can provide broad levels of security advantages, their main objective is to economize on address space.

NAT allows organizations to resolve the problem of IP address depletion when they have existing networks and need to access the Internet. Sites that do not yet possess NIC-registered IP addresses must acquire them. Cisco IOS NAT eliminates concern and bureaucratic delay by dynamically mapping thousands of hidden internal addresses to a range of easy-to-get addresses.

In general, a NAT system makes it more difficult for an attacker to determine the following:

- Number of systems running on a network
- Type of machines and operating systems they are running
- Network topology and arrangement

NAT integration with MPLS VPNs allows multiple MPLS VPNs to be configured on a single device to work together. NAT can differentiate which MPLS VPN it receives IP traffic from even if the MPLS VPNS are all using the same IP addressing scheme. This enables multiple MPLS VPN customers to share services while ensuring that each MPLS VPN is completely separate from the other.

MPLS service providers would like to provide value-added services such as Internet connectivity, domain name servers (DNS), and VoIP service to their customers. This requires that their customers IP addresses be different when reaching the services. Because MPLS VPN allows customers to use overlapped IP addresses in their networks, NAT must be implemented to make the services possible.

There are two approaches to implementing NAT in the MPLS VPN network. NAT can be implemented on the CE router, which is already supported by NAT, or it can be implemented on a PE router. The NAT Integration with MPLS VPNs feature enables the implementation of NAT on a PE router in an MPLS cloud.

VRF-aware IPSec

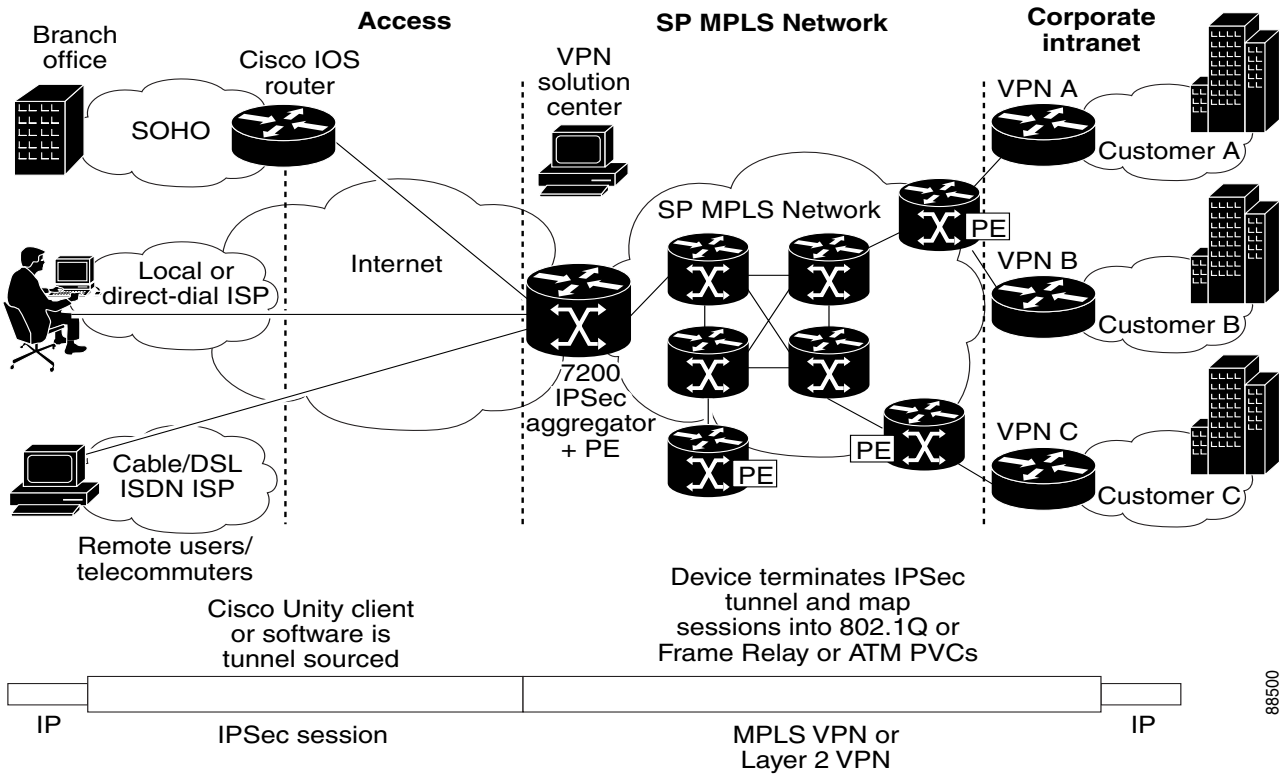
The VRF-aware IPSec feature maps an IP Security (IPSec) tunnel to an MPLS VPN. Using the VRF-aware IPSec feature, you can map IPSec tunnels to VRF instances using a single public-facing address.

Each IPSec tunnel is associated with two VRF domains. The outer encapsulated packet belongs to a VRF domain called the Front Door VRF (FVRF). The inner, protected IP packet belongs to a domain called the Inside VRF (IVRF). In other words, the local endpoint of the IPSec tunnel belongs to the FVRF, whereas the source and destination addresses of the inside packet belong to the IVRF.

One or more IPSec tunnels can terminate on a single interface. The FVRF of all these tunnels is the same and is set to the VRF that is configured on that interface. The IVRF of these tunnels can be different and depends on the VRF that is defined in the Internet Security Association and Key Management Protocol (ISAKMP) profile that is attached to a crypto map entry.

Figure 36 illustrates a scenario showing IPSec to MPLS and Layer 2 VPNs.

Figure 36 IPSec-to-MPLS and Layer 2 VPNs



VRF Aware Cisco IOS Firewall Deployment

A firewall can be deployed at many points within the network to protect VPN sites from Shared Service (or the Internet) and vice versa. The following firewall deployments are described:

- Distributed Network Inclusion of VRF Aware Cisco IOS Firewall, page 695
- Hub-and-Spoke Network Inclusion of VRF Aware Cisco IOS Firewall, page 697

Distributed Network Inclusion of VRF Aware Cisco IOS Firewall

A VRF Aware Cisco IOS Firewall in a distributed network has the following advantages:

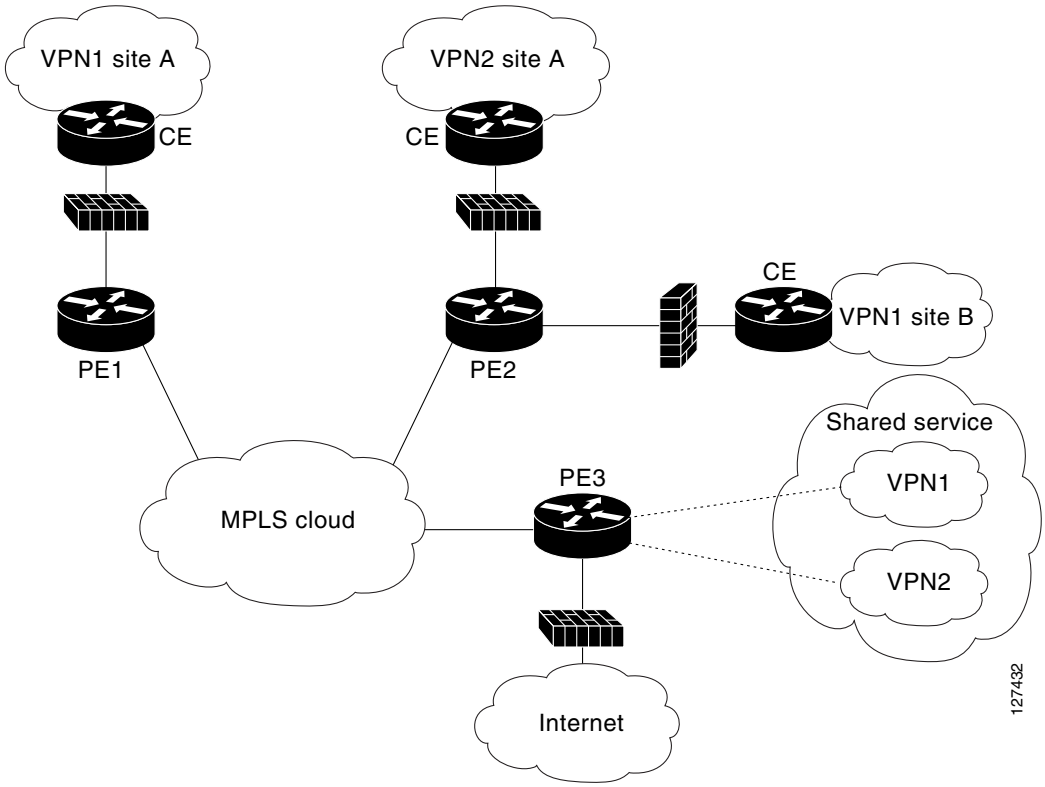
- The firewall is distributed across the MPLS core, so the firewall processing load is distributed to all ingress PE routers.
- VPN Firewall features can be deployed in the inbound direction.
- Shared Service is protected from the VPN site at the ingress PE router; therefore, malicious packets from VPN sites are filtered at the ingress PE router before they enter the MPLS core.

However, the following disadvantages exist:

- There is no centralized firewall deployment, which complicates the deployment and management of the firewall.
- Shared Service firewall features cannot be deployed in the inbound direction.
- The MPLS core is open to the Shared Service. Therefore, malicious packets from Shared Service are filtered only at the ingress PE router after traveling through all core routers.

Figure 37 illustrates a typical situation in which an SP offers firewall services to VPN customers VPN1 and VPN2, thereby protecting VPN sites from the external network (for example, Shared Services and the Internet) and vice versa.

Figure 37 Distributed Network



In this example, VPN1 has two sites, Site A and Site B, that span across the MPLS core. Site A is connected to PE1, and Site B is connected to PE2. VPN2 has only one site that is connected to PE2.

Each VPN (VPN1 and VPN2) has the following:

- A VLAN segment in the Shared Service that is connected to the corresponding VLAN subinterface on PE3.
- Internet access through the PE3 router that is connected to the Internet

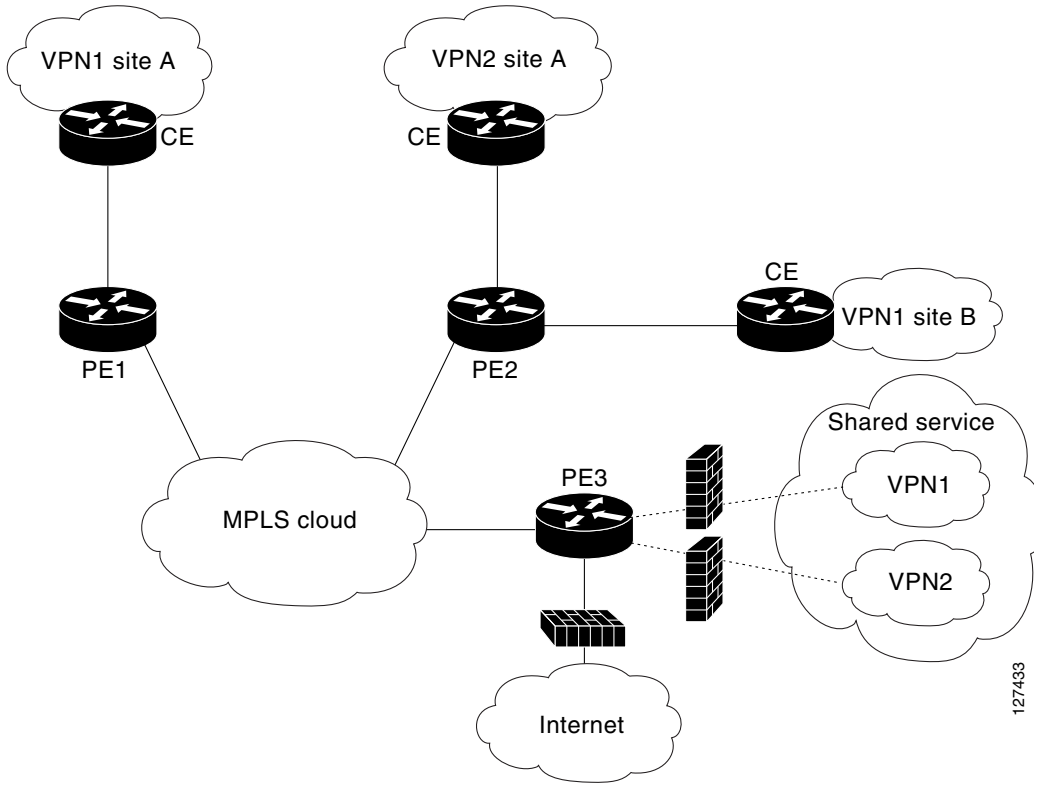
A distributed network requires the following firewall policies:

- VPN Firewall (VPN1-FW and VPN2-FW)—Inspects VPN-generated traffic that is destined to Shared Service or the Internet and blocks all non-firewall traffic that is coming from outside (Shared Service or the Internet), thereby protecting the VPN sites from outside traffic. This firewall typically is deployed on the VRF interface of the ingress PE router that is connected to the VPN site being protected. It is deployed in the inbound direction because the VRF interface is inbound to the VPN site being protected.
- Shared Service Firewall (SS-FW)—Inspects Shared Service-originated traffic that is destined to VPN sites and blocks all non-firewall traffic that is coming from outside (the VPN site), thereby protecting the Shared Service network from VPN sites. This firewall typically is deployed on the VRF interface of the ingress PE router that is connected to the VPN site from where the Shared Service is being protected. It is deployed in the outbound direction because the VRF interface is outbound to the Shared Service that is being protected.
- Generic-VPN Firewall (GEN-VPN-FW)—Inspects VPN-generated traffic that is destined to the Internet and blocks all non-firewall traffic that is coming from the Internet, thereby protecting all VPNs from the Internet. This firewall typically is deployed on the Internet-facing interface of the PE router that is connected to the Internet. It is deployed in the outbound direction because the Internet-facing interface is outbound to VPNs being protected.
- Internet Firewall (INET-FW)—Inspects Internet-generated traffic that is destined to Shared Service and blocks all non-firewall traffic that is coming from VPNs or Shared Service, thereby protecting the Internet from VPNs. This firewall typically is deployed on the Internet-facing interface of the PE router that is connected to the Internet. It is deployed in the inbound direction because the Internet-facing interface is inbound to the Internet being protected.

Hub-and-Spoke Network Inclusion of VRF Aware Cisco IOS Firewall

Figure 38 illustrates a hub-and-spoke network where the firewalls for all VPN sites are applied on the egress PE router PE3 that is connected to the Shared Service.

Figure 38 Hub-and-Spoke Network



Typically each VPN has a VLAN and/or VRF subinterface connected to the Shared Service. When a packet arrives from an MPLS interface, the inner tag represents the VPN-ID. MPLS routes the packet to the corresponding subinterface that is connected to Shared Service.

A Hub-and-Spoke network requires the following firewall policies:

- VPN Firewall (VPN1-FW and VPN2-FW)—Inspects VPN-generated traffic that is destined to Shared Service and blocks all non-firewall traffic that is coming from Shared Service, thereby protecting the VPN sites from Shared Service traffic. This firewall typically is deployed on the VLAN subinterface of the egress PE router that is connected to the Shared Service network. It is deployed in the outbound direction because the VLAN interface is outbound to the VPN site being protected.
- Shared Service Firewall (SS-FW)—Inspects Shared Service originated traffic that is destined to the VPN/Internet and blocks all non-firewall traffics that is coming from outside, thereby protecting the Shared Service network from VPN/Internet traffic. This firewall typically is deployed on the VLAN interface of the egress PE router that is connected to the Shared Service being protected. It is deployed in the inbound direction because the VLAN interface is inbound to the Shared Service being protected.

- Generic-VPN firewall (GEN-VPN-FW)—Inspects VPN-generated traffic that is destined to the Internet and blocks all non-firewall traffic that is coming from the Internet, thereby protecting all VPNs from the Internet. This firewall typically is deployed on the Internet-facing interface of the PE router that is connected to the Internet. It is deployed in the outbound direction because the Internet-facing interface is outbound to the VPNs being protected.
- Internet firewall (INET-FW)—Inspects Internet-generated traffic that is destined to Shared Service and blocks all non-firewall traffic that is coming from VPNs or Shared Service, thereby protecting the Internet from VPNs. This firewall typically is deployed on the Internet-facing interface of the PE router that is connected to the Internet. It is deployed in the inbound direction because the Internet-facing interface is inbound to the Internet being protected.

How to Configure VRF Aware Cisco IOS Firewall

This section contains the following procedures:

- Configuring and Checking ACLs to Ensure that Only Inspected Traffic Can Pass Through the Firewall and that Non-Firewall Traffic is Blocked, page 699 (required)
- Creating and Naming Firewall Rules and Applying the Rules to the Interface, page 700 (required)
- Identifying and Setting Firewall Attributes, page 701 (optional)

Configuring and Checking ACLs to Ensure that Only Inspected Traffic Can Pass Through the Firewall and that Non-Firewall Traffic is Blocked

To configure ACLs and verify that only inspected traffic can pass through the firewall, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *access-list-name*
4. **interface** *interface-type*
5. **ip access-group** {*access-list-number* | *access-list-name*} {**in** | **out**}
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list extended** *access-list-name* **Example:** `Router(config)# ip access-list extended vpn-ac1`	Defines an extended IP ACL to block non-firewall traffic in both inbound and outbound directions.
Step 4	**interface** *interface-type* **Example:** `Router(config)# interface ethernet0/1.10`	Enters interface configuration mode and specifies an interface that is associated with a VRF.
Step 5	**ip access-group** {*access-list-number* | *access-list-name*} {**in** | **out**} **Example:** `Router(config-if)# ip access-group vpn-acl in`	Controls access to an interface. Applies the previously defined IP access list to a VRF interface whose non-firewall traffic is blocked.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode. Returns to global configuration mode.

Creating and Naming Firewall Rules and Applying the Rules to the Interface

To create and name firewall rules and apply the rules to the interface, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip inspect name** *inspection-name* [**parameter max-sessions** *number*] *protocol* [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout** *seconds*]
4. **interface** *interface-id*
5. **ip inspect** *rule-name* {**in** | **out**}
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip inspect name** *inspection-name* [**parameter max-sessions** *number*] *protocol* [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout** *seconds*] **Example:** `Router(config)# ip inspect name vpn_fw ftp`	Defines a set of inspection rules.
Step 4	**interface** *interface-id* **Example:** `Router(config)# interface ethernet0/1.10`	Enters interface configuration mode and specifies an interface that is associated with a VRF.
Step 5	**ip inspect** *rule-name* {**in** | **out**} **Example:** `Router(config-if)# ip inspect vpn_fw in`	Applies the previously defined inspection role to a VRF interface whose traffic needs to be inspected.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

Identifying and Setting Firewall Attributes

To identify and set firewall attributes, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip inspect tcp max-incomplete host** *number* **block-time** *minutes* [**vrf** *vrf-name*]
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip inspect tcp max-incomplete host** *number* **block-time** *minutes* [**vrf** *vrf-name*] **Example:** `Router(config)# ip inspect tcp max-incomplete host 256 vrf bank-vrf`	Specifies threshold and blocking time values for TCP host-specific denial-of-service detection and prevention.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Verifying the VRF Aware Cisco IOS Firewall Configuration and Functioning

Verify the configuration and functioning of the firewall by entering the commands shown below. For detailed descriptions of these commands and other verification commands, see the "Command Reference" section on page 714.

SUMMARY STEPS

1. **show ip inspect** {**name** *inspection-name* | **config** | **interfaces** | **session** [**detail**] | **statistics** | **all**}[**vrf** *vrf-name*]
2. **show ip urlfilter** {**config** | **cache** | **statistics**} [**vrf** *vrf-name*]

DETAILED STEPS

Step 1 **show ip inspect** {**name** *inspection-name* | **config** | **interfaces** | **session** [**detail**] | **statistics** | **all**}[**vrf** *vrf-name*]

Use this command to view the firewall configurations, sessions, statistics, and so forth, pertaining to a specified VRF. For example, to view the firewall sessions pertaining to the VRF bank, enter the following command:

```
Router# show ip inspect interfaces vrf bank
```

Step 2 **show ip urlfilter** {**config** | **cache** | **statistics**} [**vrf** *vrf-name*]

Use this command to view the configurations, cache entries, statistics, and so forth, pertaining to a specified VRF. For example, to view the URL filtering statistics pertaining to the VRF bank, enter the following command:

```
Router# show ip urlfilter statistics vrf bank
```

Configuration Examples for VRF Aware Cisco IOS Firewall

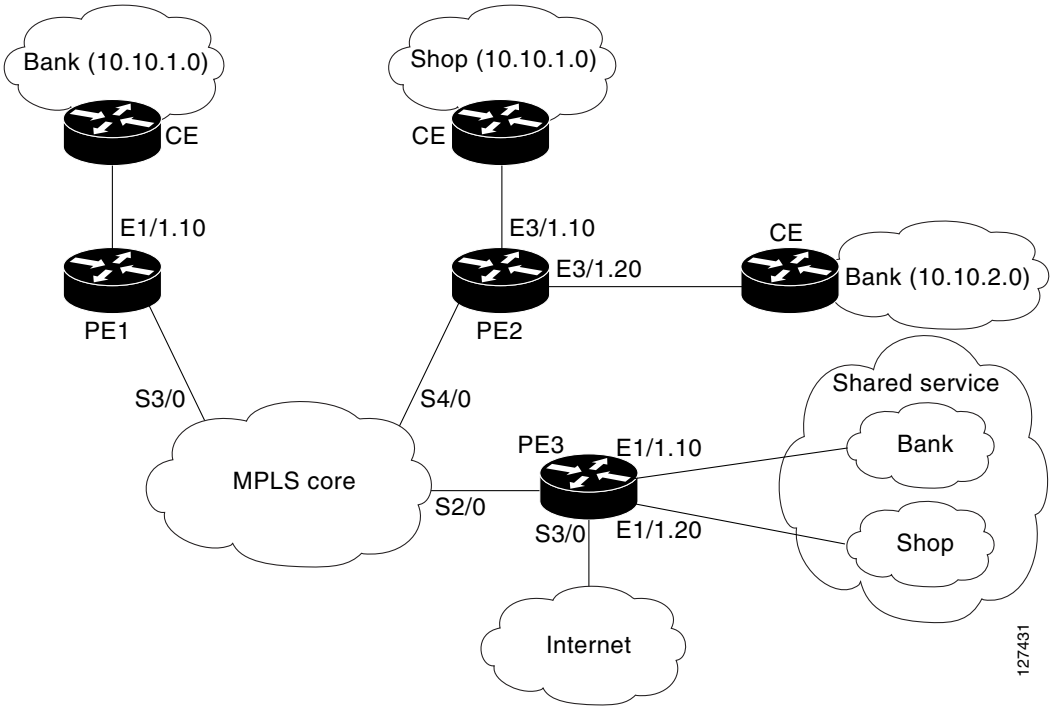
In the example illustrated in Figure 39, a service provider offers firewall service to VPN customers **Bank** and **Shop**. The Bank VPN has the following two sites in an MPLS network:

- Site connected to PE1, whose network address is 10.10.1.0/24
- Site connected to PE2, whose network address is 10.10.2.0/24

The Bank VPN also has a VLAN network segment in Shared Service that is connected to PE3.

The Shop VPN has only one site, which is connected to PE4. The network address 10.10.1.0/24 is the same network address to which the Bank VPN site is connected.

Figure 39 VPN with Two Sites Across MPLS Network



Each VPN needs the following two firewalls:

- VPN firewall to protect the VPN site from Shared Services
- Shared Service (SS) firewall to protect SS from the VPN site

In addition, the following two firewalls are required:

- Internet firewall to protect VPNs from the Internet
- Generic VPN firewall to protect the Internet from VPNs

In this example, the security policies for Bank and Shop VPNs are as follows:

- Bank VPN Firewall—bank_vpn_fw (Inspects FTP, HTTP, and ESMTP protocols)
- Bank SS Firewall—bank_ss_fw (Inspects ESMTP protocol)
- Shop VPN Firewall—shop_vpn_fw (Inspects HTTP and RTSP protocols)
- Shop SS Firewall—shop_ss_fw (Inspects H323 protocol)

The security policies for the Internet firewall and generic VPN firewall are as follows:

- Internet firewall—inet_fw (Inspects HTTP and ESMTP protocols)
- Generic VPN firewall—gen_vpn_fw (Inspects FTP, HTTP, ESMTP, and RTSP protocols)

DISTRIBUTED NETWORK

PE1:

```
! VRF instance for the Bank VPN
ip vrf bank
rd 100:10
route-target export 100:10
route-target import 100:10
!
! VPN Firewall for Bank VPN protects Bank VPN from Shared Service
ip inspect name bank_vpn_fw ftp
ip inspect name bank_vpn_fw http
ip inspect name bank_vpn_fw esmtp
!
! Shared Service firewall for Bank VPN protects Shared Service from Bank VPN
ip inspect name bank_ss_fw esmtp
!
! VRF interface for the Bank VPN
interface ethernet0/1.10
!
! description of VPN site Bank to PE1
encapsulation dot1Q 10
ip vrf forwarding bank
ip address 10.10.1.2 255.255.255.0
ip access-group bank_ss_acl in
ip access-group bank_vpn_acl out
ip inspect bank_vpn_fw in
ip inspect bank_ss_fw out
!
! MPLS interface
interface Serial3/0
 ip unnumbered Loopback0
 tag-switching ip
 serial restart-delay 0
!
! ACL that protects the VPN site Bank from Shared Service
ip access-list extended bank_vpn_acl
 permit ip 10.10.2.0 0.0.0.255 10.10.1.0 0.0.0.255
 permit tcp any any eq smtp
 deny   ip  any any log
!
! ACL that protects Shared Service from VPN site Bank
ip access-list extended bank_ss_acl
 permit ip 10.10.1.0 0.0.0.255 10.10.2.0 0.0.0.255
 permit tcp any any eq ftp
 permit tcp any any eq http
 permit tcp any any eq smtp
 deny   ip  any any log
```

PE2:

```
! VRF instance for the Bank VPN
ip vrf bank
rd 100:10
route-target export 100:10
route-target import 100:10
!
! VRF instance for the Shop VPN
ip vrf shop
rd 200:20
route-target export 200:20
route-target import 200:20
!
```

```
! VPN firewall for Bank BPN protects Bank VPN from Shared Service
ip inspect name bank_vpn_fw ftp
ip inspect name bank_vpn_fw http
ip inspect name bank_vpn_fw esmtp
!
! Shared Service firewall for Bank VPN protects Shared Service from Bank VPN
ip inspect name bank_ss_fw esmtp
!
! VPN firewall for Shop VPN protects Shop VPN from Shared Service
ip inspect name shop_vpn_fw http
ip inspect name shop_vpn_fw rtsp
!
! Shared Service firewall for Shop VPN protects Shared Service from Shop VPN
ip inspect name shop_ss_fw h323
!
! VRF interface for the Bank VPN
interface Ethernet3/1.10
!
! description of VPN site Bank to PE2
encapsulation dot1Q 10
ip vrf forwarding bank
ip address 10.10.2.2 255.255.255.0
ip access-group bank_ss_acl in
ip access-group bank_vpn_acl out
ip inspect bank_vpn_fw in
ip inspect bank_ss_fw out
!
interface Ethernet3/1.20
!
! description of VPN site Shop to PE2
encapsulation dot1Q 20
ip vrf forwarding shop
ip address 10.10.1.2 255.255.255.0
ip access-group shop_ss_acl in
ip access-group shop_vpn_acl out
ip inspect shop_vpn_fw in
ip inspect shop_ss_fw out
!
interface Serial4/0
 ip unnumbered Loopback0
 tag-switching ip
 serial restart-delay 0
!
! ACL that protects the VPN site Bank from Shared Service
ip access-list extended bank_vpn_acl
 permit ip 10.10.1.0 0.0.0.255 10.10.2.0 0.0.0.255
 permit tcp any any eq smtp
 deny   ip  any any log
!
! ACL that protects Shared Service from VPN site Bank
ip access-list extended bank_ss_acl
 permit ip 10.10.2.0 0.0.0.255 10.10.1.0 0.0.0.255
 permit tcp any any eq ftp
 permit tcp any any eq http
 permit tcp any any eq smtp
 deny   ip  any any log
!
```

```
! ACL that protects VPN site Shop from Shared Service
ip access-list extended shop_vpn_acl
 permit tcp any any eq h323
 deny   ip any any log
!
ip access-list extended shop_ss_acl
 permit tcp any any eq http
 permit tcp any any eq rtsp
deny   ip  any any log
```

PE3:

```
! VRF instance for the Bank VPN
ip vrf bank
rd 100:10
route-target export 100:10
route-target import 100:10
!
! VRF instance for the Shop VPN
ip vrf shop
rd 200:20
route-target export 200:20
route-target import 200:20
!
! Generic VPN firewall to protect Shop and Bank VPNs from internet
ip inspect name gen_vpn_fw esmtp
ip inspect name gen_vpn_fw ftp
ip inspect name gen_vpn_fw http
ip inspect name gen_vpn_fw rtsp
!
! Internet firewall to prevent malicious traffic from being passed
! to internet from Bank and Shop VPNs
ip inspect name inet_fw esmtp
ip inspect name inet_fw http
!
! VRF interface for the Bank VPN
interface Ethernet1/1.10
!
! Description of Shared Service to PE3
encapsulation dot1Q 10
ip vrf forwarding bank
ip address 10.10.1.50 255.255.255.0
!
! VRF interface for the Shop VPN
interface Ethernet1/1.20
!
! Description of Shared Service to PE3
encapsulation dot1Q 20
ip vrf forwarding shop
ip address 10.10.1.50 255.255.255.0
!
interface Serial2/0
 ip unnumbered Loopback0
 tag-switching ip
 serial restart-delay 0
!
! VRF interface for the Bank VPN
interface Serial3/0
!
! Description of Internet-facing interface
ip address 192.168.10.2 255.255.255.0
ip access-group inet_acl out
ip access-group gen_vpn_acl in
ip inspect gen_vpn_fw out
```

```
ip inspect inet_fw in
!
! ACL that protects the Bank and Shop VPNs from internet
ip access-list extended gen_vpn_acl
 permit tcp any any eq smtp
 permit tcp any any eq www
 deny   ip  any any log
!
! ACL that protects internet from Bank and Shop VPNs
ip access-list extended inet_acl
 permit tcp any any eq ftp
 permit tcp any any eq http
 permit tcp any any eq smtp
 permit tcp any any eq rtsp
 deny   ip  any any log
```

HUB-AND-SPOKE NETWORK

PE3:

```
! VRF instance for the VPN Bank
ip vrf bank
rd 100:10
route-target export 100:10
route-target import 100:10
!
! VRF instance for the VPN Shop
ip vrf shop
rd 200:20
route-target export 200:20
route-target import 200:20
!
! VPN firewall for Bank BPN protects Bank VPN from Shared Service
ip inspect name bank_vpn_fw ftp
ip inspect name bank_vpn_fw http
ip inspect name bank_vpn_fw esmtp
!
! Shared Service firewall for Bank VPN protects Shared Service from Bank VPN
ip inspect name bank_ss_fw esmtp
!
! VPN firewall for Shop VPN protects Shop VPN from Shared Service
ip inspect name shop_vpn_fw http
ip inspect name shop_vpn_fw rtsp
!
! Shared Service firewall for Shop VPN protects Shared Service from Shop VPN
ip inspect name shop_ss_fw h323
!
! Generic VPN firewall protects Shop and Bank VPNs from internet
ip inspect name gen_vpn_fw esmtp
ip inspect name gen_vpn_fw ftp
ip inspect name gen_vpn_fw http
ip inspect name gen_vpn_fw rtsp
!
! Internet firewall prevents malicious traffic from being passed
! to internet from Bank and Shop VPNs
ip inspect name inet_fw esmtp
ip inspect name inet_fw http
!
! VRF interface for the Bank VPN
interface Ethernet1/1.10
!
! description of Shared Service to PE3
encapsulation dot1Q 10
```

```
ip vrf forwarding bank
ip address 10.10.1.50 255.255.255.0
ip access-group bank_ss_acl out
ip access-group bank_vpn_acl in
ip inspect bank_vpn_fw out
ip inspect bank_ss_fw in
!
! VRF interface for the Shop VPN
interface Ethernet1/1.20
!
! description of Shared Service to PE3
encapsulation dot1Q 20
ip vrf forwarding shop
ip address 10.10.1.50 255.255.255.0
ip access-group shop_ss_acl out
ip access-group shop_vpn_acl in
ip inspect shop_vpn_fw out
ip inspect shop_ss_fw in
!
interface Serial2/0
 ip unnumbered Loopback0
 tag-switching ip
 serial restart-delay 0
!
! VRF interface for the Bank VPN
interface Serial3/0
!
! description of Internet-facing interface
ip address 192.168.10.2 255.255.255.0
ip access-group inet_acl out
ip access-group gen_vpn_acl in
ip inspect gen_vpn_fw out
ip inspect inet_fw in
!
! ACL that protects the VPN site Bank from Shared Service
ip access-list extended bank_vpn_acl
 permit tcp any any eq smtp
 deny   ip  any any log
!
! ACL that protects Shared Service from VPN site Bank
ip access-list extended bank_ss_acl
 permit tcp any any eq ftp
 permit tcp any any eq http
 permit tcp any any eq smtp
 deny   ip  any any log
!
! ACL that protects VPN site Shop from Shared Service
ip access-list extended shop_vpn_acl
 permit tcp any any eq h323
 deny   ip any any log
!
ip access-list extended shop_ss_acl
 permit tcp any any eq http
 permit tcp any any eq rtsp
 deny   ip  any any log
!
! ACL that protects the Bank and Shop VPNs from internet
ip access-list extended gen_vpn_acl
 permit tcp any any eq smtp
 permit tcp any any eq www
 deny   ip  any any log
!
! ACL that protects internet from Bank and Shop VPNs
ip access-list extended inet_acl
```

```
permit tcp any any eq ftp
permit tcp any any eq http
permit tcp any any eq smtp
permit tcp any any eq rtsp
deny   ip  any any log
```

In the example illustrated in Figure 40, the Cisco IOS Firewall is configured on PE1 on the VRF interface E3/1. The host on NET1 wants to reach the server on NET2.

Figure 40 Sample VRF Aware Cisco IOS Firewall Network

The configuration steps are followed by a sample configuration and log messages.

1. Configure VRF on PE routers.
2. Ensure that your network supports MPLS traffic engineering.
3. Confirm that the VRF interface can reach NET1 and NET2.
4. Configure the VRF Aware Cisco IOS Firewall.
   a. Configure and apply ACLs.
   b. Create Firewall rules and apply them to the VRF interface.
5. Check for VRF firewall sessions.

VRF Configuration on PE1

```
! configure VRF for host1
ip cef
ip vrf vrf1
rd 100:1
route-target export 100:1
route-target import 100:1
exit
end
!
! apply VRF to the interface facing CE
interface ethernet3/1
ip vrf forwarding vrf1
ip address 190.1.1.2 255.255.0.0
!
! make the interface facing the MPLS network an MPLS interface
interface serial2/0
mpls ip
ip address 191.171.151.1 255.255.0.0
!
! configure BGP protocol for MPLS network
router bgp 100
no synchronization
bgp log-neighbor-changes
neighbor 191.171.151.2 remote-as 100
neighbor 191.171.151.2 update-source serial2/0
no auto-summary
address-family vpnv4
neighbor 191.171.151.2 activate
neighbor 191.171.151.2 send-community both
exit-address-family
```

```
address-family ipv4 vrf vrf1
redistribute connected
redistribute static
no auto-summary
no synchronization
exit-address-family
!
! configure VRF static route to reach CE network
ip route vrf vrf1 192.168.4.0 255.255.255.0 190.1.1.1
```

VRF Configuration on PE2

```
! configure VRF for host2
ip cef
ip vrf vrf1
rd 100:1
route-target export 100:1
route-target import 100:1
!
! apply VRF on CE-facing interface
interface fastethernet0/0
ip vrf forwarding vrf1
ip address 193.1.1.2 255.255.255.0
!
! make MPLS network-facing interface an MPLS interface
interface serial1/0
mpls ip
ip address 191.171.151.2 255.255.0.0
!
! configure BGP protocol for MPLS network
router bgp 100
no synchronization
bgp log-neighbor-changes
neighbor 191.171.151.1 remote-as 100
neighbor 191.171.151.1 update-source serial1/0
no auto-summary
address-family vpnv4
neighbor 191.171.151.1 activate
neighbor 191.171.151.1 send-community both
exit-address-family
address-family ipv4 vrf vrf1
redistribute connected
redistribute static
no auto-summary
no synchronization
exit-address-family

!configure VRF static route to reach CE network
ip route vrf vrf1 192.168.4.0 255.255.255.0 193.1.1.1
```

Configuration on CE1

```
interface e0/1
ip address 190.1.1.1 255.255.255.0

interface e0/0
ip address 192.168.4.2 255.255.255.0

ip route 192.168.104.0 255.255.255.0 190.1.1.2
```

Configuration on CE2

```
interface e0/1
ip address 190.1.1.1 255.255.255.0

interface e0/0
ip address 192.168.4.2 255.255.255.0

ip route 192.168.4.0 255.255.255.0 193.1.1.2
```

Configure Firewall on PE1 and Apply on the VRF Interface

```
! configure ACL so that NET2 cannot access NET1
ip access-list extended 105
permit tcp any any fragment
permit udp any any fragment
deny tcp any any
deny udp any any
permit ip any any
!
! apply ACL to VRF interface on PE1
interface ethernet3/1
ip access-group 105 out
!
! configure firewall rule
ip inspect name test tcp
!
! apply firewall rule on VRF interface
interface ethernet3/1
 ip inspect test in
```

Check for VRF Firewall Sessions When Host on NET1 Tries to Telnet to Server on NET2

```
show ip inspect session vrf vrf1
Established Sessions
 Session 659CE534 (192.168.4.1:38772)=>(192.168.104.1:23) tcp SIS_OPEN
!
! checking for ACLs
show ip inspect session detail vrf vrf1 | include ACL 105
  Out SID 192.168.104.1[23:23]=>192.168.4.1[38772:38772] on ACL 105
(34 matches)
```

Additional References

The following sections provide references related to VRF Aware Cisco IOS Firewall.

Related Documents

Related Topic	Document Title
VRF-lite	• *Catalyst 4500 Series Switch Cisco IOS Software Configuration Guide*, Release 12.2
MPLS VPN	• *Configuring a Basic MPLS VPN*, Document ID 13733
VRF Aware IPSec	• *VRF-Aware IPSec* feature module, Release 12.2(15)T • *Cisco IOS Security Configuration Guide*, Release 12.3 • *Cisco IOS Security Command Reference*, Release 12.3T

Related Topic	Document Title
VRF management	• *Cisco 12000/10720 Router Manager User's Guide*, Release 3.2
NAT	• *NAT and Stateful Inspection of Cisco IOS Firewall*, White Paper • *Configuring Network Address Translation: Getting Started*—Document ID 13772

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **clear ip urlfilter cache**
- **ip inspect alert-off**
- **ip inspect audit trail**
- **ip inspect dns-timeout**
- **ip inspect max-incomplete high**
- **ip inspect max-incomplete low**
- **ip inspect name**
- **ip inspect one-minute high**
- **ip inspect one-minute low**
- **ip inspect tcp finwait-time**
- **ip inspect tcp idle-time**
- **ip inspect tcp max-incomplete host**
- **ip inspect tcp synwait-time**
- **ip inspect udp idle-time**
- **ip urlfilter alert**
- **ip urlfilter allowmode**
- **ip urlfilter audit-trail**
- **ip urlfilter cache**
- **ip urlfilter exclusive-domain**
- **ip urlfilter exclusive-domain**
- **ip urlfilter max-request**
- **ip urlfilter max-resp-pak**
- **ip urlfilter server vendor**
- **ip urlfilter urlf-server-log**
- **show ip inspect**
- **show ip urlfilter cache**
- **show ip urlfilter config**
- **show ip urlfilter statistics**

Glossary

CE router—customer edge router. A router that is part of a customer network and that interfaces to a provider edge (PE) router.

CBAC—Context-Based Access Control. A protocol that provides internal users with secure access control for each application and for all traffic across network perimeters. CBAC enhances security by scrutinizing both source and destination addresses and by tracking each application's connection status.

data authentication—Refers to one or both of the following: data integrity, which verifies that data has not been altered, or data origin authentication, which verifies that the data was actually sent by the claimed sender.

data confidentiality—A security service where the protected data cannot be observed.

edge router—A router that turns unlabeled packets into labeled packets, and vice versa.

firewall—A router or access server, or several routers or access servers, designated as a buffer between any connected public networks and a private network. A firewall router uses access lists and other methods to ensure the security of the private network.

inspection rule—A rule that specifies what IP traffic (which application-layer protocols) will be inspected by CBAC at an interface.

intrusion detection—The Cisco IOS Firewall's Intrusion Detection System (Cisco IOS IDS) identifies the most common attacks, using signatures to detect patterns of misuse in network traffic.

IPSec—IP Security Protocol. A framework of open standards developed by the Internet Engineering Task Force (IETF). IPSec provides security for transmission of sensitive data over unprotected networks such as the Internet.

managed security services—A comprehensive set of programs that enhance service providers' abilities to meet the growing demands of their enterprise customers. Services based on Cisco solutions include managed firewall, managed VPN (network based and premises based), and managed intrusion detection.

NAT—Network Address Translation. Translates a private IP address used inside the corporation to a public, routable address for use outside of the corporation, such as the Internet. NAT is considered a one-to-one mapping of addresses from private to public.

PE router—provider edge router. A router that is part of a service provider's network and is connected to a customer edge (CE) router.

skinny—Skinny Client Control Protocol (SCCP). A protocol that enables CBAC to inspect Skinny control packets that are exchanged between a Skinny client and the Call Manager (CM); CBAC then configures the router (also known as the Cisco IOS Firewall) to enable the Skinny data channels to traverse through the router.

traffic filtering—A capability that allows you to configure CBAC to permit specified TCP and UDP traffic through a firewall only when the connection is initiated from within the network you want to protect. CBAC can inspect traffic for sessions that originate from either side of the firewall.

traffic inspection—CBAC inspection of traffic that travels through the firewall to discover and manage state information for TCP and UDP sessions. This state information is used to create temporary openings in the firewall's access lists to allow return traffic and additional data connections for permissible sessions (sessions that originated from within the protected internal network).

UDP— User Datagram Protocol. Connectionless transport layer protocol in the TCP/IP protocol stack. UDP is a simple protocol that exchanges datagrams without acknowledgments or guaranteed delivery, requiring that error processing and retransmission be handled by other protocols.

VPN—Virtual Private Network. Enables IP traffic to travel securely over a public TCP/IP network by encrypting all traffic from one network to another. A VPN uses "tunneling" to encrypt all information at the IP level.

vrf—A VPN routing/forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a provider edge (PE) router.

VRF table—A table that stores routing data for each VPN. The VRF table defines the VPN membership of a customer site attached to the network access server (NAS). Each VRF table comprises an IP routing table, a derived Cisco Express Forwarding (CEF) table, and guidelines and routing protocol parameters that control the information that is included in the routing table.

Note Refer to *Internetworking Terms and Acronyms* for terms not included in this glossary.

Configuring Cisco IOS Intrusion Prevention System (IPS)

This module describes how to configure the Cisco IOS Intrusion Prevention System (IPS), which helps to protect a customer's network from internal and external attacks and threats. Cisco IOS IPS restructures and replaces the existing Cisco IOS Intrusion Detection System (IDS).

Cisco IOS IPS allows customers to choose between any of the following options when loading IPS signatures onto a device:

- Loading the default, built-in signatures.
- Downloading dynamic signature detection files (SDFs). These files are dynamically updated to provide customers with the latest available versions of Cisco IOS IPS to better detect security threats.
- Loading a SDF called "attack-drop.sdf" onto their router. The attack-drop.sdf file contains 118 high fidelity IPS signatures, providing customers with the latest available detection of security threats.

Customers can download the SDF to their router from Cisco.com via the virtual private network (VPN) and Security Management Solution (VMS) IDS Management Console (MC) 2.3 network management device or via the Cisco Router and Security Device Manager (SDM). Thus VMS IDS MC or SDM can immediately begin scanning for new signatures.

Module History

This module was first published on May 2, 2005, and last updated on May 2, 2005.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all features. To find information about feature support and configurations, use the "Feature Information for Configuring Cisco IOS IPS" section on page 746.

Contents

- Prerequisites for Configuring Cisco IOS IPS, page 718
- Restrictions for Configuring Cisco IOS IPS, page 718
- Information About Cisco IOS IPS, page 719
- How to Load IPS-Based Signatures onto a Router, page 730
- Configuration Examples, page 743
- Additional References, page 745

- Feature Information for Configuring Cisco IOS IPS, page 746

Prerequisites for Configuring Cisco IOS IPS

Compatibility with VMS IDS MC 2.3 and Cisco Router SDM

VMS IDS MC provides a web-based interface for configuring, managing, and monitoring multiple IDS sensors. SDM is a web-based device-management tool that allows users to import and edit SDFs from Cisco.com to the router. VMS IDS MC is for network-wide management while SDM is for single-device management. It is strongly recommended that customers download the SDF to an IDS MC 2.3 network management device or an SDM.

Customers can choose to download the SDF to a device other than VMS IDS MC or SDM (such as a router) via command-line interface (CLI); however, this approach is not recommended because it requires that the customer know which signatures come from which signature engines.

Restrictions for Configuring Cisco IOS IPS

Signature Support Deprecation

Effective Cisco IOS Release 12.(8)T, the following signatures are no longer supported by Cisco IOS IPS:

- 1100 IP Fragment Attack (Attack, Atomic)

  Triggers when any IP datagram is received with the “more fragments” flag set to 1 or if there is an offset indicated in the offset field. (To scan for application layer signatures across fragments, you can enable virtual fragment reassembly.)

- 1105 Broadcast Source Address (Compound/Attack)

  Triggers when an IP packet with a source address of 255.255.255.255 is detected. This signature may be an indicator of an IP spoof attack or an attempt to subvert a firewall, proxy, or gateway.

- 1106 Multicast IP Source Address (Compound/Attack)

  Triggers when an IP packet with a source address of 224.x.x.x is detected. This signature may be an indicator of an IP spoof attack or an attempt to subvert a firewall, proxy, or gateway.

- 8000 FTP Retrieve Password File (Attack, Atomic) SubSig ID: 2101

  Triggers on string “passwd” issued during an FTP session. May indicate that someone is attempting to retrieve the password file from a machine to try and gain unauthorized access to system resources.

Memory Impact on Low-End to Midrange Routers

Intrusion detection configuration on certain routers may not support the complete list of signatures because of lack of sufficient memory. Thus, the network administrator may have to select a smaller subset of signatures or choose to use the standard 100 (built-in) signatures with which the routers are shipped.

Action Configuration via CLI No Longer Supported

Cisco IOS IPS actions (such as resetting the TCP connection) can no longer be configured via CLI. If you are using the attack-drop.sdf signature file, the signatures are preset with actions to mitigate the attack by dropping the packet and resetting the connection, if applicable. If you are using VMS or SDM to deploy signatures to the router, you must first tune the signatures to use the desired actions.

Any CLI that is issued to configure IPS actions will be silently ignored.

Information About Cisco IOS IPS

To help secure your network via a signature-based IPS, you should understand the following concepts:

- Cisco IOS IPS Overview, page 719
- Benefits of Cisco IOS IPS, page 719
- The Signature Definition File, page 720
- Signature Microengines: Overview and Lists of Supported Engines, page 720
- Supported Cisco IOS IPS Signatures in the attack-drop.sdf File, page 722

Cisco IOS IPS Overview

The Cisco IOS IPS acts as an in-line intrusion detection sensor, watching packets and sessions as they flow through the router and scanning each packet to match any of the Cisco IOS IPS signatures. When Cisco IOS IPS detects suspicious activity, it responds before network security can be compromised and logs the event through Cisco IOS syslog messages or Security Device Event Exchange (SDEE). The network administrator can configure Cisco IOS IPS to choose the appropriate response to various threats. When packets in a session match a signature, Cisco IOS IPS can take any of the following actions, as appropriate:

- Send an alarm to a syslog server or a centralized management interface
- Drop the packet
- Reset the connection
- Deny traffic from the source IP address of the attacker for a specified amount of time
- Deny traffic on the connection for which the signature was seen for a specified amount of time

Cisco developed its Cisco IOS software-based Intrusion-Prevention capabilities and Cisco IOS Firewall with flexibility in mind, so that individual signatures can disabled in case of false positives. Generally, it is preferable to enable both the firewall and Cisco IOS IPS to support network security policies. However, each of these features may be enabled independently and on different router interfaces.

Benefits of Cisco IOS IPS

Dynamic IPS Signatures

IPS signatures are dynamically updated and posted to Cisco.com on a regular basis. Thus, customers can access signatures that help protect their network from the latest known network attacks.

Parallel Signature Scanning

Cisco IOS IPS uses a Parallel Signature Scanning Engine to scan for multiple patterns within a signature microengine (SME) at any given time. IPS signatures are no longer scanned on a serial basis.

Named and Numbered Extended ACL Support

Cisco IOS IPS supports both named and numbered extended access control lists (ACLs) by using at least one of the following commands—**ip ips** *ips-name* **list** *acl* or **ip ips signature** *signature-id* **list** *acl-list*. (Prior to Cisco IOS Release 12.3(8)T, only standard, numbered ACLs were supported.)

The Signature Definition File

An SDF has definitions for each signature it contains. After signatures are loaded and complied onto a router running Cisco IOS IPS, IPS can begin detecting the new signatures immediately. If customers do not use the default, built-in signatures that are shipped with the routers, they can choose to download one of two different types of SDFs: the attack-drop.sdf file (which is a static file) or a dynamic SDF (which is dynamically updated and accessed from Cisco.com).

The attack-drop.sdf file is available in flash on all Cisco access routers that are shipped with Cisco IOS Release 12.3(8)T or later. The attack-drop.sdf file can then be loaded directly from flash into the Cisco IOS IPS system. If flash is erased, the attack-drop.sdf file may also be erased. Thus, if you are copying a Cisco IOS image to flash and are prompted to erase the contents of flash before copying the new image, you might risk erasing the attack-drop.sdf file. If the attack-drop.sdf file is erased, the router will refer to the built-in signatures within the Cisco IOS image. The attack-drop.sdf file can also be downloaded onto your router from Cisco.com via VMS IDS MC 2.3 or SDM.

To help detect the latest vulnerabilities, Cisco provides signature updates on Cisco.com on a regular basis. Users can use VMS or SDM to download these signature updates, tune the signature parameters as necessary, and deploy the new SDF to a Cisco IOS IPS router.

Signature Microengines: Overview and Lists of Supported Engines

Cisco IOS IPS uses signature microengines (SMEs) to load the SDF and scan signatures. Signatures contained within the SDF are handled by a variety of SMEs. The SDF typically contains signature definitions for multiple engines. The SME typically corresponds to the protocol in which the signature occurs and looks for malicious activity in that protocol.

A packet is processed by several SMEs. Each SME scans for various conditions that can lead to a signature pattern match. When an SME scans the packets, it extracts certain values, searching for patterns within the packet via the regular expression engine.

For a list of supported signature engines, see the section "Lists of Supported Signature Engines."

Lists of Supported Signature Engines

Table 35 lists supported signature engines and engine-specific parameter exceptions, if applicable.

Note If the SDF contains a signature that requires an engine that is not supported, the engine will be ignored and an error message will be displayed. If a signature within a supported engine contains a parameter that is not supported, the parameter will be ignored and an error message will be displayed.

Table 35 Supported Signature Engines for Cisco IOS IPS

Signature Engine	Initial Supported Cisco IOS Release	Parameter Exceptions[1]
ATOMIC.L3.IP	12.3(8)T	—
ATOMIC.ICMP	12.3(8)T	—
ATOMIC.IPOPTIONS	12.3(8)T	—
ATOMIC.TCP	12.3(8)T	—

Table 35 Supported Signature Engines for Cisco IOS IPS (continued)

Signature Engine	Initial Supported Cisco IOS Release	Parameter Exceptions[1]
ATOMIC.UDP	12.3(8)T	—
SERVICE.DNS	12.3(8)T	—
SERVICE.HTTP	12.3(8)T	ServicePorts (applicable only in Cisco IOS Release 12.3(8)T)
SERVICE.FTP	12.3(8)T	ServicePorts
SERVICE.SMTP	12.3(8)T	ServicePorts
SERVICE.RPC	12.3(8)T	ServicePorts, Unique, and isSweep
STRING.ICMP	12.3(14)T	—
STRING.TCP	12.3(14)T	—
STRING.UDP	12.3(14)T	—

1. The following parameters, which are defined in all signature engines, are currently not supported: AlarmThrottle=Summarize (all other values are supported), MaxInspectLength, MaxTTL, Protocol, ResetAfterIdle, StorageKey, and SummaryKey.

Table 36 lists support for the 100 signatures that are available in Cisco IOS IDS prior to Cisco IOS Release 12.3(8)T. As of Cisco IOS Release 12.3(8)T, these 100 signatures are a part of the Cisco IOS IPS built-in SDF. By default, signatures are loaded from this built-in SDF. Table 36 lists support for these 100 signatures under Cisco IOS IPS.

Note Because Cisco IOS IPS counts signatures on the basis of signature-id and subsignature-id, the 100 signatures under Cisco IOS IDS are counted as 132 signatures under Cisco IOS IPS.

Table 36 Support for Signatures Available in Cisco IOS IDS (prior to 12.3(8)T)

Signature ID	Count	Signature Engine
1000–1006	7	ATOMIC.IPOPTIONS
1101, 1102	2	ATOMIC.L3.IP
1004, 1007	2	ATOMIC.L3.IP
2000–2012, 2150	14	ATOMIC.ICMP
2151, 2154	2	ATOMIC.L3.IP
3038–3043	6	ATOMIC.TCP
3100–3107	8	SERVICE.SMTP
3153, 3154	2	SERVICE.FTP
4050–4052, 4600	4	ATOMIC.UDP
6100–6103	4	SERVICE.RPC
6150–6155	6	SERVICE.RPC
6175, 6180, 6190	3	SERVICE.RPC
6050–6057	8	SERVICE.DNS
6062–6063	2	SERVICE.DNS

Table 36 Support for Signatures Available in Cisco IOS IDS (prior to 12.3(8)T) (continued)

Signature ID	Count	Signature Engine
3215, 3229, 3223	3	SERVICE.HTTP
5034–5035	2	SERVICE.HTTP
5041, 5043–5045	4	SERVICE.HTTP
5050, 5055, 5071	3	SERVICE.HTTP
5081, 5090, 5123	3	SERVICE.HTTP
5114, 5116–5118	4	SERVICE.HTTP
1100	1	Not applicable. Signature is replaced by 12xx series.
1105–1106	2	Cisco IOS IPS deprecates these signatures, which do not appear in the SDF.
1201–1208	10	OTHER[1] (fragment attack signatures)
3050	2	OTHER[1] (SYN attack signatures)
3150–3152	3	STRING.TCP
4100	1	STRING.UDP
8000	1	Cisco IOS IPS deprecates these signatures, which do not appear in the SDF.

1. The OTHER engine contains existing, hard-coded signatures. Although the standard SDF contains an entry for these signatures, the engine is not dynamically updated. If the SDF that is loaded onto the engine does not contain the signature, the signature will be treated as though it has been disabled.

Supported Cisco IOS IPS Signatures in the attack-drop.sdf File

Customers can choose to use Cisco IOS IPS in one of the following ways:

- Download new signatures that are posted on Cisco.com. These signatures can be obtained at the Cisco Intrusion Prevention Alert Center web page. (You must have a valid Cisco.com account to access this web page.)
- Download the attack-drop.sdf file, which contains the signatures that are identified in Table 37.

Table 37 Cisco IOS IPS Signatures Supported in Cisco IOS Release 12.3(8)T

Signature ID: SubSig ID	Signature Name	Action[1]	SME	Signature Description
1006:0	IP options-Strict Source Route	A, D	ATOMIC.IPOPTIONS	Triggers on receipt of an IP datagram in which the IP option list for the datagram includes option 2 (Strict Source Routing).
1102:0	Impossible IP Packet	A, D	ATOMIC.L3.IP	Triggers when an IP packet arrives with source equal to destination address. This signature will catch the Land Attack.
1104:0	IP Localhost Source Spoof	A, D	ATOMIC.L3.IP	Triggers when an IP packet with the address of 127.0.0.1, a local host IP address that should never be seen on the network, is detected. This signature can detect the Blaster attack.

Table 37 Cisco IOS IPS Signatures Supported in Cisco IOS Release 12.3(8)T (continued)

Signature ID: SubSig ID	Signature Name	Action[1]	SME	Signature Description
1108:0	IP Packet with Proto 11	A, D	ATOMIC.L3.IP	Alarms upon detecting IP traffic with the protocol set to 11. There have been known “backdoors” running on IP protocol 11.
2154:0	Ping Of Death Attack	A, D	ATOMIC.L3.IP	Triggers when an IP datagram is received with the protocol field in the IP header set to 1 (Internet Control Message Protocol [ICMP]), the Last Fragment bit is set. The IP offset (which represents the starting position of this fragment in the original packet and which is in 8-byte units) plus the rest of the packet is greater than the maximum size for an IP packet.
3038:0	Fragmented NULL TCP Packet	A, D	ATOMIC.TCP	Triggers when a single, fragmented TCP packet with none of the SYN, FIN, ACK, or RST flags set has been sent to a specific host. A reconnaissance sweep of your network may be in progress.
3039:0	Fragmented Orphaned FIN packet	A, D	ATOMIC.TCP	Triggers when a single, fragmented, orphan TCP FIN packet is sent to a privileged port (having a port number less than 1024) on a specific host. A reconnaissance sweep of your network may be in progress.
3040:0	NULL TCP Packet	A, D	ATOMIC.TCP	Triggers when a single TCP packet with none of the SYN, FIN, ACK, or RST flags set has been sent to a specific host. A reconnaissance sweep of your network may be in progress.
3041:0	SYN/FIN Packet	A, D	ATOMIC.TCP	Triggers when a single TCP packet with the SYN and FIN flags set is sent to a specific host. A reconnaissance sweep of your network may be in progress. The use of this type of packet indicates an attempt to conceal the sweep.
3043:0	Fragmented SYN/FIN Packet	A, D	ATOMIC.TCP	Triggers when a single, fragmented TCP packet with the SYN and FIN flags set is sent to a specific host. A reconnaissance sweep of your network may be in progress. The use of this type of packet indicates an attempt to conceal the sweep.
3129:0	Mimail Virus C Variant File Attachment	A, D, R	SERVICE. SMTP	Fires when an e-mail attachment matching the C Variant of the Mimail virus is detected. The virus sends itself to recipients as the e-mail attachment “photos.zip” that contains the file “photos.jpg.exe” and has “our private photos” in the e-mail subject line. If launched, the virus harvests e-mail addresses and possible mail servers from the infected system.
3140:3	Bagle Virus Activity[2]	A, D, R	SERVICE.HTTP	Fires when HTTP propagation using .jpeg associated with the .Q variant is detected.

Table 37 Cisco IOS IPS Signatures Supported in Cisco IOS Release 12.3(8)T (continued)

Signature ID: SubSig ID	Signature Name	Action[1]	SME	Signature Description
3140:4	Bagle Virus Activity[3]	A, D, R	SERVICE.HTTP	Fires when HTTP propagation using .php associated with the .Q variant is detected.
3300:0	NetBIOS OOB Data	A, D	ATOMIC.TCP	Triggers when an attempt to send Out Of Band data to port 139 is detected.
5045:0	WWW xterm display attack	A, D, R	SERVICE.HTTP	Triggers when any cgi-bin script attempts to execute the command xterm -display. An attempt to illegally log into your system may be in progress.
5047:0	WWW Server Side Include POST attack	A, D, R	SERVICE.HTTP	Triggers when an attempt is made to embed a server side include (SSI) in an http POST command. An attempt to illegally access system resources may be in progress.
5055:0	HTTP Basic Authentication Overflow	A, D	SERVICE.HTTP	A buffer overflow can occur on vulnerable web servers if a very large username and password combination is used with basic authentication.
5071:0	WWW msacds.dll Attack	A, D, R	SERVICE.HTTP	An attempt has been made to execute commands or view secured files, with privileged access. Administrators are highly recommended to check the affected systems to ensure that they have not been illicitly modified.
5081:0	WWW WinNT cmd.exe Access	A, D, R	SERVICE.HTTP	Triggers when the use of the Windows NT cmd.exe is detected in a URL. This signature can detect the NIMDA attack.
5114: 0 5114:1 5114:2	WWW IIS Unicode Attack	A, D, R	SERVICE.HTTP	Triggers when an attempt to exploit the Unicode ../ directory traversal vulnerability is detected. Looks for the commonly exploited combinations that are included in publicly available exploit scripts. SubSig 2 is know to detect the NIMDA attack.
5126:0	WWW IIS .ida Indexing Service Overflow	A, D, R	SERVICE.HTTP	Alarms if web traffic is detected with the ISAPI extension .ida? and a data size of greater 200 characters.
5159:0	phpMyAdmin Cmd Exec	A, D, R	SERVICE.HTTP	Triggers when access to sql.php with the arguments goto and btnDrop=No is detected.
5184:0	Apache Authentication Module ByPass	A, D, R	SERVICE.HTTP	Fires upon detecting a select statement on the Authorization line of an HTTP header.
5188:0	HTTP Tunneling[4] SubSig 0: GotomyPC	A, D, R	SERVICE.HTTP	Triggers when a computer connects to gotomyPC site.
5188:1	HTTP Tunneling SubSig 1: FireThru	A, D, R	SERVICE.HTTP	Triggers when an attempt to use /cgi-bin/proxy is detected. The /cgi-bin/proxy is used to tunnel connections to other ports using web ports.

Table 37 Cisco IOS IPS Signatures Supported in Cisco IOS Release 12.3(8)T (continued)

Signature ID: SubSig ID	Signature Name	Action[1]	SME	Signature Description
5188:2	HTTP Tunneling SubSig 2: HTTP Port	A, D, R	SERVICE.HTTP	Triggers when a connection is made to exectech-va.com. The site runs a server, which connects to the requested resource and passes the information back to the client on web ports.
5188:3	HTTP Tunneling SubSig 3: httptunnel	A, D, R	SERVICE.HTTP	Triggers when /index/html? is detected on POST request.
5245:0	HTTP 1.1 Chunked Encoding Transfer	A, D, R	SERVICE.HTTP	Fires when HTTP 1.1 chunked encoding transfer activity is detected. This signature is known to detect the Scalper Worm.
5326:0	Root.exe access	A, D, R	SERVICE.HTTP	Alarms upon detecting an HTTP request for root.exe. This signature is known to detect the NIMDA attack.
5329:0	Apache/mod_ssl Worm Probe	A, D, R	SERVICE.HTTP	Fires when a probe by the Apache/mod_ssl worm is detected. If the worm detects a vulnerable web server, a buffer overflow attack is sent to HTTPS port (TCP 443) of the web server. The worm then attempts to propagate itself to the newly infected web server and begins scanning for new hosts to attack.
5364:0	IIS WebDAV Overflow	A, D, R	SERVICE.HTTP	Fires when a long HTTP request (65000+ characters) is detected with an HTTP header option “Translate:”. An attack to exploit a weakness in the WebDAV component of the IIS web server may be in progress.
5390:0	Swen Worm HTTP Counter Update Attempt	A, D, R	SERVICE.HTTP	Triggers when an attempt to access the URL “/bin/counter.gif/link=bacillus” is detected. A system may be infected by the Swen worm trying the update a counter on a web page located on the server “ww2.fce.vutbr.cz.”
5400:0	Beagle.B (Bagle.B) Web Beacon	A, D, R	SERVICE.HTTP	Fires when a request is made for the script 1.php or 2.php residing on the hosts “www.47df.de” or “www.strato.de,” followed by the argument indicating the trojan’s listening port number, p=8866.
6055:0 6055:1 6055:2	DNS Inverse Query Buffer Overflow	A, D R for subsig 1, 2	SERVICE.DNS	Triggers when an IQUERY request arrives with a data section that is greater than 255 characters.

Table 37 Cisco IOS IPS Signatures Supported in Cisco IOS Release 12.3(8)T (continued)

Signature ID: SubSig ID	Signature Name	Action[1]	SME	Signature Description
6056:0 6056:1 6056:2	DNS NXT Buffer Overflow	A, D R for subsig 1, 2	SERVICE.DNS	Triggers when a Domain Name System (DNS) server response arrives with a long NXT resource where the length of the resource data is greater than 2069 bytes or the length of the TCP stream containing the NXT resource is greater than 3000 bytes.
6057:0 6057:1 6057:2	DNS SIG Buffer Overflow	A, D R for subsig 1, 2	SERVICE.DNS	Triggers when a DNS server response arrives with a long SIG resource where the length of the resource data is greater than 2069 bytes or the length of the TCP stream that contains the SIG resource is greater than 3000 bytes.
6058:0 6058:1	DNS SRV DoS	A, D R for subsig 1	SERVICE.DNS	Alarms when a DNS query type SRV and DNS query class IN is detected with more than ten pointer jumps in the SRV resource record.
6059:0 6059:1 6059:2	DNS TSIG Overflow	A, D R for subsig 2	SERVICE.DNS	Alarms when a DNS query type TSIG is detected and the domain name is greater than 255 characters. This signature is known to detect the Lion work.
6060:0 6060:1 6060:2 6060:3	DNS Complian Overflow	A, D R for subsig 2, 3	SERVICE.DNS	Alarms when a Name Server (NS) record is detected with a domain name greater than 255 characters and the IP address is 0.0.0.0, 255.255.255.255 or a multicast address of the form 224.x.x.x.
6100:0 6100:1	RPC Port Registration	A, D R for subsig 1	SERVICE.RPC	Triggers when attempts are made to register new RPC services on a target host. Port registration is the method used by new services to report their presence to the portmapper and to gain access to a port. Their presence is then advertised by the portmapper.
6101:0 6101:1	RPC Port Unregistration	A, D R for subsig 1	SERVICE.RPC	Triggers when attempts are made to unregister existing Remote Procedure Call (RPC) services on a target host. Port unregistration is the method used by services to report their absence to the portmapper and to remove themselves from the active port map.
6104:0 6104:1	RPC Set Spoof	A, D R for subsig 1	SERVICE.RPC	Triggers when an RPC set request with a source address of 127.x.x.x is detected.
6105:0 6105:1	RPC Unset Spoof	A, D R for subsig 1	SERVICE.RPC	Triggers when an RPC unset request with a source address of 127.x.x.x is detected.
6188:0	statd dot dot	A, D	SERVICE.RPC	Alarms upon detecting a dot dot slash (../) sequence sent to the statd RPC service.

Table 37 Cisco IOS IPS Signatures Supported in Cisco IOS Release 12.3(8)T (continued)

Signature ID: SubSig ID	Signature Name	Action[1]	SME	Signature Description
6189:0 6189:1	statd automount attack	A, D R for subsig 1	SERVICE.RPC	Alarms upon detecting a statd bounce attack on the automount process. This attack targets a vulnerability in the automount process that could be exploited only via localhost.
6190:0 6190:1	statd Buffer Overflow	A, D R for subsig 1	SERVICE.RPC	Triggers when a large statd request is sent. This attack could be an attempt to overflow a buffer and gain access to system resources.
6191:0 6191:1	RPC.tooltalk buffer overflow	A, D R for subsig 1	SERVICE.RPC	Fires when an attempt is made to overflow an internal buffer in the tooltalk rpc program.
6192:0 6192:1	RPC mountd Buffer Overflow	A, D R for subsig 1	SERVICE.RPC	Triggers on an attempt to overflow a buffer in the RPC mountd application. This attack may result in unauthorized access to system resources.
6193:0 6193:1	RPC CMSD Buffer Overflow	A, D R for subsig 1	SERVICE.RPC	Fires when an attempt is made to overflow an internal buffer in the Calendar Manager Service Daemon, rpc.cmsd.
6194:0 6194:1	sadmind RPC Buffer Overflow	A, D R for subsig 1	SERVICE.RPC	Fires when a call to RPC program number 100232 procedure 1 with a UDP packet length greater than 1024 bytes is detected.
6195:0 6195:1	RPC amd Buffer Overflow	A, D R for subsig 1	SERVICE.RPC	Detects the exploitation of the RPC AMD Buffer Overflow vulnerability. The trigger for this signature is an RPC call to the berkeley automounter daemons rpc program (300019) procedure 7 that has a UDP length greater than 1024 bytes or a TCP stream length greater than 1024 bytes. The TCP stream length is defined by the contents of the two bytes preceding the RPC header in a TCP packet.
6196:0 6196:1	snmpXdmid Buffer Overflow	A, D R for subsig 1	SERVICE.RPC	Fires when an abnormally long call to the RPC program 100249 (snmpXdmid) and procedure 257 is detected.
6197:0 6197:1	rpc yppaswdd overflow	A, D R for subsig 0	SERVICE.RPC	Fires when an overflow attempt is detected. This alarm looks for an abnormally large argument in the attempt to access yppaswdd.
6276:0 6276:1	TooltalkDB overflow	A, D R for subsig 1	SERVICE.RPC	Alarms upon detecting an RPC connection to rpc program number 100083 using procedure 103 with a buffer greater than 1024.
9200:0	Back Door Response (TCP 12345)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 12345, which is a known trojan port for NetBus as others.

Table 37 Cisco IOS IPS Signatures Supported in Cisco IOS Release 12.3(8)T (continued)

Signature ID: SubSig ID	Signature Name	Action[1]	SME	Signature Description
9201:0	Back Door Response (TCP 31337)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 31337, which is a known trojan port for BackFire.
9202:0	Back Door Response (TCP 1524)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 1524, which is a common back door placed on machines by worms and hackers.
9203:0	Back Door Response (TCP 2773)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 2773, which is a known trojan port for SubSeven.
9204:0	Back Door Response (TCP 2774)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 2774, which is a known trojan port for SubSeven.
9205:0	Back Door Response (TCP 20034)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 20034, which is a known trojan port for Netbus Pro.
9206:0	Back Door Response (TCP 27374)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 27374, which is a known trojan port for SubSeven.
9207:0	Back Door Response (TCP 1234)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 1234, which is a known trojan port for SubSeven.
9208:0	Back Door Response (TCP 1999)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 1999, which is a known trojan port for SubSeven.
9209:0	Back Door Response (TCP 6711)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 6711, which is a known trojan port for SubSeven.
9210:0	Back Door Response (TCP 6712)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 6712, which is a known trojan port for SubSeven.
9211:0	Back Door Response (TCP 6713)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 6713, which is a known trojan port for SubSeven.
9212:0	Back Door Response (TCP 6776)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 6776, which is a known trojan port for SubSeven.
9213:0	Back Door Response (TCP 16959)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 16959, which is a known trojan port for SubSeven.
9214:0	Back Door Response (TCP 27573)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 27573, which is a known trojan port for SubSeven.

Table 37 Cisco IOS IPS Signatures Supported in Cisco IOS Release 12.3(8)T (continued)

Signature ID: SubSig ID	Signature Name	Action[1]	SME	Signature Description
9215:0	Back Door Response (TCP 23432)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 23432, which is a known trojan port for asylum.
9216:0	Back Door Response (TCP 5400)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 5400, which is a known trojan port for back-construction.
9217:0	Back Door Response (TCP 5401)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 5401, which is a known trojan port for back-construction.
9218:0	Back Door Response (TCP 2115)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 2115, which is a known trojan port for bugs.
9223:0	Back Door Response (TCP 36794)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 36794, which is a known trojan port for NetBus as well Bugbear.
9224:0	Back Door Response (TCP 10168)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 10168, which is a known trojan port for lovegate.
9225:0	Back Door Response (TCP 20168)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 20168, which is a known trojan port for lovegate.
9226:0	Back Door Response (TCP 1092)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 1092, which is a known trojan port for lovegate.
9227:0	Back Door Response (TCP 2018)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 2018, which is a known trojan port for fizzer.
9228:0	Back Door Response (TCP 2019)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 2019, which is a known trojan port for fizzer.
9229:0	Back Door Response (TCP 2020)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 2020, which is a known trojan port for fizzer.
9230:0	Back Door Response (TCP 2021)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 2021, which is a known trojan port for fizzer.
9231:0	Back Door Response (TCP 6777)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 6777, which is a known trojan port for Beagle (Bagle).
9232:0	Back Door Response (TCP 5190)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 5190, which is a known trojan port for the Anig worm.

Table 37 Cisco IOS IPS Signatures Supported in Cisco IOS Release 12.3(8)T (continued)

Signature ID: SubSig ID	Signature Name	Action[1]	SME	Signature Description
9233:0	Back Door Response (TCP 3127)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 3127, which is a known trojan port for the MyDoom.A / Novarg.A virus.
9236:0	Back Door Response (TCP 3128)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 3128, which is a known trojan port for the MyDoom.B / Novarg.B virus.
9237:0	Back Door Response (TCP 8866)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 8866, which is a known trojan port for the Beagle.B (Bagle.B) virus.
9238:0	Back Door Response (TCP 2766)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 2766, which is a known trojan port for the DeadHat worm.
9239:0	Back Door Response (TCP 2745)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 2745, which is a known trojan port for the Bagle.H-J virus.
9240:0	Back Door Response (TCP 2556)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 2556, which is a known trojan port for the Bagle (.M.N.O.P) virus.
9241:0	Back Door Response (TCP 4751)	A, D	ATOMIC.TCP	Fires upon detecting a TCP SYN/ACK packet from port 4751, which is a known trojan port for the Bagle.U virus.

1. A = alarm, D = drop, R = reset
2. This signature requires port to application mapping (PAM) configuration via the command **ip port-map http port 81**.
3. This signature requires PAM configuration via the command **ip port-map http port 81**.
4. This signature requires PAM configuration via the command **ip port-map http port 8200**.

How to Load IPS-Based Signatures onto a Router

Before configuring Cisco IOS IPS on a router, you should determine which one of the following deployment scenarios best addresses your situation and configure the associated task, as appropriate:

- You are loading signatures onto a router via VMS IDS MC or SDM:

  To use VMS IDS MC, see the documents on the VMS index.

  To use SDM, see the document *SDM Intrusion Prevention System (IPS)* User's Guide.
- You are installing a new router with the latest version of Cisco IOS IPS.

  To perform this task, see the section "Installing Cisco IOS IPS on a New Router, page 731."
- Your network is transitioning to Cisco IOS IPS in Cisco IOS Release 12.3(8)T or later.

  To perform this task, see the section "Upgrading to the Latest Cisco IOS IPS Signature Definition File (SDF), page 733."
- You are merging the default (built-in) Cisco IOS IPS signatures with the latest version of the Cisco IOS IPS signature detection file, "attack-drop.sdf."

To perform this task, see the section "Merging Built-In Signatures with the attack-drop.sdf File, page 735"

After you have configured Cisco IOS IPS on your router, see the following optional sections:

- Monitoring Cisco IOS IPS Signatures via Syslog Messages or SDEE, page 739
- Troubleshooting Cisco IOS IPS, page 741

Installing Cisco IOS IPS on a New Router

Perform this task to install the latest Cisco IOS IPS signatures on a router for the first time.

This task allows you to load the default, built-in signatures or the SDF called "attack-drop.sdf"—but not both. If you want to merge the two signature files, you must load the default, built-in signatures as described in this task. Then, you can merge the default signatures with the attack-drop.sdf file as described in the task "Merging Built-In Signatures with the attack-drop.sdf File."

Note Installing the signatures provided in flash is the recommended method in Cisco IOS Release 12.3(8)T for IPS attack mitigation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ips sdf location** *url*
4. **ip ips name** *ips-name* [**list** *acl*]
5. **ip ips signature** *signature-id* [**:***sub-signature-id*] {**delete** | **disable** | **list** *acl-list*}
6. **ip ips deny-action ips-interface**
7. **interface** *type name*
8. **ip ips** *ips-name* {**in** | **out**}
9. **exit**
10. **show ip ips configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip ips sdf location** *url* **Example:** `Router(config)# ip ips sdf location disk2:attack-drop.sdf`	(Optional) Specifies the location in which the router will load the SDF, "attack-drop.sdf." **Note** If this command is not issued, the router will load the default, built-in signatures.
Step 4	**ip ips name** *ips-name* [**list** *acl*] **Example:** `Router(config)# ip ips name MYIPS`	Creates an IPS rule.
Step 5	**ip ips signature** *signature-id* [**:***sub-signature-id*] {**delete** | **disable** | **list** *acl-list*} **Example:** `Router(config)# ip ips signature 1000 disable`	(Optional) Attaches a policy to a given signature.
Step 6	**ip ips deny-action ips-interface** **Example:** `Router(config)# ip ips deny-action ips-interface`	(Optional) Creates an ACL filter for the deny actions (denyFlowInline and denyConnectionInline) on the IPS interface rather than the ingress interface. **Note** You should configure this command only if at least one signature is configured to use the supported deny actions, if the input interface is configured for load balancing, and if IPS is configured on the output interface.
Step 7	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet0/1`	Configures an interface type and enters interface configuration mode.
Step 8	**ip ips** *ips-name* {**in** | **out**} **Example:** `Router(config-if)# ip ips MYIPS in`	Applies an IPS rule at an interface and automatically loads the signatures and builds the signature engines. **Note** Whenever signatures are replaced or merged, the router prompt is suspended while the signature engines for the newly added or merged signatures are being built. The router prompt will be available again after the engines are built. Depending on your platform and how many signatures are being loaded, building the engine can take up to several seconds. It is recommended that you enable logging messages to monitor the engine building status.

	Command or Action	Purpose
Step 9	**exit** **Example:** `Router(config-if)# exit` **Example:** `Router(config)# exit`	Exits interface and global configuration modes.
Step 10	**show ip ips configuration** **Example:** `Router# show ip ips configuration`	(Optional) Verifies that Cisco IOS IPS is properly configured.

Upgrading to the Latest Cisco IOS IPS Signature Definition File (SDF)

Perform this task to replace the existing signatures on your router with the latest IPS signature file, attack-drop.sdf.

Note The latest IPS image will read and convert all commands that begin with the words "ip audit" to "ip ips." For example, the **ip audit name** command will become the **ip ips name** command.

Although IPS will accept the **audit** keyword, it will generate the **ips** keyword when you show the configuration. Also, if you issue the help character (?), the CLI will display the **ips** keyword instead of the **audit** keyword, and the Tab key used for command completion will not recognize the **audit** keyword.

Prerequisites

To install Cisco IOS IPS, you should load a new Cisco IOS image to your router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ips name** *ips-name*
4. **ip ips sdf location** *url*
5. **no ip ips location in builtin**
6. **ip ips fail closed**
7. **ip ips deny-action ips-interface**
8. **interface** *type name*
9. **ip ips** *ips-name* {**in** | **out**} [**list** *acl*]
10. **exit**
11. **show ip ips configuration**
12. **show ip ips signatures** [**detailed**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip ips name** *ips-name* **Example:** `Router(config)# ip ips name MIPS`	Creates an IPS rule.
Step 4	**ip ips sdf location** *url* **Example:** `Router(config)# ip ips sdf location disk2:attack-drop.sdf`	(Optional) Specifies the location where the router will load the SDF. If this command is not issued, the router will load the default SDF.
Step 5	**no ip ips location in builtin** **Example:** `Router(config)# no ip ips location in builtin`	(Optional) Instructs the router not load the built-in signatures if it cannot find the specified signature file. If this command is not issued, the router will load the built-in signatures if the SDF is not found. **Caution** If this command is issued and IPS fails to load the SDF, you will receive an error message stating that IPS is completely disabled.
Step 6	**ip ips fail closed** **Example:** `Router(config)# ip ips fail closed`	(Optional) Instructs the router to drop all packets until the signature engine is built and ready to scan traffic. If this command is issued, one of the following scenarios will occur: • If IPS fails to load the SDF, all packets will be dropped—unless the user specifies an ACL for packets to send to IPS. • If IPS successfully loads the SDF but fails to build a signature engine, all packets that are destined for that engine will be dropped. If this command is not issued, all packets will be passed without scanning if the signature engine fails to build.

	Command or Action	Purpose
Step 7	**ip ips deny-action ips-interface** **Example:** `Router(config)# ip ips deny-action ips-interface`	(Optional) Creates an ACL filter for the deny actions (denyFlowInline and denyConnectionInline) on the IPS interface rather than the ingress interface. **Note** You should configure this command only if at least one signature is configured to use the supported deny actions, if the input interface is configured for load balancing, and if IPS is configured on the output interface.
Step 8	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet0/1`	Configures an interface type and enters interface configuration mode.
Step 9	**ip ips** *ips-name* {**in** | **out**} [**list** *acl*] **Example:** `Router(config-if)# ip ips MYIPS in`	Applies an IPS rule at an interface and automatically loads the signatures and builds the signature engines. • **list** *acl*—Packets that are permitted via a specified ACL will be scanned by IPS. **Note** Whenever signatures are replaced or merged, the router prompt is suspended while the signature engines for the newly added or merged signatures are being built. The router prompt will be available again after the engines are built.
Step 10	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 11	**show ip ips configuration** **Example:** `Router# show ip ips configuration`	(Optional) Verifies that Cisco IOS IPS is properly configured.
Step 12	**show ip ips signatures** [**detailed**] **Example:** `Router# show ip ips signatures`	(Optional) Verifies signature configuration, such as signatures that have been disabled.

Merging Built-In Signatures with the attack-drop.sdf File

You may want to merge the built-in signatures with the attack-drop.sdf file if the built-in signatures are not providing your network with adequate protection from security threats. Perform this task to add the SDF and to change default parameters for a specific signature within the SDF or signature engine.

Prerequisites

Before you can merge the attack-drop.sdf file with the built-in signatures, you should already have the built-in signatures loaded onto the router as described in the task "Installing Cisco IOS IPS on a New Router."

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no ip ips location in builtin**
4. **ip ips fail closed**
5. **exit**
6. **copy** [**/erase**] *url* **ips-sdf**
7. **copy ips-sdf** *url*
8. **configure terminal**
9. **ip ips signature** *signature-id*[**:***sub-signature-id*] {**delete** | **disable** | **list** *acl-list*}
10. **ip ips sdf location** *url*
11. **ip ips deny-action ips-interface**
12. **interface** *type name*
13. **ip ips** *ips-name* {**in** | **out**}
14. **exit**
15. **exit**
16. **show ip ips signatures** [**detailed**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**no ip ips location in builtin** **Example:** `Router(config)# no ip ips location in builtin`	(Optional) Instructs the router not to load the built-in signatures if it cannot find the specified signature file. If this command is not issued, the router will load the built-in signatures if the SDF is not found. **Caution** If this command is issued and IPS fails to load the SDF, you will receive an error message stating that IPS is completely disabled.

	Command or Action	Purpose
Step 4	**ip ips fail closed** **Example:** `Router(config)# ip ips fail closed`	(Optional) Instructs the router to drop all packets until the signature engine is built and ready to scan traffic. If this command is issued, one of the following scenarios will occur: • If IPS fails to load the SDF, all packets will be dropped—unless the user specifies an ACL for packets to send to IPS. • If IPS successfully loads the SDF but fails to build a signature engine, all packets that are destined for that engine will be dropped. If this command is not issued, all packets will be passed without scanning if the signature engine fails to build.
Step 5	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 6	**copy** [**/erase**] *url* **ips-sdf** **Example:** `Router# copy disk2:attack-drop.sdf ips-sdf`	Loads the SDF in the router. The SDF will merge with the signatures that are already loaded in the router, unless the **/erase** keyword is issued. The **/erase** keyword replaces the built-in signatures with the SDF. **Note** The SDF location is not saved in the configuration. The next time the router is reloaded, it will refer to a previously specified SDF location in the configuration or it will load the built-in signatures. **Note** Whenever signatures are replaced or merged, the router prompt is suspended while the signature engines for the newly added or merged signatures are being built. The router prompt will be available again after the engines are built. Depending on your platform and how many signatures are being loaded, building the engine can take up to several seconds. It is recommended that you enable logging messages to monitor the engine building status.
Step 7	**copy ips-sdf** *url* **Example:** `Router# copy ips-sdf disk2:my-signatures.sdf`	Saves the SDF that was loaded in the previous step to a specified location. The SDF location will not be saved unless this command is issued.
Step 8	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 9	**ip ips signature** *signature-id*[**:***sub-signature-id*] {**delete** | **disable** | **list** *acl-list*} **Example:** `Router(config)# ip ips signature 1107 disable`	(Optional) Instructs the router to scan for the specified signature but not take any action if the signature is detected.
Step 10	**ip ips sdf location** *url* **Example:** `Router(config)# ip ips sdf location disk2:my-signatures.sdf`	Configures the router to initialize the new SDF.
Step 11	**ip ips deny-action ips-interface** **Example:** `Router(config)# ip ips deny-action ips-interface`	(Optional) Creates an ACL filter for the deny actions (denyFlowInline and denyConnectionInline) on the IPS interface rather than the ingress interface. **Note** You should configure this command only if at least one signature is configured to use the supported deny actions, if the input interface is configured for load balancing, and if IPS is configured on the output interface.
Step 12	**interface** *type name* **Example:** `Router(config)# interface GigabitEthernet0/1`	Configures an interface type and enters interface configuration mode.
Step 13	**ip ips** *ips-name* {**in** | **out**} **Example:** `Router(config-if)# ip ips MYIPS in`	Applies an IPS rule at an interface and reloads the router and reinitializes Cisco IOS IPS. • **list** *acl*—Packets that are permitted via a specified ACL will be scanned by IPS. **Note** The router prompt will disappear while the signatures are loading and the signature engines are building. The router prompt will reappear after the signatures have been loaded and the signature engines have been built.
Step 14	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 15	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 16	**show ip ips signatures** [**detailed**] **Example:** `Router# show ip ips signatures`	(Optional) Verifies signature configuration, such as signatures that have been disabled or marked for deletion.

Monitoring Cisco IOS IPS Signatures via Syslog Messages or SDEE

Cisco IOS IPS provides two methods to report IPS intrusion alerts—Cisco IOS logging (syslog) and SDEE. Perform this task to enable SDEE to report IPS intrusion alerts.

To configure syslog messages, see the chapter "Troubleshooting and Fault Management" in the *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.4.

Note Effective Cisco IOS Release 12.3(14)T, the Post Office Protocol is no longer supported. To configure the Post Office Protocol, see the chapter "Configuring Cisco IOS Firewall Intrusion Detection System" in the *Cisco IOS Security Configuration Guide*, Release 12.2.

SDEE Overview

SDEE is an application-level communication protocol that is used to exchange IPS messages between IPS clients and IPS servers.

SDEE is always running, but it does not receive and process events from IPS unless SDEE notification is enabled. If SDEE notification is not enabled and a client sends a request, SDEE will respond with a fault response message, indicating that notification is not enabled.

Storing SDEE Events in the Buffer

When SDEE notification is enabled (via the **ip ips notify sdee** command), 200 events can automatically be stored in the buffer. When SDEE notification is disabled, all stored events are lost. A new buffer is allocated when the notifications are reenabled.

When specifying the size of an events buffer, note the following functionality:

- It is circular. When the end of the buffer is reached, the buffer will start overwriting the earliest stored events. (If overwritten events have not yet been reported, you will receive a buffer overflow notice.)
- If a new, smaller buffer is requested, all events that are stored in the previous buffer will be lost.
- If a new, larger buffer is requested, all existing events will be saved.

Prerequisites

To use SDEE, the HTTP server must be enabled (via the **ip http server** command). If the HTTP server is not enabled, the router cannot respond to the SDEE clients because it cannot not "see" the requests.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ips notify sdee**
4. **ip sdee events** *events*
5. **ip sdee subscriptions** *subscriptions*
6. **exit**
7. **show ip sdee** {[**alerts**] [**all**] [**errors**] [**events**] [**configuration**] [**status**] [**subscriptions**]}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip ips notify sdee** **Example:** `Router(config)# ip ips notify sdee`	Enables SDEE event notification on a router.
Step 4	**ip sdee events** *events* **Example:** `Router(config)# ip sdee events 500`	(Optional) Sets the maximum number of SDEE events that can be stored in the event buffer. Maximum value: 1000 events. **Note** By default, 200 events can be stored in the buffer when SDEE is enabled. When SDEE is disabled, all stored events are lost; a new buffer is allocated when the notifications are reenabled.
Step 5	**ip sdee subscriptions** *subscriptions* **Example:** `Router(config)# ip sdee subscriptions 1`	(Optional) Sets the maximum number of SDEE subscriptions that can be open simultaneously. Valid value ranges from 1 to 3.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 7	**show ip sdee** {[**alerts**] [**all**] [**errors**] [**events**] [**configuration**] [**status**] [**subscriptions**]} **Example:** `Router# show ip sdee configuration`	(Optional) Verifies SDEE configuration information and notification functionality.

Troubleshooting Tips

To print out new SDEE alerts on the router console, issue the **debug ip sdee** command.

To clear the event buffer or SDEE subscriptions from the router (which helps with error recovery), issue the **clear ip sdee** command.

Troubleshooting Cisco IOS IPS

This section contains the following information, which may help you troubleshoot Cisco IOS IPS:

- Interpreting Cisco IOS IPS System Messages, page 741
- Conditions of an SME Build Failure, page 743

Interpreting Cisco IOS IPS System Messages

Table 38 lists some of the alarm, status, and error messages that may be shown when using Cisco IOS IPS.

Table 38 Cisco IOS IPS System Messages

System Message	Description
Alarm Messages	
`%IPS-4-SIGNATURE:Sig:1107 Subsig:0 Sev:2 RFC1918 address [192.168.121.1:137 -> 192.168.121.255:137]`	An IPS signature has been triggered.
`%IPS-5-SIGNATURE:Sig:1107 Subsig:0 Global Summary:50 alarms in this interval`	A flood of the specified IPS signature has been seen and summarized. (For example, signature 1107 has been seen 50 times.)
Status Messages	
`%IPS-6-ENGINE_READY:SERVICE.HTTP - 183136 ms - packets for this engine will be scanned`	An IPS signature engine has been built and is ready to scan packets.
`%IPS-6-ENGINE_BUILD_SKIPPED:STRING.UDP - there are no new signature definitions for this engine`	There are not any signature definitions or changes to the existing signature definitions of an IPS signature engine, and the engine does not have to be rebuilt.
`%IPS-5-PACKET_DROP:SERVICE.DNS - packets dropped while engine is building`	Packets are being dropped because the specified IPS module is not functioning and the **ip ips fail closed** command is configured. The message is rate limited to 1 message per 60 seconds.
`%IPS-5-PACKET_UNSCANNED:SERVICE.DNS - packets passed unscanned while engine is building`	Packets are passing through the network but are not being scanned because the specified IPS module is not functioning and the **ip ips fail closed** command is not configured. The message is rate limited to 1 message per 60 seconds.

Table 38 Cisco IOS IPS System Messages (continued)

System Message	Description
`%IPS-6-SDF_LOAD_SUCCESS:SDF loaded successfully from flash:sdf_8http.xml`	An SDF is successfully loaded from a given location.
Error Messages	
`%IPS-3-BUILTIN_SIGS:Configured to load builtin signatures` `%IPS-3-BUILTIN_SIGS:Not Configured to load builtin signatures` `%IPS-3-BUILTIN_SIGS:Failed to load builtin signatures`	One of these three messages can be displayed when IPS loads the built-in signatures.
`%IPS-5-ENGINE_UNKNOWN: SERVICE.GENERIC - unknown engine encountered while parsing SDF`	The router has encountered an unknown and unsupported signature engine while parsing the SDF. To prevent this message from being generated again, ensure that the SDF being loaded on the router does not contain any engines that are not supported by IPS.
`%IPS-5-UNSUPPORTED_PARAM: SERVICE.RPC 6275:1 isSweep=False - bad parameter - removing parameter`	The router has encountered an unsupported parameter while parsing the SDF. The signature is deleted if the unsupported parameter is required for the signature. The parameter is removed from the signature if it is not required. To prevent this message from being generated again, ensure that the SDF being loaded on the router does not contain any parameters that are not supported by IPS.
`%IPS-3-ENGINE_BUILD_FAILED: SERVICE.HTTP - 158560 ms - engine build`	One of the signature engines fails to build after an SDF is loaded. A message is sent for each engine that fails. An engine typically fails to build because of low memory, so increasing router memory can alleviate the problem. Also, try to load the SDF immediately after a route reboots, which is when system resources are available.
`%IPS-4-SDF_PARSE_FAILED: not well-formed (invalid token) at Line 1 Col 0 Byte 0 Len 1006`	An SDF has not parsed correctly. The SDF might have been corrupt.
`%IPS-4-SDF_LOAD_FAILED: failed to parse SDF from tftp://tftp-server/sdf.xml`	An SDF fails to load. The SDF may fail for any of the following reasons: • Fails to load if it resides on a network server that cannot be reached • Does not have the correct read permissions
`%IPS-2-DISABLED: IPS removed from all interfaces - IPS disabled`	IPS has been disabled. This message will indicate why IPS has been disabled.

Conditions of an SME Build Failure

Sometimes an SME that is being built will fail. The SME can fail because it is attempting to load a corrupted SDF file or it exceeds memory limitations of the router. If a failure occurs, Cisco IOS IPS is designed to handle it. Possible failures are as follows:

- By default, IPS is designed to "fail open," which means that if an SME does not build, all packets that are destined for that particular engine will pass traffic without scanning.
- If IPS cannot load the attack-drop.sdf file onto a router, the router will revert to the previously loaded available signatures. (In most cases, the previously loaded signatures are the Cisco IOS built-in signatures.)
- If an engine build fails when you are merging the attack-drop.sdf file with the built-in signatures, IPS will revert, by default, to the previously available engine (or engines).

The default behavior for engine failure allows for packets to be passed unscanned. To prevent traffic from being passed unscanned, issue the **ip ips fail closed** command, which forces the router to drop all packets if an SME build fails.

Note If a signature or a signature parameter is not supported, Cisco IOS will print a syslog message, indicating that the signature or parameter is not supported.

Configuration Examples

This section contains the following configuration examples:

- Loading the Default Signatures: Example, page 743
- Loading the attack-drop.sdf: Example, page 744
- Merging the attack-drop.sdf File with the Default, Built-in Signatures: Example, page 744

Loading the Default Signatures: Example

The following example shows the Cisco IOS IPS commands required to load the default, built-in signatures. Note that a configuration option for specifying an SDF location is not necessary; built-in signatures reside statically in Cisco IOS.

```
!
ip ips po max-events 100
ip ips name MYIPS
!
interface GigabitEthernet0/1
 ip address 10.1.1.16 255.255.255.0
 ip ips MYIPS in
 duplex full
 speed 100
 media-type rj45
 no negotiation auto
!
```

Loading the attack-drop.sdf: Example

The following example shows the basic configuration necessary to load the attack-drop.sdf file onto a router running Cisco IOS IPS. Note that the configuration is almost the same as loading the default signatures onto a router, except for the **ip ips sdf location** command, which specifies the attack-drop.sdf file.

```
!
ip ips sdf location disk2:attack-drop.sdf
ip ips name MYIPS
!
interface GigabitEthernet0/1
 ip address 10.1.1.16 255.255.255.0
 ip ips MYIPS in
 duplex full
 speed 100
 media-type rj45
 no negotiation auto
!
```

Merging the attack-drop.sdf File with the Default, Built-in Signatures: Example

The following example shows how to configure the router to load and merge the attack-drop.sdf file with the default signatures. After you have merged the two files, it is recommended that you copy the newly merged signatures to a separate file. The router can then be reloaded (via the **reload** command) or reinitalized to recognize the newly merged file (as shown the following example).

```
!
ip ips name MYIPS
!
interface GigabitEthernet0/1
 ip address 10.1.1.16 255.255.255.0
 ip ips MYIPS in
 duplex full
 speed 100
 media-type rj45
 no negotiation auto
!
!
! Merge the flash-based SDF (attack-drop.sdf) with the built-in signatures.
Router# copy disk2:attack-drop.sdf ips-sdf
! Save the newly merged signatures to a separate file.
Router# copy ips-sdf disk2:my-signatures.sdf
!
! Configure the router to use the new file, my-signatures.sdf
Router# configure terminal
Router(config)# ip ips sdf location disk2:my-signatures.sdf
! Reinitialize the IPS by removing the IPS rule set and reapplying the rule set.
Router(config-if)# interface gig 0/1
Router(config-if)# no ip ips MYIPS in
!
*Apr 8 14:05:38.243:%IPS-2-DISABLED:IPS removed from all interfaces - IPS disabled
!
Router(config-if)# ip ips MYIPS in
!
Router(config-if)# exit
```

Additional References

The following sections provide references related to Cisco IOS IPS.

Related Documents

Related Topic	Document Title
SDM IPS user's guide	*SDM Intrusion Prevention System (IPS)*
VMS IDS MC documentation	*Management Center for IDS Sensors*
IPS and firewall commands	*Cisco IOS Security Command Reference*, Release 12.4
Loading images and file systems	The section "File Management" in the *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.4
Fragment attack support via VFR	*Virtual Fragmentation Reassembly*, Cisco IOS Release 12.3(8)T feature module

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Feature Information for Configuring Cisco IOS IPS

Table 39 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Releases 12.2(1) or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for a specific command was introduced, see the command reference documentation.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Table 39 lists only the Cisco IOS software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 39 Feature Information for Configuring Cisco IOS IPS

Feature Name	Software Releases	Feature Configuration Information
Cisco IOS Intrusion Prevention System (IPS)	12.3(8)T	This feature restructured the existing Cisco IDS by allowing users to choose between loading the default, built-in signatures; downloading dynamic SDFs; or loading the SDF "attack-drop.sdf" file onto their router. Also, SDEE (an application-level communication protocol that is used to exchange IPS messages between IPS clients and IPS servers) was introduced. The following sections provide information about this feature: • Information About Cisco IOS IPS • How to Load IPS-Based Signatures onto a Router The following commands were introduced by this feature: **clear ip sdee**, **copy ips-sdf**, **debug ip ips**, **debug ip sdee**, **ip ips fail closed**, **ip ips sdf location**, **ip sdee events**, **ip sdee subscriptions**, **no ip ips sdf builtin**, **show ip sdee**
Cisco IOS Intrusion Prevention System (IPS)	12.3(14)T	Cisco IOS IPS was enhanced to support the following additional functionality: • Access to more recent virus and attack signatures via the addition of three more SMEs—STRING.TCP, STRING.ICMP, and STRING.UDP. • Intelligent and local shunning, which allows Cisco IOS IPS to shun offending traffic on the same router that Cisco IOS IPS is configured. • The **ip ips deny-action ips-interface** command, which allows users to choose between two available ACL filter settings for detecting offending packets. Support for the Post Office Protocol was deprecated and the following commands were removed from the Cisco IOS software: **ip ips po local**, **ip ips po max-events**, **ip ips po protected**, and **ip ips po remote**.

Network Admission Control

The Network Admission Control feature addresses the increased threat and impact of worms and viruses to networked businesses. This feature is part of the Cisco Self-Defending Network Initiative that helps customers identify, prevent, and adapt to security threats.

In its initial phase, the Cisco Network Admission Control functionality enables Cisco routers to enforce access privileges when an endpoint attempts to connect to a network. This access decision can be on the basis of information about the endpoint device, such as its current antivirus state. The antivirus state includes information such as version of antivirus software, virus definitions, and version of scan engine.

Network admission control systems allow noncompliant devices to be denied access, placed in a quarantined area, or given restricted access to computing resources, thus keeping insecure nodes from infecting the network.

The key component of the Cisco Network Admission Control program is the Cisco Trust Agent, which resides on an endpoint system and communicates with Cisco routers on the network. The Cisco Trust Agent collects security state information, such as what antivirus software is being used, and communicates this information to Cisco routers. The information is then relayed to a Cisco Secure Access Control Server (ACS) where access control decisions are made. The ACS directs the Cisco router to perform enforcement against the endpoint.

Feature History for Network Admission Control

Release	Modification
12.3(8)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for Network Admission Control, page 750
- Restrictions for Network Admission Control, page 750
- Information About Network Admission Control, page 750
- How to Configure Network Admission Control, page 753

- Configuration Examples for Network Admission Control, page 763
- Additional References, page 764
- Command Reference, page 765
- Glossary, page 767

Prerequisites for Network Admission Control

- You must have a Cisco IOS router that is running Cisco IOS software, Release 12.3(8)T or later.
- You must have Cisco Trust Agent installed on the endpoint devices (for example, on PCs and laptops).
- You must have a Cisco Secure Access Control Server (ACS) for AAA.
- You must be familiar with configuring access control lists (ACLs).
- You should be familiar with configuring authentication, authorization, and accounting (AAA).

Restrictions for Network Admission Control

- This feature is available only on Cisco IOS firewall feature sets.

Information About Network Admission Control

Before configuring the Network Admission Control feature, you should understand the following concepts:

- Virus Infections and Their Effect on Networks, page 750
- How Network Admission Control Works, page 751
- Network Access Device, page 751
- Cisco Trust Agent, page 751
- Cisco Secure ACS, page 752
- Remediation, page 752
- Network Admission Control and Authentication Proxy, page 753

Virus Infections and Their Effect on Networks

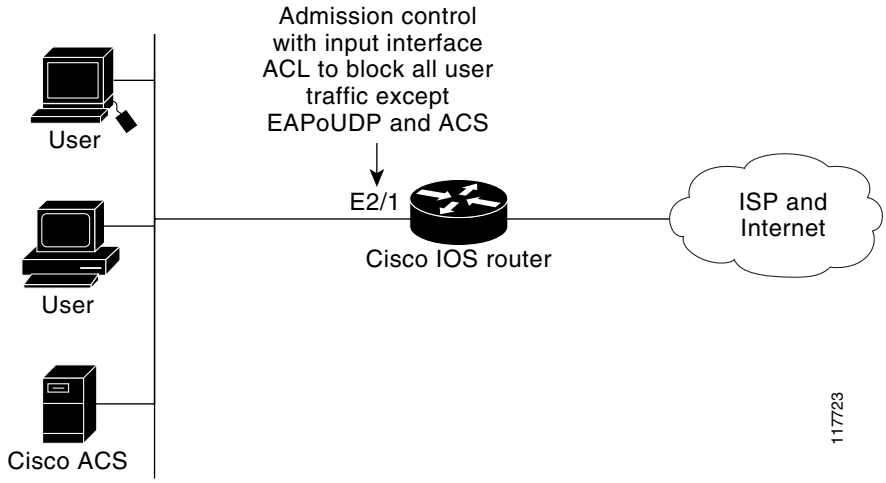
Virus infections are the single largest cause of serious security breaches for networks and often result in huge financial losses. Sources of virus infections are insecure endpoints (for example, PCs, laptops, and servers). Although the endpoints may have antivirus software installed, the software is often disabled. Even if the software is enabled, the endpoints may not have the latest virus definitions and scan engines. A larger security risk is from devices that do not have any antivirus software installed. Although antivirus vendors today are making it more difficult to disable the antivirus software, they are not addressing the risk of outdated virus definitions and scan engines.

How Network Admission Control Works

Endpoint systems, or clients, are normally hosts on the network, such as PCs, laptops, workstations, and servers. The endpoint systems are a potential source of virus infections, and their antivirus states need to be validated before they are granted network access. When an endpoint attempts an IP connection to a network through an upstream Cisco network access device (typically a Cisco IOS router), the router challenges the endpoint for its antivirus state. The endpoint systems run a client called Cisco Trust Agent, which collects antivirus state information from the end device and transports the information to the Cisco network access device. This information is then communicated to a Cisco Secure ACS where the antivirus state of the endpoint is validated and access control decisions are made and returned to Cisco network access devices. The network devices either permit, deny, or quarantine the end device. The Cisco Secure ACS may in turn use back-end antivirus vendor-specific servers for evaluating the antivirus state of the endpoint.

Figure 41 illustrates how Cisco Network Admission Control works.

Figure 41 Cisco IOS Network Admission Control System



Network Access Device

A network access device (NAD) is typically a Cisco IOS router (a Layer 3 Extensible Authentication Protocol over UDP [EAPoUDP] access point) that provides connectivity to external networks, such as the Internet or remote enterprise networks. Cisco Network Admission Control functionality may have an Intercept Access Control List (ACL), which determines connections that are intercepted for network admission. Connections from endpoints that match the access list are intercepted by Network Admission Control and are challenged for their antivirus states over a Layer 3 association before they are granted network access.

Cisco Trust Agent

Cisco Trust Agent is a specialized software that runs on endpoint systems. Cisco Trust Agent responds to challenges from the router about the antivirus state of an endpoint system. If an endpoint system is not running the Cisco Trust Agent, the network access device (router) classifies the endpoint system as

"clientless." The network access device uses the eou clientless username and eou clientless password that are configured on the network access device as the credentials of the endpoint system for validation with Cisco Secure ACS. The policy attributes that are associated with this username are enforced against the endpoint system.

Cisco Secure ACS

Cisco Secure ACS provides authentication, authorization, and accounting services for network admission control using industry-standard RADIUS authentication protocol. Cisco Secure ACS returns access control decisions to the network access device on the basis of the antivirus credentials of the endpoint system.

Using RADIUS cisco_av_pair vendor-specific attributes (VSAs), you can set the following attribute-value pairs (AV pairs) on the Cisco Secure ACS. These AV pairs will be sent to the network access device along with other access-control attributes.

- url-redirect—Enables the AAA client to intercept an HTTP request and redirect it to a new URL. This redirection is especially useful if the result of posture validation indicates that the network access control endpoint requires an update or patch that you have made available on a remediation web server. For example, a user can be redirected to a remediation web server to download and apply a new virus Directory Administration Tool (DAT) file or an operating system patch. (See the following example.)

  ```
  url-redirect=http://10.1.1.1
  ```

- posture-token—Enables Cisco Secure ACS to send a text version of a system posture token (SPT) that is derived by posture validation. The SPT is always sent in numeric format, and using the posture-token AV pair makes it easier to view the result of a posture validation request on the AAA client. (See the following example.)

  ```
  posture-token=Healthy
  ```

  Valid SPTs, in order of best to worst, are as follows:

  - Healthy
  - Checkup
  - Quarantine
  - Infected
  - Unknown

- status-query-timeout—Overrides the status-query default value of the AAA client with the value you specify, in seconds. (See the following example.)

  ```
  status-query-timeout=150
  ```

For more information about AV pairs that are supported by Cisco IOS software, see the documentation for the releases of Cisco IOS software that are implemented on your AAA clients.

Remediation

Network Admission Control supports HTTP redirection that redirects any HTTP request from the endpoint device to a specified redirect address. This support mechanism redirects all HTTP requests from a source to a specified web page (URL) to which the latest antivirus files can be downloaded. For the HTTP redirection to work, you need to set the value of the “url-redirect” VSA on the ACS and,

correspondingly, associate an access control entry in the downloadable ACL that permits the access of the endpoint system to the redirect URL address. After it has been set up, any HTTP request that matches the IP admission Intercept ACL will be redirected to the specified redirect URL address.

Network Admission Control and Authentication Proxy

It is possible that network admission control and authentication proxy can be configured for the same set of hosts on a given interface. In each case, the Intercept ACL should be the same for IP admission EAPoUDP and authentication proxy. IP admission proxy with proxy authentication should be configured first, followed by IP admission control.

How to Configure Network Admission Control

This section contains the following procedures:

- Configuring the ACL and Admission Control, page 753 (required)
- Configuring Global EAPoUDP Values, page 756 (optional)
- Configuring an Interface-Specific EAPoUDP Association, page 757 (optional)
- Configuring AAA for EAPoUDP, page 758 (optional)
- Configuring the Identity Profile and Policy, page 759 (required)
- Clearing EAPoUDP Sessions That Are Associated with an Interface, page 761 (optional)
- Verifying Network Admission Control, page 761
- Troubleshooting Network Admission Control, page 762 (optional)

Configuring the ACL and Admission Control

Network admission control is applied in the inbound direction at any interface. Applying network admission control inbound at an interface causes network admission control to intercept the initial IP connections of the intercept end system through the router.

Figure 41 shows that IP admission control is applied at the LAN interface. All network devices must be validated for their antivirus states upon their initial IP connections through the router. Until then all traffic from endpoint systems (except for EAPoUDP and Cisco Secure ACS traffic) is blocked at the interface.

The endpoint system is then challenged for its antivirus state over an EAPoUDP association. The endpoint system gains access to the network if it complies with the network admission control policy as evaluated by the Cisco Secure ACS. If the endpoint system does not comply, the device is either denied access or quarantined.

To configure an intercept ACL, perform the DETAILED STEPS below.

In this configuration, an intercept ACL is defined as “101,” and the Intercept ACL is associated with the IP admission control rule “greentree.” Any IP traffic that is destined to the 192.50.0.0 network will be subjected to validation. In addition, beginning with Step 5, an intercept ACL is applied inbound to the interface that is associated with network admission control. This ACL typically blocks access to endpoint systems until they are validated. This ACL is referred to as the default access list.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**permit** | **deny**} *protocol source destination*
4. **ip admission name** *admission-name* [**eapoudp** | **proxy** {**ftp** | **http** | **telnet**}] [**list** {*acl* | *acl-name*}]
5. **interface** *type slot/port*
6. **ip address** *ip-address mask*
7. **ip admission** *admission-name*
8. **exit**
9. **access-list** *access-list-number* [**permit** | **deny**] *protocol source destination*
10. **ip access-group** {*access-list-number* | *access-list-name*} **in**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* {**permit** | **deny**} *protocol source destination* **Example:** `Router (config)# access-list 101 permit ip any 192.50.0.0 0.0.0.255`	Defines a numbered access list.

	Command or Action	Purpose
Step 4	**ip admission name** *admission-name* [**eapoudp** | **proxy** {**ftp** | **http** | **telnet**}] [**list** {*acl* | *acl-name*}] **Example:** `Router (config)# ip admission name greentree eapoudp list 101`	Creates IP network admission control rules. The rules define how you apply admission control. The rules are as follows: • **eapoudp**—Specifies IP network admission control using EAPoUDP. • **proxy ftp**—Specifies FTP to trigger authentication proxy. • **proxy http**—Specifies HTTP to trigger authentication proxy. • **proxy telnet**—Specifies Telnet to trigger authentication proxy. You can associate the named rule with an ACL, providing control over which hosts use the admission control feature. If no standard access list is defined, the named admission rule intercepts IP traffic from all hosts whose connection-initiating packets are received at the configured interface. The list option allows you to apply a standard, extended (1 through 199) or named access list to a named admission control rule. IP connections that are initiated by hosts in the access list are intercepted by the admission control feature.
Step 5	**interface** *type slot/port* **Example:** `Router (config)# interface ethernet 2/1`	Defines an interface and enters interface configuration mode.
Step 6	**ip address** *ip-address mask* **Example:** `Router (config-if)# ip address 192.0.0.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 7	**ip admission** *admission-name* **Example:** `Router (config-if)# ip admission greentree`	Applies the named admission control rule at the interface.
Step 8	**exit** **Example:** `Router (config-if)# exit`	Exits interface configuration mode.

	Command or Action	Purpose
Step 9	**acess-list** *access-list-number* {**permit** | **deny**} *protocol source destination* **Example:** `Router (config)# access-list 105 permit udp any any` or `Router (config)# access-list 105 permit ip host 192.168.0.2 any` or `Router (config)# access-list 105 deny ip any any`	Defines a numbered access list. **Note** In the first two examples (under "Command or Action"), ACL "105" denies all IP traffic except UDP and access to 192.168.0.2 (Cisco Secure ACS). **Note** In the third example (under "Command or Action", ACL "105" will be applied on the interface that is configured for network admission control, and access to endpoint systems (except for EAPoUDP traffic and access to Cisco Secure ACS [192.168.0.2 in the example] is blocked until their antivirus states are validated. This ACL ("105") is referred to as "Interface ACL."
Step 10	**ip access-group** {*access-list-number* | *access-list-name*} **in** **Example:** `Router (config)# ip access-group 105 in`	Controls access to an interface.

Configuring Global EAPoUDP Values

To configure global EAPoUDP values, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **eou** {**allow** | **clientless** | **default** | **initialize** | **logging** | **max-retry** | **port** | **rate-limit** | **revalidate** | **timeout**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**eou** {**allow** | **clientless** | **default** | **initialize** | **logging** | **max-retry** | **port** | **rate-limit** | **revalidate** | **timeout**} **Example:** `Router (config)# eou initialize allow`	Specifies EAPoUDP values. • For a breakout of available keywords and arguments for the **eou** command, see the following commands: – **eou allow** – **eou clientless** – **eou default** – **eou initialize** – **eou logging** – **eou max-retry** – **eou port** – **eou rate-limit** – **eou revalidate** – **eou timeout**

Configuring an Interface-Specific EAPoUDP Association

To configure an EAPoUDP association that can be changed or customized for a specific interface that is associated with network admission control, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/port*
4. **eou** [**default** | **max-retry** | **revalidate** | **timeout**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type slot/port* **Example:** `Router (config)# interface ethernet 2/1`	Defines an interface and enters interface configuration mode.
Step 4	**eou** [**default** | **max-retry** | **revalidate** | **timeout**] **Example:** `Router (config-if)# eou revalidate`	Enables an EAPoUDP association for a specific interface. • For a breakout of available keywords and arguments for the **eou** command, see the following commands: – **eou default** – **eou max-retry** – **eou revalidate** – **eou timeout**

Configuring AAA for EAPoUDP

To set up AAA for EAPoUDP, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication eou default enable group radius**
5. **radius-server host** {*hostname* | *ip-address*}
6. **radius-server key** {**0** *string* | **7** *string* | *string*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router (config)# aaa new-model`	Enables the AAA access control model.

	Command or Action	Purpose
Step 4	**aaa authentication eou default enable group radius** **Example:** `Router (config)# aaa authentication eou default enable group radius`	Sets authentication lists for an EAPoUDP association.
Step 5	**radius-server host** {*hostname* | *ip-address*} **Example:** `Router (config)# radius-server host 192.0.0.40`	Specifies a RADIUS server host.
Step 6	**radius-server key** {**0** *string* | **7** *string* | *string*} **Example:** `Router (config)# radius-server key cisco`	Sets the authentication and encryption key for all RADIUS communications between the router and the RADIUS daemon.

Configuring the Identity Profile and Policy

Identity is a common infrastructure that is used to specify local profile and policy configurations. The identity profile allows you to statically authorize or validate individual devices on the basis of IP address, MAC address, or device type. Each statically authenticated device can be associated with a local policy that specifies the network access control attributes. Hosts are added to this "exception list" using the **identity profile** command, and corresponding policies are associated with these hosts using the **identity policy** command.

If the client is part of the identity (that is, the client is on the exception list), the status of the client is set on the basis of the identity configuration. The client does not have to go through the posture validation process, and the associated identity policy is applied for the client.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **identity profile eapoudp**
4. **device** {**authorize** {**ip address** *ip-address* {**policy** *policy-name*} | **mac-address** *mac-address* | **type** {**cisco** | **ip** | **phone**}} | **not-authorize**}
5. **exit**
6. **identity policy** *policy-name* [**access-group** *group-name* | **description** *line-of-description* | **redirect** *url* | **template** [**virtual-template** *interface-name*]]
7. **access-group** *group-name*
8. **exit**
9. **exit**
10. **ip access-list extended** *access-list-name*
11. **permit | deny** *source source-wildcard destination destination wildcard*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**identity profile eapoudp** **Example:** `Router (config)# identity profile eapoudp`	Creates an identity profile and enters identity profile configuration mode.
Step 4	**device** {**authorize** {**ip address** *ip-address* {**policy** *policy-name*} | **mac-address** *mac-address* | **type** {**cisco** | **ip** | **phone**}} | **not-authorize**} **Example:** `Router (config-identity-prof)# device authorize ip address 10.10.142.25 policy bluemoon`	Statically authorizes an IP device and applies an associated policy to the device.
Step 5	**exit** **Example:** `Router (config-identity-prof)# exit`	Exits identity profile configuration mode.
Step 6	**identity policy** *policy-name* [**access-group** *group-name* | **description** *line-of-description* | **redirect** *url* | **template** [**virtual-template** *interface-name*]] **Example:** `Router (config-identity-prof)# identity policy bluemoon`	Creates an identity policy and enters identity policy configuration mode.
Step 7	**access-group** *group-name* **Example:** `Router (config-identity-policy)# access-group exempt-acl`	Defines network access attributes for the identity policy.
Step 8	**exit** **Example:** `Router (config-identity-policy)# exit`	Exits identity policy configuration mode.
Step 9	**exit** **Example:** `Router (config-identity-prof)# exit`	Exits identity profile configuration mode.

	Command or Action	Purpose
Step 10	**ip access-list extended** *access-list-name* **Example:** `Router (config)# ip access-list extended exempt-acl`	Defines access control for statically authenticated devices (and enters network access control configuration mode).
Step 11	**permit** | **deny** *source source-wildcard destination destination wildcard* **Example:** `Router (config-ext-nacl)# permit ip any 192.50.0.0. 0.0.0.255`	Set conditions to allow a packet to pass a named IP access list.

Clearing EAPoUDP Sessions That Are Associated with an Interface

To clear EAPoUDP sessions that are associated with a particular interface or that are on the NAD, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **clear eou all**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear eou all** **Example:** `Router# clear eou all`	Clears all EAPoUDP sessions on the NAD.

Verifying Network Admission Control

To verify EAP and EAPoUDP messages or sessions, perform the following steps. The **show** commands may be used in any order or independent of the other **show** command.

SUMMARY STEPS

1. **enable**
2. **show eou all**
3. **show ip admission eapoudp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show eou all** **Example:** `Router# show eou all`	Displays information about EAPoUDP sessions on the network access device.
Step 3	**show ip admission eapoudp** **Example:** `Router# show ip admission eapoudp`	Displays the network admission control configuration or network admission cache entries.

Troubleshooting Network Admission Control

The following commands may be used to display information about EAP and EAPoUDP messages or sessions. The **debug** commands may be used in any order or independent of the other **debug** commands.

SUMMARY STEPS

1. **enable**
2. **debug eap** {**all** | **errors** | **packets** | **sm**}
3. **debug eou** {**all** | **eap** | **errors** | **packets** | **sm**}
4. **debug ip admission eapoudp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug eap** {**all**	**errors**	**packets**	**sm**} **Example:** `Router# debug eap all`	Displays information about EAP messages.

	Command or Action	Purpose
Step 3	**debug eou** {**all** | **eap** | **errors** | **packets** | **sm**} **Example:** `Router# debug eou all`	Displays information about EAPoUDP messages.
Step 4	**debug ip admission eapoudp** **Example:** `Router# debug ip admission eapoudp`	Displays information about IP admission events.

Configuration Examples for Network Admission Control

This section includes the following example.

- Network Admission Control: Example, page 763

Network Admission Control: Example

The following output example shows that IP admission control has been configured on a Cisco IOS router:

```
Router# show running-config

Building configuration...

Current configuration: 1240 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
aaa new-model
!
!
aaa authentication eou default group radius
aaa session-id common
ip subnet-zero
ip cef
!
! The following line creates a network admission rule. A list is not specified; therefore,
! the rule intercepts all traffic on the applied interface.
ip admission name avrule eapoudp
!
eou logging
!
!
```

```
interface FastEthernet0/0
 ip address 10.13.11.106 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 10.0.0.1 255.255.255.0
 ip access-group 102 in
! The following line configures an IP admission control interface.
 ip admission avrule
 duplex auto
 speed auto
!
ip http server
no ip http secure-server
ip classless
!
!
! The following lines configure an interface access list that allows EAPoUDP traffic
! and blocks the rest of the traffic until it is validated.
access-list 102 permit udp any any eq 21862
access-list 102 deny   ip any any
!
!
! The following line configures RADIUS.
radius-server host 1013.11.105 auth-port 1645 acct-port 1646 key cisco
!
control-plane
!
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
!
!
end
```

Additional References

The following sections provide references related to Network Admission Control.

Related Documents

Related Topic	Document Title
Configuring ACLs	"Access Control Lists: Overview and Guidelines" chapter of the "Traffic Filtering and Firewalls" section of the *Cisco IOS Security Configuration Guide*, Release 12.3.
Authentication, authorization, and accounting	"Authentication, Authorization, and Accounting" section of *Cisco IOS Security Configuration Guide*, Release 12.3.
Interfaces, configuring	*Cisco IOS Interface and Hardware Component Configuration Guide*, Release 12.3.

Standards

Standards	Title
No new or modified standards are supported by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

New Commands

- **aaa authentication eou default enable group radius**
- **access-group (identity policy)**
- **auth-type**
- **clear eou**

- clear ip admission cache
- debug eap
- debug eou
- debug ip admission eapoudp
- description (identity policy)
- eou allow
- eou clientless
- eou default
- eou initialize
- eou logging
- eou max-retry
- eou port
- eou rate-limit
- eou revalidate
- eou timeout
- identity policy
- identity profile eapoudp
- ip admission
- ip admission name
- redirect (identity policy)
- show eou
- show ip admission
- template (identity policy)

Modified Commands

- description (identity profile)
- device (identity profile)

Glossary

default access policy—Set of ACLs that are applied to a client device until its credentials are validated by the AAA server.

EAPoUDP—Extensible Authentication Protocol over User Datagram Protocol. EAP is a framework that supports multiple, optional authentication mechanisms for PPP, including clear-text passwords, challenge-response, and arbitrary dialogue sequences. UDP is a connectionless transport layer protocol in the TCP/IP protocol stack. UDP is a simple protocol that exchanges datagrams without acknowledgments or guaranteed delivery, and it requires that error processing and retransmission be handled by other protocols. UDP is defined in RFC 768.

ip admission rule—A named rule that defines how IP admission control is applied. The IP admission rule is associated with an Intercept ACL and provides control over which hosts can use the IP admission feature. To create an IP admission control rule, use the **ip admission name** command.

posture token—Status that is used to convey the result of the evaluation of posture credentials. The AAA server maps the posture token (its status can be Healthy, Checkup, Quarantine, Infected, or Unknown) to a network access policy (ACL, URL, redirect, or status query timer) for the peer that the client wants to reach.

Note Refer to *Internetworking Terms and Acronyms* for terms not included in this glossary.

Authentication Proxy

This part consists of the following:

- Configuring Authentication Proxy
- Firewall Support of HTTPS Authentication Proxy
- Firewall Authentication Proxy for FTP and Telnet Sessions

Configuring Authentication Proxy

This chapter describes the Cisco IOS Firewall Authentication Proxy feature. Authentication proxy provides dynamic, per-user authentication and authorization, authenticating users against industry standard TACACS+ and RADIUS authentication protocols. Authenticating and authorizing connections by users provides more robust protection against network attacks.

For a complete description of the authentication proxy commands in this chapter, refer to the "Authentication Proxy Commands" chapter of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Finding Additional Feature Support Information" section on page cxvii in the chapter Using Cisco IOS Software for Release 12.4.

In This Chapter

This chapter contains the following sections:

- About Authentication Proxy
- Authentication Proxy Configuration Task List
- Monitoring and Maintaining the Authentication Proxy
- Authentication Proxy Configuration Examples

About Authentication Proxy

The Cisco IOS Firewall authentication proxy feature allows network administrators to apply specific security policies on a per-user basis. Previously, user identity and related authorized access were associated with a user IP address, or a single security policy had to be applied to an entire user group or subnetwork. Now, users can be identified and authorized on the basis of their per-user policy. Tailoring of access privileges on an individual basis is possible, as opposed to applying a general policy across multiple users.

With the authentication proxy feature, users can log in to the network or access the Internet via HTTP, and their specific access profiles are automatically retrieved and applied from a CiscoSecure ACS, or other RADIUS, or TACACS+ authentication server. The user profiles are active only when there is active traffic from the authenticated users.

The authentication proxy is compatible with other Cisco IOS security features such as Network Address Translation (NAT), Context-based Access Control (CBAC), IP Security (IPSec) encryption, and Cisco Secure VPN Client (VPN client) software.

This section contains the following sections:

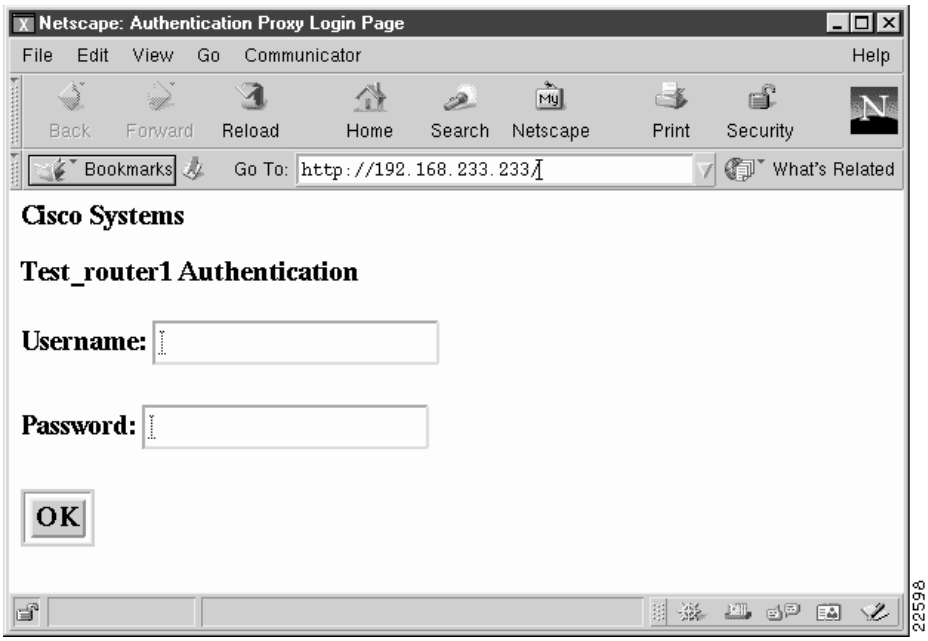
- How the Authentication Proxy Works
- Secure Authentication
- Using the Authentication Proxy
- When to Use the Authentication Proxy
- Applying the Authentication Proxy
- Operation with One-Time Passwords
- Compatibility with Other Security Features
- Compatibility with AAA Accounting
- Protection Against Denial-of-Service Attacks
- Risk of Spoofing with Authentication Proxy
- Comparison with the Lock-and-Key Feature
- Restrictions
- Prerequisites to Configuring Authentication Proxy

How the Authentication Proxy Works

When a user initiates an HTTP session through the firewall, the authentication proxy is triggered. The authentication proxy first checks to see if the user has been authenticated. If a valid authentication entry exists for the user, the connection is completed with no further intervention by the authentication proxy. If no entry exists, the authentication proxy responds to the HTTP connection request by prompting the user for a username and password.

Figure 42 illustrates the authentication proxy HTML login page.

Figure 42 Authentication Proxy Login Page



Users must successfully authenticate themselves with the authentication server by entering a valid username and password.

If the authentication succeeds, the user's authorization profile is retrieved from the AAA server. The authentication proxy uses the information in this profile to create dynamic access control entries (ACEs) and add them to the inbound (input) access control list (ACL) of an input interface and to the outbound (output) ACL of an output interface, if an output ACL exists at the interface. This process enables the firewall to allow authenticated users access to the network as permitted by the authorization profile. For example, a user can initiate a Telnet connection through the firewall if Telnet is permitted in the user's profile.

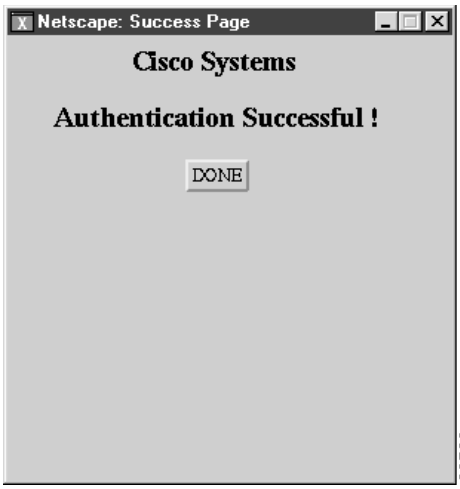
If the authentication fails, the authentication proxy reports the failure to the user and prompts the user with multiple retries. If the user fails to authenticate after five attempts, the user must wait two minutes and initiate another HTTP session to trigger authentication proxy.

The login page is refreshed each time the user makes requests to access information from a web server.

The authentication proxy customizes each of the access list entries in the user profile by replacing the source IP addresses in the downloaded access list with the source IP address of the authenticated host.

At the same time that dynamic ACEs are added to the interface configuration, the authentication proxy sends a message to the user confirming that the login was successful. Figure 43 illustrates the login status in the HTML page.

Figure 43 Authentication Proxy Login Status Message



The authentication proxy sets up an inactivity (idle) timer for each user profile. As long as there is activity through the firewall, new traffic initiated from the user's host does not trigger the authentication proxy, and authorized user traffic is permitted access through the firewall.

If the idle timer expires, the authentication proxy removes the user's profile information and dynamic access lists entries. When this happens, traffic from the client host is blocked. The user must initiate another HTTP connection to trigger the authentication proxy.

Secure Authentication

The authentication proxy uses JavaScript to help achieve secure authentication using the client browser. Secure authentication prevents a client from mistakenly submitting a username and password to a network web server other than the authentication proxy router.

This section contains the following sections:

- Operation with JavaScript
- Operation Without JavaScript

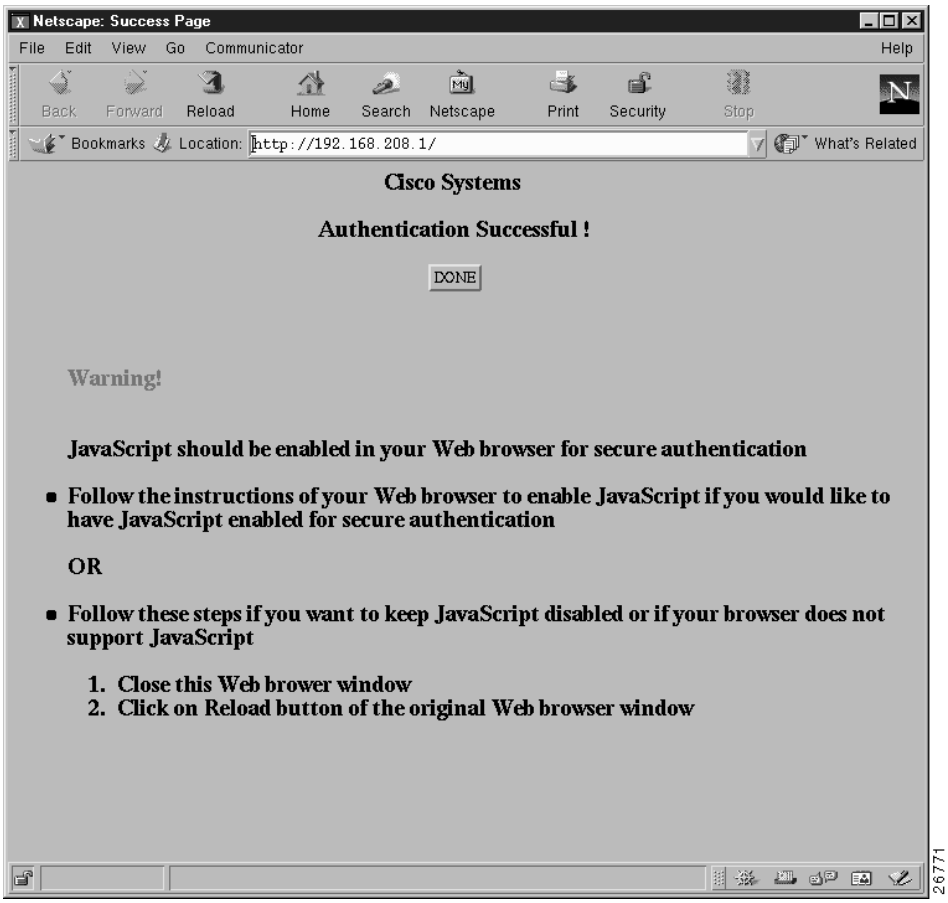
Operation with JavaScript

Users should enable JavaScript on the browser prior to initiating an HTTP connection. With JavaScript enabled on the browser, secure authentication is done automatically, and the user sees the authentication message shown in Figure 43. The HTTP connection is completed automatically for the user.

Operation Without JavaScript

If the client browser does not support JavaScript, or if site security policy prevents users from enabling JavaScript, any login attempt generates a popup window with instructions for manually completing the connection. Figure 44 illustrates the authentication proxy login status message with JavaScript disabled on the browser.

Figure 44 Authentication Proxy Login Status Message with JavaScript Disabled



To close this window, click Close on the browser File menu.

After closing the popup window, the user should click Reload (Refresh for Internet Explorer) in the browser window in which the authentication login page is displayed. If the user's last authentication attempt succeeds, clicking Reload brings up the web page the user is trying to retrieve. If the user's last attempt fails, clicking Reload causes the authentication proxy to intercept the client HTTP traffic again, prompting the user with another login page that solicits the username and password.

If JavaScript is not enabled, it is strongly recommended that site administrators advise users of the correct procedure for closing the popup window as described in the section "Establishing User Connections Without JavaScript."

Using the Authentication Proxy

Unlike some Cisco IOS Firewall features that operate transparently to the user, the authentication proxy feature requires some user interaction on the client host. Table 40 describes the interaction of the authentication proxy with the client host.

Table 40 Authentication Proxy Interaction with the Client Host

Authentication Proxy Action with Client	Description
Triggering on HTTP connections	If a user is not currently authenticated at the firewall router, any HTTP connection initiated by the user triggers the authentication proxy. If the user is already authenticated, the authentication proxy is transparent to the user.
Logging in using the login page	Triggering the authentication proxy generates an HTML-based login page.The user must enter a username and password to be authenticated with the AAA server. Figure 42 illustrates the authentication proxy login page.
Authenticating the user at the client	Following the login attempt, the authentication proxy action can vary depending on whether JavaScript is enabled in the browser. If JavaScript is enabled, and authentication is successful, the authentication proxy displays a message indicating the status of the authentication as shown in Figure 43. After the authentication status is displayed, the proxy automatically completes the HTTP connection. If JavaScript is disabled, and authentication is successful, the authentication proxy generates a popup window with additional instructions for completing the connection. See Figure 44. If authentication is unsuccessful in any case, the user must log in again from the login page.

When to Use the Authentication Proxy

Here are examples of situations in which you might use the authentication proxy:

- You want to manage access privileges on an individual (per-user) basis using the services provided by the authentication servers instead of configuring access control based on host IP address or global access policies. Authenticating and authorizing users from any host IP address also allows network administrators to configure host IP addresses using DHCP.
- You want to authenticate and authorize local users before permitting access to intranet or Internet services or hosts through the firewall.
- You want to authenticate and authorize remote users before permitting access to local services or hosts through the firewall.
- You want to control access for specific extranet users. For example, you might want to authenticate and authorize the financial officer of a corporate partner with one set of access privileges while authorizing the technology officer for that same partner to use another set of access privileges.
- You want to use the authentication proxy in conjunction with VPN client software to validate users and to assign specific access privileges.
- You want to use the authentication proxy in conjunction with AAA accounting to generate “start” and “stop” accounting records that can be used for billing, security, or resource allocation purposes, thereby allowing users to track traffic from the authenticated hosts.

Applying the Authentication Proxy

Apply the authentication proxy in the inbound direction at any interface on the router where you want per-user authentication and authorization. Applying the authentication proxy inbound at an interface causes it to intercept a user's initial connection request before that request is subjected to any other processing by the firewall. If the user fails to gain authentication with the AAA server, the connection request is dropped.

How you apply the authentication proxy depends on your security policy. For example, you can block all traffic through an interface and enable the authentication proxy feature to require authentication and authorization for all user initiated HTTP connections. Users are authorized for services only after successful authentication with the AAA server.

The authentication proxy feature also allows you to use standard access lists to specify a host or group of hosts whose initial HTTP traffic triggers the proxy.

Figure 45 shows the authentication proxy applied at the LAN interface with all network users required to be authenticated upon the initial connection (all traffic is blocked at each interface).

Figure 45 Applying the Authentication Proxy at the Local Interface

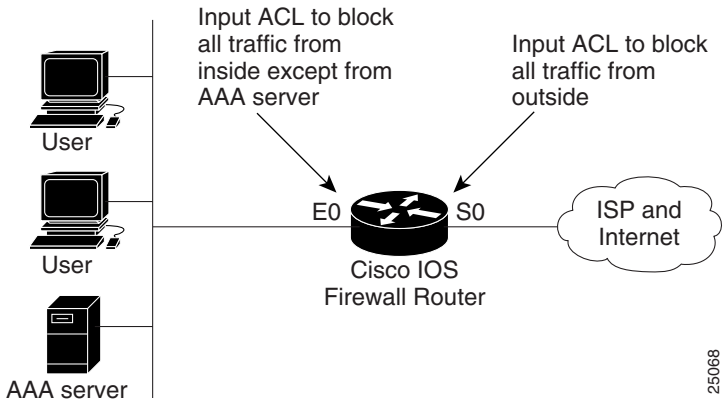
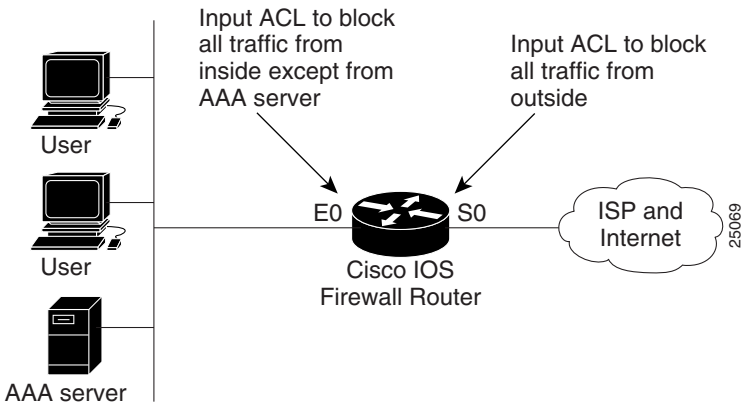


Figure 46 shows the authentication proxy applied at the dial-in interface with all network traffic blocked at each interface.

Figure 46 Applying the Authentication Proxy at an Outside Interface



Operation with One-Time Passwords

Given a one-time password, the user enters the username and one-time password in the HTML login page as usual.

The user must enter the correct token password within the first three attempts. After three incorrect entries, the user must enter two valid token passwords in succession before authentication is granted by the AAA server.

Compatibility with Other Security Features

The authentication proxy is compatible with Cisco IOS software and with Cisco IOS security features:

- Cisco IOS Firewall Intrusion Detection System (IDS)
- NAT
- CBAC
- IPSec encryption
- VPN client software

The authentication proxy works transparently with the Cisco IOS Firewall IDS and IPSec encryption features. The following sections describe the relationship of the NAT, CBAC, and VPN client software features with the authentication proxy:

- NAT Compatibility
- CBAC Compatibility
- VPN Client Compatibility

NAT Compatibility

The authentication proxy feature is compatible with NAT only if the ACL and authentication are completed prior to the NAT translation. Although NAT is compatible with the authentication proxy feature, NAT is not a requirement of the feature.

CBAC Compatibility

Although authentication proxy is compatible with CBAC security functions, CBAC is not required to use the authentication proxy feature.

Authentication proxy's authorization returns Access Control Entries (ACEs) that are dynamically prepended into a manually created ACL. Thereafter, apply the ACL to the "protected side" inbound interface, allowing or disallowing an authorized user's source IP address access to the remote networks.

VPN Client Compatibility

Using the authentication proxy, network administrators can apply an extra layer of security and access control for VPN client traffic. If a VPN client initiates an HTTP connection, the authentication proxy first checks for prior client authentication. If the client is authenticated, authorized traffic is permitted. If the client is not authenticated, the HTTP request triggers the authentication proxy, and the user is prompted for a username and password.

If the user authentication is successful, the authentication proxy retrieves the user profile from the AAA server. The source address in the user profile entries is replaced with the IP address of the authenticated VPN client from the decrypted packet.

Compatibility with AAA Accounting

Using the authentication proxy, you can generate "start" and "stop" accounting records with enough information to be used for billing and security auditing purposes. Thus, you can monitor the actions of authenticated hosts that use the authentication proxy service.

When an authentication proxy cache and associated dynamic access control lists are created, the authentication proxy will start to track the traffic from the authenticated host. Accounting saves data about this event in a data structure stored with the data of other users. If the accounting start option is enabled, you can generate an accounting record (a "start" record) at this time. Subsequent traffic from the authenticated host will be recorded when the dynamic ACL created by the authentication proxy receives the packets.

When an authentication proxy cache expires and is deleted, additional data, such as elapsed time, is added to the accounting information and a "stop" record is sent to the server. At this point, the information is deleted from the data structure.

The accounting records for the authentication proxy user session are related to the cache and the dynamic ACL usage.

Note The accounting records must include RADIUS attributes 42, 46, and 47 for both RADIUS and TACACS+.

For more information on RADIUS attributes, refer to the appendix "RADIUS Attributes."

Protection Against Denial-of-Service Attacks

The authentication proxy monitors the level of incoming HTTP requests. For each request, the authentication proxy prompts the user's for login credentials. A high number of open requests could indicate that the router is the subject of a denial-of-service (DoS) attack. The authentication proxy limits the level of open requests and drops additional requests until the number of open requests has fallen below 40.

If the firewall is experiencing a high level of connection requests requiring authentication, legitimate network users may experience delays when making connections, or the connection may be rejected and the user must try the connection again.

Risk of Spoofing with Authentication Proxy

When the authentication proxy is triggered, it creates a dynamic opening in the firewall by temporarily reconfiguring an interface with user access privileges. While this opening exists, another host might spoof the authenticated users address to gain access behind the firewall. The authentication proxy does not cause the address spoofing problem; the problem is only identified here as a matter of concern to the user. Spoofing is a problem inherent to all access lists, and the authentication proxy does not specifically address this problem.

Comparison with the Lock-and-Key Feature

Lock-and-key is another Cisco IOS Firewall feature that uses authentication and dynamic access list to provide user access through the firewall. Table 41 compares the authentication proxy and lock-and-key features.

Table 41 Comparison of the Authentication Proxy and Lock-and-Key Features

Lock-and-Key	Authentication Proxy
Triggers on Telnet connection requests.	Triggers on HTTP connection requests.
TACACS+, RADIUS, or local authentication.	TACACS+ or RADIUS authentication and authorization.
Access lists are configured on the router only.	Access lists are retrieved from the AAA server only.
Access privileges are granted on the basis of the user's host IP address.	Access privileges are granted on a per-user and host IP address basis.
Access lists are limited to one entry for each host IP address.	Access lists can have multiple entries as defined by the user profiles on the AAA server.
Associates a fixed IP addresses with a specific user. Users must log in from the host with that IP address.	Allows DHCP-based host IP addresses, meaning that users can log in from any host location and obtain authentication and authorization.

Use the authentication proxy in any network environment that provides a per-user security policy. Use lock-and-key in network environments that might benefit from local authentication and a limited number of router-based access control policies based on host addresses. Use lock-and-key in environments not using the Cisco Secure Integrated Software.

Restrictions

- The authentication proxy triggers only on HTTP connections.
- HTTP services must be running on the standard (well-known) port, which is port 80 for HTTP.
- Client browsers must enable JavaScript for secure authentication.
- The authentication proxy access lists apply to traffic passing through the router. Traffic destined to the router is authenticated by the existing authentication methods provided by Cisco IOS software.
- The authentication proxy does not support concurrent usage; that is, if two users try to log in from the same host at the same time, authentication and authorization applies only to the user who first submits a valid username and password.
- Load balancing using multiple or different AAA servers is not supported.

Prerequisites to Configuring Authentication Proxy

Prior to configuring authentication proxy, review the following:

- For the authentication proxy to work properly, the client host must be running the following browser software:
  - Microsoft Internet Explorer 3.0 or later
  - Netscape Navigator 3.0 or later
- The authentication proxy has an option to use standard access lists. You must have a solid understanding of how access lists are used to filter traffic before you attempt to configure the authentication proxy. For an overview of how to use access lists with the Cisco IOS Firewall, refer to the chapter "Access Control Lists: Overview and Guidelines."
- The authentication proxy employs user authentication and authorization as implemented in the Cisco authentication, authorization, and accounting (AAA) paradigm. You must understand how to configure AAA user authentication, authorization, and accounting before you configure the authentication proxy. User authentication, authorization, and accounting are explained in the chapter "Authentication, Authorization, and Accounting (AAA)."
- To run the authentication proxy successfully with Cisco IOS Firewall, configure CBAC on the firewall. For complete information on the CBAC feature, refer to the chapter "Configuring Context-Based Access Control."

Authentication Proxy Configuration Task List

To configure the authentication proxy feature, perform the following tasks:

- Configuring AAA (Required)
- Configuring the HTTP Server (Required)
- Configuring the Authentication Proxy (Required)
- Verifying the Authentication Proxy (Optional)

For authentication proxy configuration examples using the commands in this chapter, refer to the section "Authentication Proxy Configuration Examples" at the end of this chapter.

Configuring AAA

You must configure the authentication proxy for AAA services. Use the following commands in global configuration mode to enable authorization and to define the authorization methods:

	Command	Purpose
Step 1	`router(config)# aaa new-model`	Enables the AAA functionality on the router.
Step 2	`router(config)# aaa authentication login default` *TACACS+ RADIUS*	Defines the list of authentication methods at login.
Step 3	`router(config)# aaa authorization auth-proxy default` [*method1* [*method2*...]]	Uses the **auth-proxy** keyword to enable authentication proxy for AAA methods.
Step 4	`router(config)# aaa accounting auth-proxy default start-stop group tacacs+`	Uses the **auth-proxy** keyword to set up the authorization policy as dynamic ACLs that can be downloaded. This command activates authentication proxy accounting.
Step 5	`router(config)# tacacs-server host` *hostname*	Specifies an AAA server. For RADIUS servers, use the **radius server host** command.
Step 6	`router(config)# tacacs-server key` *key*	Sets the authentication and encryption key for communications between the router and the AAA server. For RADIUS servers use the **radius server key** command.
Step 7	`router(config)# access-list` *access-list-number* `permit tcp host` *source* `eq` *tacacs* `host` *destination*	Creates an ACL entry to allow the AAA server to return traffic to the firewall. The source address is the IP address of the AAA server, and the destination is the IP address of the router interface where the AAA server resides.

In addition to configuring AAA on the firewall router, the authentication proxy requires a per-user access profile configuration on the AAA server. To support the authentication proxy, configure the AAA authorization service **auth-proxy** on the AAA server as outlined here:

- Define a separate section of authorization for the **auth-proxy** keyword to specify the downloadable user profiles. This keyword does not interfere with other type of services, such as EXEC. The following example shows a user profile on a TACACS server:

```
default authorization = permit
key = cisco
user = newuser1 {
login = cleartext cisco
service = auth-proxy
{
priv-lvl=15
proxyacl#1="permit tcp any any eq 26"
proxyacl#2="permit icmp any host 60.0.0.2″
proxyacl#3="permit tcp any any eq ftp"
proxyacl#4="permit tcp any any eq ftp-data"
proxyacl#5="permit tcp any any eq smtp"
proxyacl#6="permit tcp any any eq telnet"
}
}
```

- The only supported attribute in the AAA server user configuration is proxyacl#n. Use the proxyacl#n attribute when configuring the access lists in the profile. The attribute proxyacl#n is for both RADIUS and TACACS+ attribute-value (AV) pairs.
- The privilege level must be set to 15 for all users.

- The access lists in the user profile on the AAA server must have access commands that contain only the **permit** keyword.
- Set the source address to the **any** keyword in each of the user profile access list entries. The source address in the access lists is replaced with the source address of the host making the authentication proxy request when the user profile is downloaded to the firewall.
- The supported AAA servers are:
  - CiscoSecure ACS 2.1.x for Windows NT
  - CiscoSecure ACS 2.3 for Windows NT
  - CiscoSecure ACS 2.2.4 for UNIX
  - CiscoSecure ACS 2.3 for UNIX
  - TACACS+ server (vF4.02.alpha)
  - Ascend RADIUS server radius-980618 (required attribute-value pair patch)
  - Livingston RADIUS server (v1.16)

Refer to the section "AAA Server User Profile Example" for sample AAA server configurations.

Configuring the HTTP Server

To use authentication proxy, you must also enable the HTTP server on the firewall and set the HTTP server authentication method to use AAA. Enter the following commands in global configuration mode:

	Command	Purpose
Step 1	`router(config)# ip http server`	Enables the HTTP server on the router. The authentication proxy uses the HTTP server to communicate with the client for user authentication.
Step 2	`router(config)# ip http authentication aaa`	Sets the HTTP server authentication method to AAA.
Step 3	`router(config)# ip http access-class` *access-list-number*	Specifies the access list for the HTTP server. Use the standard access list number configured in the section "Interface Configuration Example."

Configuring the Authentication Proxy

Note Set the **auth-cache-time** option for any authentication proxy rule to a higher value than the idle timeout value for any CBAC inspection rule. When the authentication proxy removes an authentication cache along with its associated dynamic user ACL, there may be some idle connections monitored by CBAC, and removal of user-specific ACLs could cause those idle connections to hang. If CBAC has a shorter idle timeout, CBAC resets these connections when the idle timeout expires; that is, before the authentication proxy removes the user profile.

To configure the authentication proxy, use the following commands, beginning in global configuration mode:

	Command	Purpose
Step 1	`router(config)# ip auth-proxy auth-cache-time min`	Sets the global authentication proxy idle timeout value in minutes. If the timeout expires, user authentication entries are removed, along with any associated dynamic access lists. The default value is 60 minutes.
Step 2	`router(config)# ip auth-proxy auth-proxy-banner`	(Optional) Displays the name of the firewall router in the authentication proxy login page. The banner is disabled by default.
Step 3	`router(config)# ip auth-proxy name auth-proxy-name http [auth-cache-time min] [list {acl |acl-name}]`	Creates authentication proxy rules. The rules define how you apply authentication proxy. This command associates connections initiating HTTP protocol traffic with an authentication proxy name. You can associate the named rule with an access control list (ACL), providing control over which hosts use the authentication proxy feature. If no standard access list is defined, the named authentication proxy rule intercepts HTTP traffic from all hosts whose connection initiating packets are received at the configured interface. (Optional) The **auth-cache-time** option overrides the global authentication proxy cache timer. This option provides more control over timeout values for a specific authentication proxy rule. If no value is specified, the proxy rule assumes the value set with the **ip auth-proxy auth-cache-time** command. (Optional) The **list** option allows you to apply a standard, extended (1-199), or named access list to a named authentication proxy rule. HTTP connections initiated by hosts in the access list are intercepted by the authentication proxy.
Step 4	`router(config)# interface type`	Enters interface configuration mode by specifying the interface type on which to apply the authentication proxy.
Step 5	`router(config-if)# ip auth-proxy auth-proxy-name`	In interface configuration mode, applies the named authentication proxy rule at the interface. This command enables the authentication proxy rule with that name.

Verifying the Authentication Proxy

Verifying the authentication proxy configuration can have several components:

- Checking the Authentication Proxy Configuration (Optional)
- Establishing User Connections with JavaScript (Optional)
- Establishing User Connections Without JavaScript (Optional)

Checking the Authentication Proxy Configuration

To check the current authentication proxy configuration, use the **show ip auth-proxy configuration** command in privileged EXEC mode:

Command	Purpose
`router# show ip auth-proxy configuration`	Displays the authentication proxy configuration.

In the following example, the global authentication proxy idle timeout value is set to 60 minutes, the named authentication proxy rule is "pxy", and the idle timeout value for this named rule is one minute. The display shows that no host list is specified, meaning that all connections initiating HTTP traffic at the interface are subject to the authentication proxy rule.

```
router# show ip auth-proxy configuration
Authentication cache time is 60 minutes
Authentication Proxy Rule Configuration
Auth-proxy name pxy
http list not specified auth-cache-time 1 minutes
```

To verify that the authentication proxy is successfully configured on the router, ask a user to initiate an HTTP connection through the router. The user must have authentication and authorization configured at the AAA server. If the user authentication is successful, the firewall completes the HTTP connection for the user. If the authentication is unsuccessful, check the access list and the AAA server configurations.

Display the user authentication entries using the **show ip auth-proxy cache** command in privileged EXEC mode:

Command	Purpose
`router# show ip auth-proxy cache`	Displays the list of user authentication entries.

The authentication proxy cache lists the host IP address, the source port number, the timeout value for the authentication proxy, and the state of the connection. If the authentication proxy state is HTTP_ESTAB, the user authentication was successful.

```
router# show ip auth-proxy cache
Authentication Proxy Cache
 Client IP 192.168.25.215 Port 57882, timeout 1, state HTTP_ESTAB
```

Wait for one minute, which is the timeout value for this named rule, and ask the user to try the connection again. After one minute, the user connection is denied because the authentication proxy has removed the user's authentication entry and any associated dynamic ACLs. The user is presented with a new authentication login page and must log in again to gain access through the firewall.

Establishing User Connections with JavaScript

To verify client connections using the authentication proxy with JavaScript enabled on the client browser, follow this procedure:

Step 1 From a client host, initiate an HTTP connection through the firewall. This generates the authentication proxy login page.

Step 2 At the authentication proxy login page, enter a username and password.

Step 3 Click **OK** to submit the username and password to the AAA server.

A popup window appears indicating whether the login attempt succeeded or failed. If the authentication is successful, the connection is completed automatically. If the authentication fails, the authentication proxy reports the failure to the user and prompts the user with multiple retries.

Note If the authentication attempt is unsuccessful after five attempts, the user must wait two minutes and initiate another HTTP session to trigger authentication proxy.

Establishing User Connections Without JavaScript

To ensure secure authentication, the authentication proxy design requires JavaScript. You can use the authentication proxy without enabling JavaScript on the browser, but this poses a potential security risk if users do not properly establish network connections. The following procedure provides the steps to properly establish a connection with JavaScript disabled. Network administrators are strongly advised to instruct users on how to properly establish connections using the procedure in this section.

Note Failure to follow this procedure can cause user credentials to be passed to a network web server other than the authentication proxy or can cause the authentication proxy to reject the login attempt.

To verify client connections using the authentication proxy when JavaScript is not enabled on the client browser, follow this procedure:

Step 1 Initiate an HTTP connection through the firewall.

This generates the authentication proxy login page.

Step 2 From the authentication proxy login page at the client, enter the username and password.

Step 3 Click **OK** to submit the username and password to the AAA server.

A popup window appears indicating whether the login attempt succeeded or failed. If the popup window indicates successful authentication, go to Step 7.

Step 4 If the popup window displays a failed authentication message, click **Close** on the browser **File** menu.

Note Do not click **Reload** (**Refresh** for Internet Explorer) to close the popup window.

Step 5 From the original authentication login page, click **Reload** (**Refresh** for Internet Explorer) on the browser toolbar. The user login credentials are cleared from the form.

Note Do not click **OK**. You must click **Reload** or **Refresh** to clear the username and password and to reload the form before attempting to log in again.

Step 6 Enter the username and password again.

If the authentication is successful, a window appears displaying a successful authentication message. If the window displays a failed authentication message, go to Step 4.

Step 7 Click **Close** on the browser **File** menu.

Step 8 From the original authentication proxy login page, click **Reload** (**Refresh** for Internet Explorer) on the browser toolbar.

The authentication proxy completes the authenticated connection with the web server.

Monitoring and Maintaining the Authentication Proxy

This section describes how to view dynamic access list entries and how to manually remove authentication entries. This section contains the following sections:

- Displaying Dynamic ACL Entries
- Deleting Authentication Proxy Cache Entries

Displaying Dynamic ACL Entries

You can display dynamic access list entries when they are in use. After an authentication proxy entry is cleared by you or by the idle timeout parameter, you can no longer display it. The number of matches displayed indicates the number of times the access list entry was hit.

To view dynamic access lists and any temporary access list entries that are currently established by the authentication proxy, use the **show ip access-lists** command in privileged EXEC mode:

Command	Purpose
`router# show ip access-lists`	Displays the standard and extended access lists configured on the firewall, including dynamic ACL entries.

Consider the following example where ACL 105 is applied inbound at the input interface where you configure authentication proxy. The initial display shows the contents of the ACLs prior to authentication. The second display shows the same displays after user authentication with the AAA server.

Note If NAT is configured, the **show ip access list** command might display the translated host IP address for the dynamic ACL entry or the IP address of the host initiating the connection. If the ACL is applied on the NAT outside interface, the translated address is displayed. If the ACL is applied on the NAT inside interface, the IP address of the host initiating the connection is displayed. The **show ip auth-proxy cache** command always displays the IP address of the host initiating the connection.

For example, the following is a list of ACL entries prior to the authentication proxy:

```
Router# show ip access-lists
    .
    .
    .
Extended IP access list 105
 deny tcp any any eq telnet
 deny udp any any
 permit tcp any any (28 matches)
 permit ip any any
```

The following sample output shows a list of ACL entries following user authentication:

```
Router# show ip access-lists
     .
     .
     .
Extended IP access list 105
! The ACL entries following user authentication are shown below.
 permit tcp host 192.168.25.215 any eq 26
 permit icmp host 192.168.25.215 host 60.0.0.2
 permit tcp host 192.168.25.215 any eq telnet
 permit tcp host 192.168.25.215 any eq ftp
 permit tcp host 192.168.25.215 any eq ftp-data
 permit tcp host 192.168.25.215 any eq smtp
 deny tcp any any eq telnet
 deny udp any any
 permit tcp any any (76 matches)
 permit ip any any
```

Deleting Authentication Proxy Cache Entries

When the authentication proxy is in use, dynamic access lists dynamically grow and shrink as authentication entries are added and deleted. To display the list of authentication entries, use the **show ip auth-proxy cache** command. To manually delete an authentication entry, use the **clear ip auth-proxy cache** command in privileged EXEC mode:

Command	Purpose
router# **clear ip auth-proxy cache** {* | *host ip address*}	Deletes authentication proxy entries from the firewall before they time out. Use an asterisk to delete all authentication cache entries. Enter a specific IP address to delete an entry for a single host.

Authentication Proxy Configuration Examples

Configuring the authentication proxy feature requires configuration changes on both the router and the AAA server. The following sections provide authentication proxy configuration examples:

- Authentication Proxy Configuration Example
- Authentication Proxy, IPSec, and CBAC Configuration Example
- Authentication Proxy, IPSec, NAT, and CBAC Configuration Example
- AAA Server User Profile Example

Throughout these examples, the exclamation point (!) indicates a comment line. Comment lines precede the configuration entries being described.

Authentication Proxy Configuration Example

The following examples highlight the specific authentication proxy configuration entries. These examples do not represent a complete router configuration. Complete router configurations using the authentication proxy are included later in this chapter.

This section contains the following examples:

- AAA Configuration Example
- HTTP Server Configuration Example
- Authentication Proxy Configuration Example
- Interface Configuration Example

AAA Configuration Example

```
aaa new-model
aaa authentication login default group tacacs group radius
! Set up the aaa new model to use the authentication proxy.
aaa authorization auth-proxy default group tacacs group radius
! Define the AAA servers used by the router.
aaa accounting auth-proxy default start-stop group tacacs+
! Set up authentication proxy with accounting.
tacacs-server host 172.31.54.143
tacacs-server key cisco
radius-server host 172.31.54.143
radius-server key cisco
```

HTTP Server Configuration Example

```
! Enable the HTTP server on the router.
ip http server
! Set the HTTP server authentication method to AAA.
ip http authentication aaa
! Define standard access list 61 to deny any host.
access-list 61 deny any
! Use ACL 61 to deny connections from any host to the HTTP server.
ip http access-class 61
```

Authentication Proxy Configuration Example

```
! Set the global authentication proxy timeout value.
ip auth-proxy auth-cache-time 60
! Apply a name to the authentication proxy configuration rule.
ip auth-proxy name HQ_users http
```

Interface Configuration Example

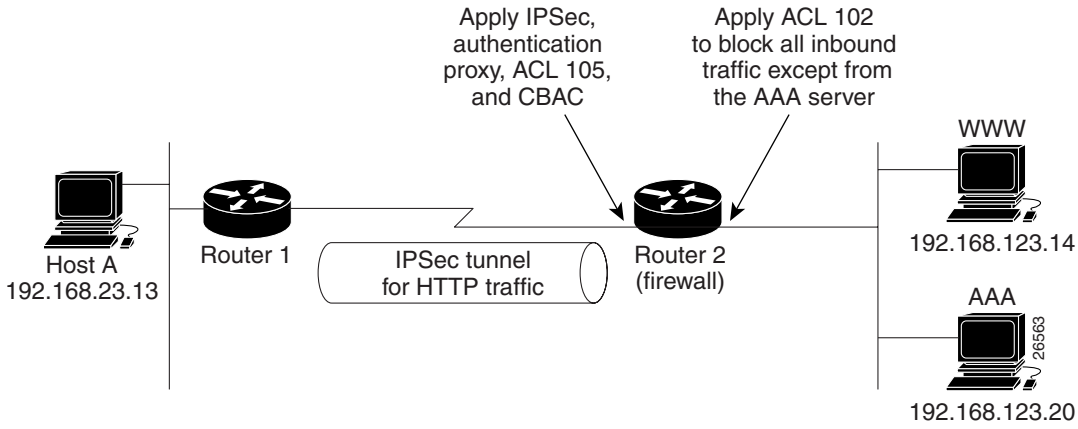
```
! Apply the authentication proxy rule at an interface.
interface e0
    ip address 10.1.1.210 255.255.255.0
    ip auth-proxy HQ_users
```

Authentication Proxy, IPSec, and CBAC Configuration Example

The following example shows a router configuration with the authentication proxy, IPSec, and CBAC features. Figure 47 illustrates the configuration.

Note If you are using this feature with Cisco IOS software release 12.3(8)T or later, see the document Crypto Access Check on Clear-Text Packets (feature module, release 12.3(8)T).

Figure 47 Authentication Proxy, IPSec, and CBAC Configuration Example



In this example, Host A initiates an HTTP connection with the web server (WWW). The HTTP traffic between Router 1 and Router 2 is encrypted using IPSec. The authentication proxy, IPSec, and CBAC are configured at interface Serial0 on Router 2, which is acting as the firewall. ACL 105 blocks all traffic at interface Serial0. ACL 102 is applied at interface Ethernet0 on Router 2 to block all traffic on that interface except traffic from the AAA server.

When Host A initiates an HTTP connection with the web server, the authentication proxy prompts the user at Host A for a username and password. These credentials are verified with the AAA server for authentication and authorization. If authentication is successful, the per-user ACLs are downloaded to the firewall to permit services.

The following examples provide both the Router 1 and Router 2 configurations for completeness:

- Router 1 Configuration Example
- Router 2 Configuration Example

Router 1 Configuration Example

```
! Configure Router 1 for IPSec.
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router1
!
logging buffered 4096 debugging
no logging console
enable secret 5 $1$E0OB$AQF1vFZM3fLr3LQAOsudL/
enable password junk
!
username Router2 password 0 welcome
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco1234 address 10.0.0.2
!
crypto ipsec transform-set rule_1 ah-sha-hmac esp-des esp-sha-hmac
!
!
 crypto map testtag 10 ipsec-isakmp
```

```
 set peer 10.0.0.2
 set transform-set rule_1
 match address 155
!
interface Ethernet0/0
 ip address 192.168.23.2 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Serial3/1
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 encapsulation PPP
 ip route-cache
 no ip mroute-cache
 no keepalive
 no fair-queue
 clockrate 56000
 crypto map testtag
!
!
ip classless
ip route 192.168.123.0 255.255.255.0 10.0.0.2
! Identify the IPSec specific traffic.
access-list 155 permit tcp host 192.168.23.13 host 192.168.123.14 eq www
access-list 155 permit tcp host 192.168.23.13 eq www host 192.168.123.14
```

Router 2 Configuration Example

```
! Configure Router 2 as the firewall, using the authentication proxy, IPSec, and CBAC.
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router2
!
logging buffered 4096 debugging
aaa new-model
aaa authentication login default group tacacs
aaa authentication login console_line none
aaa authentication login special none
aaa authentication ppp default group tacacs
aaa authorization exec default group tacacs
! Configure AAA for the authentication proxy.
aaa authorization auth-proxy default group tacacs+
enable password junk
!
! Create the CBAC inspection rule HTTP_TEST.
ip inspect name rule22 http
ip inspect name rule22 tcp
ip inspect name rule22 ftp
ip inspect name rule22 smtp
!
! Create the authentication proxy rule PXY.
ip auth-proxy name pxy http
! Turn on display of the router name in the authentication proxy login page.
ip auth-proxy auth-proxy-banner
ip audit notify log
ip audit po max-events 100
!
```

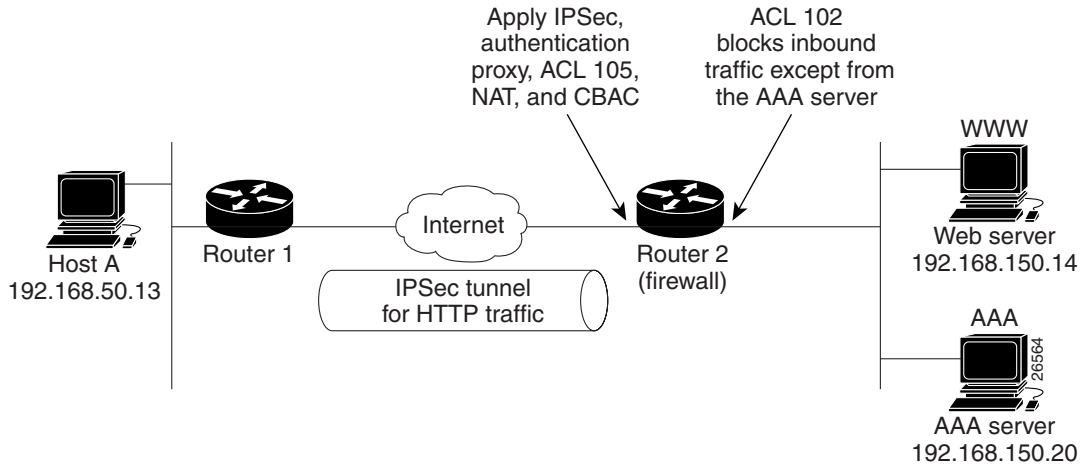
```
! Configure IPSec.
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco1234 address 10.0.0.1
!
crypto ipsec transform-set rule_1 ah-sha-hmac esp-des esp-sha-hmac
!
 crypto map testtag 10 ipsec-isakmp
 set peer 10.0.0.1
 set transform-set rule_1
 match address 155
!
! Apply the CBAC inspection rule and the authentication proxy rule at interface
! Serial0/0.
interface Serial0/0
 ip address 10.0.0.2 255.0.0.0
 ip access-group 105 in
 no ip directed-broadcast
 ip inspect rule22 in
 ip auth-proxy pxy
 encapsulation ppp
 no ip route-cache
 no ip mroute-cache
 no keepalive
 no fair-queue
crypto map testtag
!
interface Ethernet0/1
 ip address 192.168.123.2 255.255.255.0
 ip access-group 102 in
 no ip directed-broadcast
 ip route-cache
 no ip mroute-cache
!
no ip classless
ip route 192.168.23.0 255.255.255.0 10.0.0.1
ip route 192.168.50.0 255.255.255.0 16.0.0.1
! Configure the HTTP server.
ip http server
ip http access-class 15
ip http authentication aaa
!
! Create ACL 15 to block all traffic for the http server.
access-list 15 deny any
! Create ACL 102 to block all traffic inbound on interface Ethernet0/1 except for
! traffic from the AAA server.
access-list 102 permit tcp host 192.168.123.20 eq tacacs host 192.168.123.2
access-list 102 deny   tcp any any
access-list 102 deny   udp any any
access-list 102 permit ip any any
! Create ACL 105 to block all traffic inbound on interface Serial0/0. Permit only IP
! protocol traffic.
access-list 105 deny   tcp any any
access-list 105 deny   udp any any
access-list 105 permit ip any any
! Identify the IPSec specific traffic.
access-list 155 permit tcp host 192.168.123.14 host 192.168.23.13 eq www
access-list 155 permit tcp host 192.168.123.14 eq www host 192.168.23.13
!
! Define the AAA server host and encryption key.
tacacs-server host 192.168.123.14
tacacs-server key cisco
!
line con 0
```

```
 exec-timeout 0 0
 login authentication special
 transport input none
line aux 0
 transport input all
 speed 38400
 flowcontrol hardware
line vty 0 4
 password lab
```

Authentication Proxy, IPSec, NAT, and CBAC Configuration Example

The following example provides a router configuration with the authentication proxy, IPSec, NAT, and CBAC features. Figure 48 illustrates the configuration.

Figure 48 Authentication Proxy, IPSec, and CBAC Configuration Example



In this example, Host A initiates an HTTP connection with the web server (WWW). The HTTP traffic between router 1 (interface BRI0) and router 2 (interface Serial2) is encrypted using IPSec. The authentication proxy is configured on router 2, which is acting as the firewall. The authentication proxy, NAT, and CBAC are configured at interface Serial2, which is acting as the firewall. ACL 105 blocks all traffic at interface Serial2. ACL 102 is applied at interface Ethernet0 on router 2 to block all traffic on that interface except traffic from the AAA server. In this example, the authentication proxy uses standard ACL 10 to specify the hosts using the authentication proxy feature.

When any host in ACL 10 initiates an HTTP connection with the web server, the authentication proxy prompts the user at that host for a username and password. These credentials are verified with AAA server for authentication and authorization. If authentication is successful, the per-user ACLs are downloaded to the firewall to permit services.

The following examples provide both the router 1 and router 2 configurations for completeness:

- Router 1 Configuration Example
- Router 2 Configuration Example

Router 1 Configuration Example

```
! Configure router 1 for IPSec.
version 12.0
 service timestamps debug uptime
 service timestamps log uptime
 no service password-encryption
 !
 hostname Router1
 !
 logging buffered 4096 debugging
 no logging console
 !
isdn switch-type basic-5ess
 !
 crypto isakmp policy 1
  authentication pre-share
 crypto isakmp key cisco1234 address 16.0.0.2
 crypto ipsec transform-set rule_1 ah-sha-hmac esp-des esp-sha-hmac
 !
  !
  crypto map testtag 10 ipsec-isakmp
  set peer 16.0.0.2
  set transform-set rule_1
  match address 155
 !
 !
 process-max-time 200
 !
 interface BRI0
  ip address 16.0.0.1 255.0.0.0
  no ip directed-broadcast
  encapsulation ppp
  dialer idle-timeout 5000
  dialer map ip 16.0.0.2 name router2 broadcast 50006
  dialer-group 1
  isdn switch-type basic-5ess
  crypto map testtag
 !
 interface FastEthernet0
  ip address 192.168.50.2 255.255.255.0
  no ip directed-broadcast
 !
 ip classless
 ip route 192.168.150.0 255.255.255.0 16.0.0.2
 no ip http server
! Identify the IPSec specific traffic.
 access-list 155 permit tcp host 192.168.50.13 host 192.168.150.100 eq www
 access-list 155 permit tcp host 192.168.50.13 eq www host 192.168.150.100
 dialer-list 1 protocol ip permit
 !
 line con 0
  exec-timeout 0 0
  transport input none
 line aux 0
 line vty 0 4
  password lab
  login
```

Router 2 Configuration Example

```
! Configure router 2 as the firewall, using the authentication proxy, IPSec, NAT, and
```

```
! CBAC.
 version 12.0
 service timestamps debug uptime
 service timestamps log uptime
 no service password-encryption
 !
 hostname router2
 !
 logging buffered 4096 debugging
 aaa new-model
 aaa authentication login default group tacacs+
 aaa authentication login console_line none
 aaa authorization exec default group tacacs+
 ! Configure AAA for the authentication proxy.
 aaa authorization auth-proxy default group tacacs+
!
 ! Create the CBAC inspection rule "rule44."
 ip inspect name rule44 http java-list 5
 ip inspect name rule44 tcp
 ip inspect name rule44 ftp
 ip inspect name rule44 smtp
 !
 ! Create the authentication proxy rule "pxy." Set the timeout value for rule
 ! pxy to three minutes. Standard ACL 10 is applied to the rule.
 ip auth-proxy name pxy http list 10 auth-cache-time 3
 isdn switch-type primary-5ess
 !
 ! Configure IPSec.
 crypto isakmp policy 1
  authentication pre-share
 crypto isakmp key cisco1234 address 16.0.0.1
 !
 !
 crypto ipsec transform-set rule_1 ah-sha-hmac esp-des esp-sha-hmac
 !
  !
  crypto map testtag 10 ipsec-isakmp
  set peer 16.0.0.1
  set transform-set rule_1
  match address 155
!
 controller T1 2/0
  framing esf
  linecode b8zs
  pri-group timeslots 1-24
!
! Apply ACL 102 inbound at interface Ethernet0/1 and configure NAT.
interface Ethernet0/1
 ip address 192.168.150.2 255.255.255.0
 ip access-group 102 in
 no ip directed-broadcast
 ip nat inside
 no ip mroute-cache
!
! Apply the authentication proxy rule PXY, CBAC inspection rule HTTP_TEST, NAT, and
! and ACL 105 at interface Serial2/0:23.
interface Serial2/0:23
 ip address 16.0.0.2 255.0.0.0
 ip access-group 105 in
 no ip directed-broadcast
 ip nat outside
 ip inspect rule44 in
 ip auth-proxy pxy
 encapsulation ppp
```

```
 ip mroute-cache
 dialer idle-timeout 5000
 dialer map ip 16.0.0.1 name router1 broadcast 71011
 dialer-group 1
 isdn switch-type primary-5ess
 fair-queue 64 256 0
 crypto map testtag
!
! Use NAT to translate the Web server address.
ip nat inside source static 192.168.150.14 192.168.150.100
ip classless
ip route 192.168.50.0 255.255.255.0 16.0.0.1
! Configure the HTTP server.
ip http server
ip http access-class 15
ip http authentication aaa
!
! Create standard ACL 5 to specify the list of hosts from which to accept java applets.
! ACL 5 is used to block Java applets in the CBAC inspection rule named "rule44," which
! is applied at interface Serial2/0:23.
access-list 5 permit any
! Create standard ACL 10 to specify the hosts using the authentication proxy. This ACL
! used in the authentication proxy rule named "PXY", which is applied at interface
! Serial2/0:23.
access-list 10 permit any
! Create ACL 15 to block all traffic for the http server.
access-list 15 deny any
! Create extended ACL 102 to block all traffic inbound on interface Ethernet0/1
! except for traffic from the AAA server.
access-list 102 permit tcp host 192.168.150.20 eq tacacs 192.168.150.2
access-list 102 deny   tcp any any
access-list 102 deny   udp any any
access-list 102 permit ip any any
! Create extended ACL 105 to block all TCP and UDP traffic inbound on interface
! Serial2/0:23.
access-list 105 deny   tcp any any
access-list 105 deny   udp any any
access-list 105 permit ip any any
! Identify the IPSec specific traffic.
access-list 155 permit tcp host 192.168.150.100 host 192.168.50.13 eq www
access-list 155 permit tcp host 192.168.150.100 eq www host 192.168.50.13
dialer-list 1 protocol ip permit
! Define the AAA server host and encryption key.
tacacs-server host 192.168.126.14
tacacs-server key cisco
!
line con 0
 exec-timeout 0 0
! Define the AAA server host and encryption key.
 login authentication console_line
 transport input none
 line aux 0
 line vty 0 4
 password lab
!
!
 end
```

AAA Server User Profile Example

This section includes examples of the authentication proxy user profile entries on the AAA servers. The "proxyacl" entries define the user access privileges. After the user has successfully used the authentication proxy to log in, these entries are transferred to the firewall router. Each entry in the profile must specify "permit" access for the service or application. The source address in each entry is set to "any", which is replaced with the IP address of the authenticating host when the profile is downloaded to the firewall. The privilege level must be set to 15 for all AAA users.

This section contains the following sections:

- CiscoSecure ACS 2.3 for Windows NT
- CiscoSecure ACS 2.3 for UNIX
- TACACS+ Server
- Livingston Radius Server
- Ascend Radius Server

CiscoSecure ACS 2.3 for Windows NT

This section describes how to configure authentication proxy on CiscoSecure ACS 2.3 for Windows NT. For detailed information about CiscoSecure ACS, refer to the documentation for that product.

The following sample configuration is for the TACACS+ service of CiscoSecure ACS for Windows NT.

Step 1 Click the Interface Configuration icon and click **TACACS+** (Cisco).

a. Scroll down to New Services.

b. Add a new service, "auth-proxy", in the Service field. Leave the Protocol field empty.

c. Select both the User and Group check boxes for the new service.

d. Scoll down to Advance Configuration Options and check the Per-user Advance TACACS+ features.

e. Click **Submit**.

Step 2 Click the Network Configuration icon.

a. Click the Add Entry icon for Network Access Servers and fill in the Network Access Server Hostname, IP address, and key (the key configured on the router) fields.

b. Select TACACS+ (Cisco) for the Authenticate Using option.

c. Click the Submit + Restart icon.

Step 3 Click the Group Setup icon.

a. Select a user group from the drop-down menu.

b. Select the Users in Group check box.

c. Select a user from the user list.

d. In the User Setup list, scroll down to TACACS+ Settings and select the "auth-proxy" check box.

e. Select the Custom Attributes check box.

f. Add the profile entries (do not use single or double quotes around the entries) and set the privilege level to 15.

```
priv-lvl=15
proxyacl#1=permit tcp any any eq 26
```

```
proxyacl#2=permit icmp any host 60.0.0.2
proxyacl#3=permit tcp any any eq ftp
proxyacl#4=permit tcp any any eq ftp-data
proxyacl#5=permit tcp any any eq smtp
proxyacl#6=permit tcp any any eq telnet
```

g. Click **Submit**.

Step 4 Click the User Setup icon.

a. Click **List All Users**.

b. Add a username.

c. Scoll down to User Setup Password Authentication.

d. Select SDI SecurID Token Card from the Password Authentication drop-down menu.

e. Select the previous configured user group 1.

f. Click **Submit**.

Step 5 Click Group Setup icon again.

a. Select the user group 1.

b. Click **Users in Group**.

c. Click **Edit Settings**.

d. Click the Submit + Restart icon to make sure the latest configuration is updated and sent to the AAA server.

CiscoSecure ACS 2.3 for UNIX

This section describes how to configure authentication proxy on CiscoSecure ACS 2.3 for UNIX. For detailed information regarding CiscoSecure ACS, refer to the documentation for that product.

To manage the CiscoSecure ACS using the Administrator program, you need a web browser that supports Java and JavaScript. You must enable Java in the browser application. You can start the Java-based CiscoSecure Administrator advanced configuration program from any of the CiscoSecure ACS Administrator web pages.

The following sample configuration procedure is for the TACACS+ service of CiscoSecure ACS 2.3 for UNIX.

Step 1 On the CiscoSecure ACS web menu bar of the CiscoSecure ACS web interface, click **Advanced** and then click **Advanced** again.

The Java-based CiscoSecure Administrator advanced configuration program appears. It might require a few minutes to load.

Step 2 In the CiscoSecure Administrator advanced configuration program, locate and deselect Browse in the Navigator pane of the tabbed Members page.

This displays the Create New Profile icon.

Step 3 In the Navigator pane, do one of the following:

- Locate and click the group to which the user will belong.
- If you do not want the user to belong to a group, click the [Root] folder icon.

Step 4 Click **Create Profile** to display the New Profile dialog box.

Step 5 Make sure the Group check box is cleared.

Step 6 Enter the name of the user you want to create and click **OK**. The new user appears in the tree.

Step 7 Click the icon for the group or user profile in the tree that is displayed in the Navigator pane of the tabbed Members page.

Step 8 If necessary, in the Profile pane, click the Profile icon to expand it.

A list or dialog box that contains attributes applicable to the selected profile or service appears in the window at the bottom right of the screen. The information in this window changes depending on what you have selected in the Profile pane.

Step 9 Click **Service-String**.

Step 10 Click **string**, enter **auth-proxy** in the text field, and click **Apply**.

Step 11 Select the **Option** menu.

Step 12 On the **Option** menu, click **Default Attributes**.

Step 13 Change the attribute from Deny to **Permit**.

Step 14 Click **Apply**.

Step 15 On the **Option** menu, click **Attribute** and enter the privilege level in the text field:

```
priv-lvl=15
```

Step 16 On the **Option** menu, click **Attribute** and enter the **proxyacl** entries in the text field:

```
proxyacl#1=”permit tcp any any eq 26”
```

Repeat this step for each additional service or protocol to add:

```
proxyacl#2=”permit icmp any host 60.0.0.2”
proxyacl#3=”permit tcp any any eq ftp”
proxyacl#4=”permit tcp any any eq ftp-data”
proxyacl#5=”permit tcp any any eq smtp”
proxyacl#6=”permit tcp any any eq telnet”
```

Step 17 When you have finished making all your changes, click **Submit**.

TACACS+ Server

```
default authorization = permit
key = cisco
user = Brian {
login = cleartext cisco
service = auth-proxy
 {
  priv-lvl=15
  proxyacl#1="permit tcp any any eq 26"
  proxyacl#2="permit icmp any host 60.0.0.2
  proxyacl#3="permit tcp any any eq ftp"
  proxyacl#4="permit tcp any any eq ftp-data"
  proxyacl#5="permit tcp any any eq smtp"
  proxyacl#6="permit tcp any any eq telnet"
 }
}
```

Livingston Radius Server

```
Bob Password = "cisco" User-Service-Type=Outbound-User
cisco-avpair = "auth-proxy:priv-lvl=15",
cisco-avpair = "auth-proxy:proxyacl#1=permit tcp any any eq 26",
cisco-avpair = "auth-proxy:proxyacl#2=permit icmp any host 60.0.0.2",
cisco-avpair = "auth-proxy:proxyacl#3=permit tcp any any eq ftp",
cisco-avpair = "auth-proxy:proxyacl#4=permit tcp any any eq ftp-data",
cisco-avpair = "auth-proxy:proxyacl#5=permit tcp any any eq smtp",
cisco-avpair = "auth-proxy:proxyacl#6=permit tcp any any eq telnet"
```

Ascend Radius Server

```
Alice Password = "cisco" User-Service = Dialout-Framed-User
cisco-avpair = "auth-proxy:priv-lvl=15",
cisco-avpair = "auth-proxy:proxyacl#1=permit tcp any any eq 26",
cisco-avpair = "auth-proxy:proxyacl#2=permit icmp any host 60.0.0.2",
cisco-avpair = "auth-proxy:proxyacl#3=permit tcp any any eq ftp",
cisco-avpair = "auth-proxy:proxyacl#4=permit tcp any any eq ftp-data",
cisco-avpair = "auth-proxy:proxyacl#5=permit tcp any any eq smtp",
cisco-avpair = "auth-proxy:proxyacl#6=permit tcp any any eq telnet"
```

Firewall Support of HTTPS Authentication Proxy

The Firewall Support of HTTPS Authentication Proxy feature allows a user to encrypt the change of the username and password between the HTTP client and the Cisco IOS router via Secure Socket Layer (SSL) when authentication proxy is enabled on the Cisco IOS firewall, thereby ensuring confidentiality of the data passing between the HTTP client and the Cisco IOS router.

Feature Specifications for the Firewall Support of HTTPS Authentication Proxy feature

Feature History	
Release	**Modification**
12.2(11)YU	This feature was introduced.
12.2(15)T	This feature was integrated into Cisco IOS Release 12.2(15)T.
Supported Platforms	
For platforms supported in Cisco IOS Releases 12.2(11)YU and 12.2(15)T, consult Cisco Feature Navigator.	

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for Firewall Support of HTTPS Authentication Proxy, page 802
- Restrictions for Firewall Support of HTTPS Authentication Proxy, page 802
- Information About Firewall Support of HTTPS Authentication Proxy, page 802
- How to Use HTTPS Authentication Proxy, page 804
- Monitoring Firewall Support of HTTPS Authentication Proxy, page 806
- Additional References, page 812
- Command Reference, page 814
- Glossary, page 815

Prerequisites for Firewall Support of HTTPS Authentication Proxy

Before enabling this feature, ensure that your router is running a crypto image with k8 and k9 designations and that your Cisco IOS image supports SSL.

Restrictions for Firewall Support of HTTPS Authentication Proxy

- Although Port to Application Mapping (PAM) configuration is allowed in Cisco IOS Firewall processing, authentication proxy is limited to the server ports that are configured by the HTTP subsystem of the router.
- To conform to a proper TCP connection handshake, the authentication proxy login page will be returned from the same port and address as the original request. Only the postrequest, which contains the username and password of the HTTP client, will be forced to use HTTP over SSL (HTTPS).

Information About Firewall Support of HTTPS Authentication Proxy

To configure the Firewall Support of HTTPS Authentication Proxy feature, you must understand the following concepts:

- Authentication Proxy, page 802
- Feature Design for HTTPS Authentication Proxy, page 803

Authentication Proxy

Authentication proxy grants Internet access to an authorized user through the Cisco Secure Integrated Software (also known as a Cisco IOS firewall). Access is granted on a per-user basis after the proper identification process is completed and the user policies are retrieved from a configured authentication, authorization, and accounting (AAA) server.

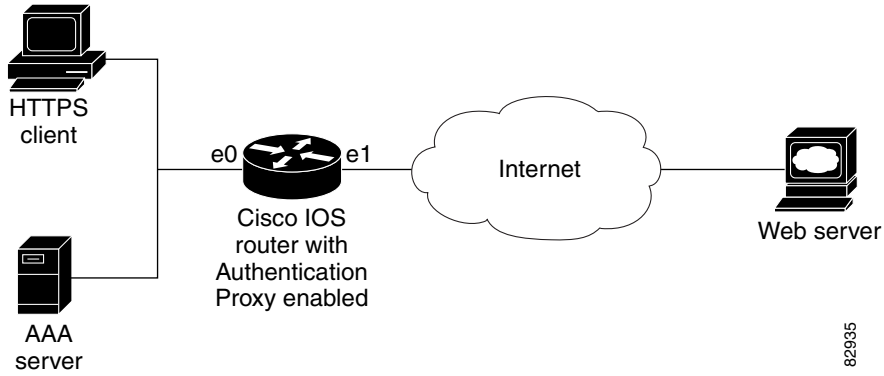
When authentication proxy is enabled on a Cisco router, users can log into the network or access the Internet via HTTP(S). When a user initiates an HTTP(S) session through the firewall, the authentication proxy is triggered. Authentication proxy first checks to see if the user has been authenticated. If a valid authentication entry exists for the user, the connection is completed with no further intervention by authentication proxy. If no entry exists, the authentication proxy responds to the HTTP(S) connection request by prompting the user for a username and password. When authenticated, the specific access profiles are automatically retrieved and applied from a CiscoSecure Access Control Server (ACS), or other RADIUS or TACACS+ authentication server. The user profiles are active only when there is active traffic from the authenticated users.

Feature Design for HTTPS Authentication Proxy

Authentication proxy support using HTTPS provides encryption between the HTTPS client and the Cisco IOS router during the username and password exchange, ensuring secure communication between trusted entities.

Figure 49 and the corresponding steps explain how the data flows from the time the client issues a HTTP request to the time the client receives a response from the Cisco IOS router.

Figure 49 HTTPS Authentication Proxy Data Flow



1. The HTTP or HTTPS client requests a web page.
2. The HTTP or HTTPS request is intercepted by the Cisco IOS router with authentication proxy.
3. The router marks the TCP/IP connection and forwards the request (with the client address) to the web server, if authentication is required.
4. The web server builds the authentication request form and sends it to the HTTP or HTTPS client via the original request protocol—HTTP or HTTPS.
5. The HTTP or HTTPS client receives the authentication request form.
6. The user enters his or her username and password in the HTTPS POST form and returns the form to the router. At this point, the authentication username and password form is sent via HTTPS. The web server will negotiate a new SSL connection with the HTTPS client.

Note Your Cisco IOS image must support HTTPS, and HTTPS must be configured; otherwise, an HTTP request form will be generated.

7. The router receives the HTTPS POST form from the HTTPS client and retrieves the username and password.
8. The router sends the username and password to the AAA server for client authentication.
9. If the AAA server validates the username and password, it sends the configured user profile to the router. (If it cannot validate the username and password, an error is generated and sent to the router.)

10. If the router receives a user profile from the AAA server, it updates the access list with the user profile and returns a successful web page to the HTTPS client. (If the router receives an error from the AAA server, it returns an error web page to the HTTPS client.)
11. After the HTTPS client receives the successful web page, it retries the original request. Thereafter, HTTPS traffic will depend on HTTPS client requests; no router intervention will occur.

How to Use HTTPS Authentication Proxy

To enable HTTPS authentication proxy, you must enable AAA service, configure the HTTPS server, and enable authentication proxy. This section contains the following procedures:

- Configuring the HTTPS Server, page 804
- Verifying HTTPS Authentication Proxy, page 805

Configuring the HTTPS Server

To use HTTPS authentication proxy, you must enable the HTTPS server on the firewall and set the HTTPS server authentication method to use AAA.

Prerequisites

Before configuring the HTTPS server, you must perform the following procedures:

- Configure the authentication proxy for AAA services by enabling AAA and configuring a RADIUS or TACACS+ server. For information on completing these tasks, refer to the section "Configuring AAA" in the chapter "Configuring Authentication Proxy" of the *Cisco IOS Security Configuration Guide*, Release 12.2.
- Obtain a certification authority (CA) certificate. For information on completing this task, refer to the section "Configuring a Trustpoint CA" in the *Trustpoint CLI,* Cisco IOS Release 12.2(8)T feature module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http server**
4. **ip http authentication aaa**
5. **ip http secure-server**
6. **ip http secure-trustpoint** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip http server** **Example:** `Router (config)# ip http server`	Enables the HTTP server on the router. • The authentication proxy uses the HTTP server to communicate with the client for user authentication.
Step 4	**ip http authentication aaa** `Router (config)# ip http authentication aaa`	Sets the HTTP server authentication method to AAA.
Step 5	**ip http secure-server** **Example:** `Router (config)# ip http secure-server`	Enables HTTPS.
Step 6	**ip http secure-trustpoint** *name* **Example:** `Router (config)# ip http secure-trustpoint netCA`	Enables HTTP secure server certificate trustpoint.

What to Do Next

After you have finished configuring the HTTPS server, you must configure the authentication proxy (globally and per interface). For information on completing this task, refer to the section "Configuring the Authentication Proxy" in the chapter "Configuring Authentication Proxy" of the *Cisco IOS Security Configuration Guide*, Release 12.2.

Verifying HTTPS Authentication Proxy

To verify your HTTPS authentication proxy configuration, perform the following optional steps:

SUMMARY STEPS

1. **enable**
2. **show ip auth-proxy configuration**
3. **show ip auth-proxy cache**
4. **show ip http server secure status**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip auth-proxy configuration** **Example:** `Router# show ip auth-proxy configuration`	Displays the current authentication proxy configuration.
Step 3	**show ip auth-proxy cache** **Example:** `Router# show ip auth-proxy cache`	Displays the list of user authentication entries. The authentication proxy cache lists the host IP address, the source port number, the timeout value for the authentication proxy, and the state of the connection. If the authentication proxy state is HTTP_ESTAB, the user authentication was successful.
Step 4	**show ip http server secure status** **Example:** `Router# show ip http server secure status`	Displays HTTPS status.

Monitoring Firewall Support of HTTPS Authentication Proxy

Perform the following task to troubleshoot your HTTPS authentication proxy configuration:

SUMMARY STEPS

1. **enable**
2. **debug ip auth-proxy detailed**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**Example:** **debug ip auth-proxy detailed** **Example:** `Router# debug ip auth-proxy detailed`	Displays the authentication proxy configuration information on the router.

Configuration Examples for HTTPS Authentication Proxy

This section provides the following comprehensive configuration examples:

- HTTPS Authentication Proxy Support Example, page 807
- RADIUS User Profile Example, page 810
- TACACS User Profile Example, page 810
- HTTPS Authentication Proxy Debug Example, page 811

HTTPS Authentication Proxy Support Example

The following example is output from the **show running-config** command. This example shows how to enable HTTPS authentication proxy on a Cisco IOS router.

```
Router# show running-config

Building configuration...

Current configuration : 6128 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname 7200a
!
boot system disk0:c7200-ik9o3s-mz.emweb
aaa new-model
!
!
aaa authentication login default group tacacs+ group radius
aaa authorization auth-proxy default group tacacs+ group radius
aaa session-id common
!
ip subnet-zero
ip cef
!
!
ip domain name cisco.com
!
ip auth-proxy auth-proxy-banner
ip auth-proxy auth-cache-time 3
ip auth-proxy name authname http
ip audit notify log
ip audit po max-events 100
!
! Obtain a CA certificate.
crypto ca trustpoint netCA
 enrollment mode ra
 enrollment url http://10.3.10.228:80/certsrv/mscep/mscep.dll
 subject-name CN=7200a.cisco.com
 crl optional
crypto ca certificate chain netCA
 certificate ca 0702EFC30EC4B18D471CD4531FF77E29
  308202C5 3082026F A0030201 02021007 02EFC30E C4B18D47 1CD4531F F77E2930
  0D06092A 864886F7 0D010105 0500306D 310B3009 06035504 06130255 53310B30
  09060355 04081302 434F3110 300E0603 55040713 07426F75 6C646572 31163014
  06035504 0A130D43 6973636F 20537973 74656D73 310C300A 06035504 0B130349
```

```
  54443119 30170603 55040313 10495444 20426F75 6C646572 202D2043 41301E17
  0D303230 31323532 33343434 375A170D 31323031 32353233 35343333 5A306D31
  0B300906 03550406 13025553 310B3009 06035504 08130243 4F311030 0E060355
  04071307 426F756C 64657231 16301406 0355040A 130D4369 73636F20 53797374
  656D7331 0C300A06 0355040B 13034954 44311930 17060355 04031310 49544420
  426F756C 64657220 2D204341 305C300D 06092A86 4886F70D 01010105 00034B00
  30480241 00B896F0 7CE9DCBD 59812309 1793C610 CEC83704 D56C29CA 3E8FAC7A
  A113520C E15E3DEF 64909FB9 88CD43BD C7DFBAD6 6D523804 3D958A97 9733EE71
  114D8F3F 8B020301 0001A381 EA3081E7 300B0603 551D0F04 04030201 C6300F06
  03551D13 0101FF04 05300301 01FF301D 0603551D 0E041604 14479FE0 968DAD8A
  46774122 2276C19B 6800FA3C 79308195 0603551D 1F04818D 30818A30 42A040A0
  3E863C68 7474703A 2F2F6369 73636F2D 736A7477 77383779 792F4365 7274456E
  726F6C6C 2F495444 25323042 6F756C64 65722532 302D2532 3043412E 63726C30
  44A042A0 40863E66 696C653A 2F2F5C5C 63697363 6F2D736A 74777738 3779795C
  43657274 456E726F 6C6C5C49 54442532 30426F75 6C646572 2532302D 25323043
  412E6372 6C301006 092B0601 04018237 15010403 02010030 0D06092A 864886F7
  0D010105 05000341 0044DE07 3964E080 09050906 512D40C0 D4D86A0A 6B33E752
  6E602D96 3F68BB8E 463E3EF6 D29BE400 615E7226 87DE1DE3 96AE23EF E076EE60
  BF789728 5ED0D5FC 2C
  quit
 certificate 55A4795100000000000D
  308203FC 308203A6 A0030201 02020A55 A4795100 00000000 0D300D06 092A8648
  86F70D01 01050500 306D310B 30090603 55040613 02555331 0B300906 03550408
  1302434F 3110300E 06035504 07130742 6F756C64 65723116 30140603 55040A13
  0D436973 636F2053 79737465 6D73310C 300A0603 55040B13 03495444 31193017
  06035504 03131049 54442042 6F756C64 6572202D 20434130 1E170D30 32303631
  38323030 3035325A 170D3033 30363138 32303130 35325A30 3A311E30 1C06092A
  864886F7 0D010902 130F3732 3030612E 63697363 6F2E636F 6D311830 16060355
  0403130F 37323030 612E6369 73636F2E 636F6D30 5C300D06 092A8648 86F70D01
  01010500 034B0030 48024100 F61D6551 77F9CABD BC3ACAAC D564AE53 541A40AE
  B89B6215 6A6D8D88 831F672E 66678331 177AF07A F476CD59 E535DAD2 C145E41D
  BF33BEB5 83DF2A39 887A05BF 02030100 01A38202 59308202 55300B06 03551D0F
  04040302 05A0301D 0603551D 0E041604 147056C6 ECE3A7A4 E4F9AFF9 20F23970
  3F8A7BED 323081A6 0603551D 2304819E 30819B80 14479FE0 968DAD8A 46774122
  2276C19B 6800FA3C 79A171A4 6F306D31 0B300906 03550406 13025553 310B3009
  06035504 08130243 4F311030 0E060355 04071307 426F756C 64657231 16301406
  0355040A 130D4369 73636F20 53797374 656D7331 0C300A06 0355040B 13034954
  44311930 17060355 04031310 49544420 426F756C 64657220 2D204341 82100702
  EFC30EC4 B18D471C D4531FF7 7E29301D 0603551D 110101FF 04133011 820F3732
  3030612E 63697363 6F2E636F 6D308195 0603551D 1F04818D 30818A30 42A040A0
  3E863C68 7474703A 2F2F6369 73636F2D 736A7477 77383779 792F4365 7274456E
  726F6C6C 2F495444 25323042 6F756C64 65722532 302D2532 3043412E 63726C30
  44A042A0 40863E66 696C653A 2F2F5C5C 63697363 6F2D736A 74777738 3779795C
  43657274 456E726F 6C6C5C49 54442532 30426F75 6C646572 2532302D 25323043
  412E6372 6C3081C6 06082B06 01050507 01010481 B93081B6 30580608 2B060105
  05073002 864C6874 74703A2F 2F636973 636F2D73 6A747777 38377979 2F436572
  74456E72 6F6C6C2F 63697363 6F2D736A 74777738 3779795F 49544425 3230426F
  756C6465 72253230 2D253230 43412E63 7274305A 06082B06 01050507 3002864E
  66696C65 3A2F2F5C 5C636973 636F2D73 6A747777 38377979 5C436572 74456E72
  6F6C6C5C 63697363 6F2D736A 74777738 3779795F 49544425 3230426F 756C6465
  72253230 2D253230 43412E63 7274300D 06092A86 4886F70D 01010505 00034100
  9BAE173E 337CAD74 E95D5382 A5DF7D3C 91F69832 761E374C 0E1E4FD6 EBDE59F6
  5B8D0745 32C3233F 25CF45FE DEEEB73E 8E5AD908 BF7008F8 BB957163 D63D31AF
  quit
!!
!
voice call carrier capacity active
!
!
interface FastEthernet0/0
 ip address 192.168.126.33 255.255.255.0
 duplex half
 no cdp enable
!
```

```
interface ATM1/0
 no ip address
 shutdown
 no atm ilmi-keepalive
!
interface FastEthernet2/0
 no ip address
 shutdown
 duplex half
 no cdp enable
!
interface FastEthernet3/0
 ip address 192.168.26.33 255.255.255.0
! Configure auth-proxy interface.
 ip auth-proxy authname
 duplex half
 no cdp enable
!
interface FastEthernet4/0
 ip address 10.3.10.46 255.255.0.0
 duplex half
 no cdp enable
!
interface FastEthernet4/0.1
!
ip nat inside source static 192.168.26.2 192.168.26.25
ip classless
! Configure the HTTPS server.
ip http server
ip http authentication aaa
ip http secure-trustpoint netCA
ip http secure-server
ip pim bidir-enable
!
!
access-list 101 deny   tcp any any
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
! Configure AAA and RADIUS server.
tacacs-server host 192.168.126.3
tacacs-server key letmein
!
radius-server host 192.168.126.2 auth-port 1645 acct-port 1646
radius-server retransmit 3
radius-server key letmein
radius-server authorization permit missing Service-Type
call rsvp-sync
!
!
mgcp profile default
!
dial-peer cor custom
!!
!
gatekeeper
 shutdown
!
!
line con 0
line aux 0
line vty 0 4
 password letmein
!
```

```
!
end
```

RADIUS User Profile Example

The following example is a sample RADIUS user profile for Livingston RADIUS:

```
#--------------- Proxy user ---------------------------------

http                    Password = "test" User-Service-Type=Outbound-User
       cisco-avpair = "auth-proxy:priv-lvl=15",
       cisco-avpair = "auth-proxy:proxyacl#3=permit tcp any any eq 23"

http_1                     Password = "test"
         User-Service-Type = Shell-User,
         User-Service-Type=Dialout-Framed-User,
         cisco-avpair = "shell:priv-lvl=15",
         cisco-avpair = "shell:inacl#4=permit tcp any host 192.168.134.216
eq 23
         cisco-avpair = "auth-proxy:priv-lvl=15",
         cisco-avpair = "auth-proxy:proxyacl#3=permit tcp any any eq 23"


http_fail               Password = "test" User-Service-Type=Outbound-User
       cisco-avpair = "auth-proxy:priv-lvl=14",
       cisco-avpair = "auth-proxy:proxyacl#3=permit tcp any any eq 23"

proxy  Password = "cisco" User-Service-Type=Outbound-User       cisco-avpair =
"auth-proxy:proxyacl#4=permit tcp any any eq 20"
```

TACACS User Profile Example

The following examples are sample TACACS user profiles:

```
default authorization = permit
key = cisco
user = http_1 {
  default service = permit
      login = cleartext test
         service = exec
        {
                priv-lvl = 15
                inacl#4="permit tcp any host 192.168.134.216 eq 23"
                inacl#5="permit tcp any host 192.168.134.216 eq 20"
                inacl#6="permit tcp any host 192.168.134.216 eq 21"
                inacl#3="deny -1"

        }
      service = auth-proxy
          {
              priv-lvl=15
              proxyacl#4="permit tcp any host 192.168.105.216 eq 23"
              proxyacl#5="permit tcp any host 192.168.105.216 eq 20"
              proxyacl#6="permit tcp any host 192.168.105.216 eq 21"
              proxyacl#7="permit tcp any host 192.168.105.216 eq 25"
          }
```

```
}
user = http {
          login = cleartext test
      service = auth-proxy
         {
              priv-lvl=15
              proxyacl#4=”permit tcp any host 192.168.105.216 eq 23”
              proxyacl#5=”permit tcp any host 192.168.105.216 eq 20”
              proxyacl#6=”permit tcp any host 192.168.105.216 eq 21”
          }
}
 user = proxy_1 {
          login = cleartext test
      service = auth-proxy
          {
              priv-lvl=14
        }
}

user = proxy_3 {
          login = cleartext test
      service = auth-proxy
          {
              priv-lvl=15
```

HTTPS Authentication Proxy Debug Example

The following is a sample of **debug ip auth-proxy detailed** command output:

```
*Mar  1 21:18:18.534: AUTH-PROXY:proto_flag=7, dstport_index=4
*Mar  1 21:18:18.534:  SYN SEQ 462612879 LEN 0
*Mar  1 21:18:18.534: dst_addr 172.16.171.219 src_addr 171.69.89.25 dst_port 80
src_port 3061
*Mar  1 21:18:18.538: AUTH-PROXY:auth_proxy_half_open_count++ 1
*Mar  1 21:18:18.542: AUTH-PROXY:proto_flag=7, dstport_index=4
*Mar  1 21:18:18.542:  ACK 3715697587 SEQ 462612880 LEN 0
*Mar  1 21:18:18.542: dst_addr 172.16.171.219 src_addr 171.69.89.25 dst_port 80
src_port 3061
*Mar  1 21:18:18.542: clientport 3061 state 0
*Mar  1 21:18:18.542: AUTH-PROXY:proto_flag=7, dstport_index=4
*Mar  1 21:18:18.542:  PSH ACK 3715697587 SEQ 462612880 LEN 250
*Mar  1 21:18:18.542: dst_addr 172.16.171.219 src_addr 171.69.89.25 dst_port 80
src_port 3061
*Mar  1 21:18:18.542: clientport 3061 state 0
*Mar  1 21:18:18.554: AUTH-PROXY:proto_flag=7, dstport_index=4
*Mar  1 21:18:18.554:  ACK 3715698659 SEQ 462613130 LEN 0
*Mar  1 21:18:18.554: dst_addr 172.16.171.219 src_addr 171.69.89.25 dst_port 80
src_port 3061
*Mar  1 21:18:18.554: clientport 3061 state 0
*Mar  1 21:18:18.610: AUTH-PROXY:proto_flag=7, dstport_index=4
*Mar  1 21:18:18.610:  ACK 3715698746 SEQ 462613130 LEN 0
*Mar  1 21:18:18.610: dst_addr 172.16.171.219 src_addr 171.69.89.25 dst_port 80
src_port 3061
*Mar  1 21:18:18.610: clientport 3061 state 0
*Mar  1 21:18:18.766: AUTH-PROXY:proto_flag=7, dstport_index=4
*Mar  1 21:18:18.766:  FIN ACK 3715698746 SEQ 462613130 LEN 0
*Mar  1 21:18:18.766: dst_addr 172.16.171.219 src_addr 171.69.89.25 dst_port 80
src_port 3061
*Mar  1 21:18:18.766: clientport 3061 state 0
*Mar  1 21:18:33.070: AUTH-PROXY:proto_flag=7, dstport_index=4
*Mar  1 21:18:33.070:  SYN SEQ 466414843 LEN 0
*Mar  1 21:18:33.070: dst_addr 172.16.171.219 src_addr 171.69.89.25 dst_port 80
```

```
src_port 3064
*Mar  1 21:18:33.070: clientport 3061 state 0
*Mar  1 21:18:33.074: AUTH-PROXY:proto_flag=7, dstport_index=4
*Mar  1 21:18:33.074:  ACK 1606420512 SEQ 466414844 LEN 0
*Mar  1 21:18:33.074: dst_addr 172.16.171.219 src_addr 171.69.89.25 dst_port 80
src_port 3064
*Mar  1 21:18:33.074: clientport 3064 state 0
*Mar  1 21:18:33.078: AUTH-PROXY:proto_flag=7, dstport_index=4
*Mar  1 21:18:33.078:  PSH ACK 1606420512 SEQ 466414844 LEN 431
*Mar  1 21:18:33.078: dst_addr 172.16.171.219 src_addr 171.69.89.25 dst_port 80
src_port 3064
*Mar  1 21:18:33.078: clientport 3064 state 0
*Mar  1 21:18:33.090: AUTH-PROXY:proto_flag=7, dstport_index=0
*Mar  1 21:18:33.090: AUTH-PROXY:Protocol not configured on if_input
*Mar  1 21:18:33.226: AUTH-PROXY:proto_flag=7, dstport_index=0
*Mar  1 21:18:33.226: AUTH-PROXY:Protocol not configured on if_input
*Mar  1 21:18:33.546: AUTH-PROXY:proto_flag=7, dstport_index=0
*Mar  1 21:18:33.546: AUTH-PROXY:Protocol not configured on if_input
*Mar  1 21:18:33.550: AUTH-PROXY:proto_flag=7, dstport_index=0
*Mar  1 21:18:33.550: AUTH-PROXY:Protocol not configured on if_input
*Mar  1 21:18:33.594: AUTH-PROXY:proto_flag=7, dstport_index=0
*Mar  1 21:18:33.594: AUTH-PROXY:Protocol not configured on if_input
*Mar  1 21:18:33.594: AUTH-PROXY:proto_flag=7, dstport_index=0
*Mar  1 21:18:33.594: AUTH-PROXY:Protocol not configured on if_input
*Mar  1 21:18:33.598: AUTH-PROXY:proto_flag=7, dstport_index=0
*Mar  1 21:18:33.598: AUTH-PROXY:Protocol not configured on if_input
*Mar  1 21:18:33.706: AUTH-PROXY:proto_flag=7, dstport_index=0
*Mar  1 21:18:33.706: AUTH-PROXY:Protocol not configured on if_input
*Mar  1 21:18:33.810: AUTH-PROXY:proto_flag=7, dstport_index=0
*Mar  1 21:18:33.810: AUTH-PROXY:Protocol not configured on if_input
*Mar  1 21:18:33.810: AUTH-PROXY:proto_flag=7, dstport_index=0
*Mar  1 21:18:33.810: AUTH-PROXY:Protocol not configured on if_input
*Mar  1 21:18:33.810: AUTH-PROXY:proto_flag=7, dstport_index=4
*Mar  1 21:18:33.810:  ACK 1606421496 SEQ 466415275 LEN 0
*Mar  1 21:18:33.810: dst_addr 172.16.171.219 src_addr 171.69.89.25 dst_port 80
src_port 3064
*Mar  1 21:18:33.814: clientport 3064 state 6
*Mar  1 21:18:33.814: AUTH-PROXY:Packet in FIN_WAIT state
*Mar  1 21:18:33.838: AUTH-PROXY:proto_flag=7, dstport_index=4
*Mar  1 21:18:33.838:  FIN ACK 1606421496 SEQ 466415275 LEN 0
*Mar  1 21:18:33.838: dst_addr 172.16.171.219 src_addr 171.69.89.25 dst_port 80
src_port 3064
*Mar  1 21:18:33.838: clientport 3064 state 6
*Mar  1 21:18:33.838: AUTH-PROXY:Packet in FIN_WAIT state
```

Additional References

For additional information related to the Firewall Support of HTTPS Authentication Proxy feature, refer to the following references:

- Related Documents, page 813
- Standards, page 813
- MIBs, page 813
- RFCs, page 814
- Technical Assistance, page 814

Related Documents

Related Topic	Document Title
Authentication proxy configuration tasks	The chapter "Configuring Authentication Proxy" in the *Cisco IOS Security Configuration Guide*, Release 12.2
Authentication proxy commands	The chapter "Authentication Proxy Commands" in the *Cisco IOS Security Command Reference*, Release 12.2
Information on adding HTTPS support to the Cisco IOS web server	*Secure HTTP (HTTPS)*, Cisco IOS Release 12.1(11b)E feature module
Information on configuring and obtaining a CA certificate.	*Trustpoint CLI*, Cisco IOS Release 12.2(8)T feature module

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

RFCs[1]	Title
RFC 1945	*Hyptertext Transfer Protocol — HTTP/ 1.0*
RFC 2616	*Hyptertext Transfer Protocol — HTTP/ 1.1*

1. Not all supported RFCs are listed.

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

This feature uses no new or modified commands. To see the command pages for the commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

Glossary

ACL—access control list. An ACL is a list kept by routers to control access to or from the router for a number of services (for example, to prevent packets with a certain IP address from leaving a particular interface on the router).

Cisco IOS Firewall—The Cisco IOS Firewall is a protocol that provides advanced traffic filtering functionality and can be used as an integral part of your network's firewall.

The Cisco IOS Firewall creates temporary openings in access lists at firewall interfaces. These openings are created when specified traffic exits your internal network through the firewall. The openings allow returning traffic (that would normally be blocked) and additional data channels to enter your internal network back through the firewall. The traffic is allowed back through the firewall only if it is part of the same session as the original traffic that triggered the Cisco IOS Firewall when exiting through the firewall.

firewall—A firewall is a networking device that controls access to the network assets of your organization. Firewalls are positioned at the entrance points into your network. If your network has multiple entrance points, you must position a firewall at each point to provide effective network access control.

The most basic function of a firewall is to monitor and filter traffic. Firewalls can be simple or elaborate, depending on your network requirements. Simple firewalls are usually easier to configure and manage. However, you might require the flexibility of a more elaborate firewall.

HTTPS—HTTP over SSL. HTTPS is client communication with a server by first negotiating an SSL connection and then transmiting the HTTP protocol data over the SSL application data channel.

SSL—Secure Socket Layer. SSL is encryption technology for the web used to provide secure transactions, such as the transmission of credit card numbers for e-commerce.

Note Refer to the *Internetworking Terms and Acronyms* for terms not included in this glossary.

Firewall Authentication Proxy for FTP and Telnet Sessions

Before the introduction of the Firewall Authentication Proxy for FTP and Telnet Sessions feature, users could enable only HTTP when configuring authentication proxy. This feature introduces support for FTP and Telnet, providing users with three protocol options when configuring authentication proxy.

Feature Specifications for the Firewall Authentication Proxy for FTP and Telnet Sessions Feature

Feature History	
Release	**Modification**
12.3(1)	This feature was introduced.
Supported Platforms	
For platforms supported in Cisco IOS Release 12.3(1), consult Cisco Feature Navigator.	

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Restrictions for Firewall Authentication Proxy for FTP and Telnet Sessions, page 818
- Information About Firewall Authentication Proxy for FTP and Telnet Sessions, page 818
- How to Configure FTP or Telnet Authentication Proxy, page 823
- Configuration Examples for FTP and Telnet Authentication Proxy, page 828
- Additional References, page 831
- Command Reference, page 833

Restrictions for Firewall Authentication Proxy for FTP and Telnet Sessions

- Authentication proxy is an IP-only feature; thus, it comes with only -o3 images.
- "proxyacl#<n>" is the only supported attribute in the authentication, authorization, and accounting (AAA) server's user configuration.
- Authentication proxy is subjected only to the traffic that passes through the router; traffic that is destined for the router continues to be authenticated by the existing authentication methods that are provided by Cisco IOS.

Information About Firewall Authentication Proxy for FTP and Telnet Sessions

To configure the Authentication Proxy for FTP and Telnet Sessions feature, you must understand the following concepts:

- Feature Design for FTP and Telnet Authentication Proxy, page 818
- Absolute Timeout, page 823

Feature Design for FTP and Telnet Authentication Proxy

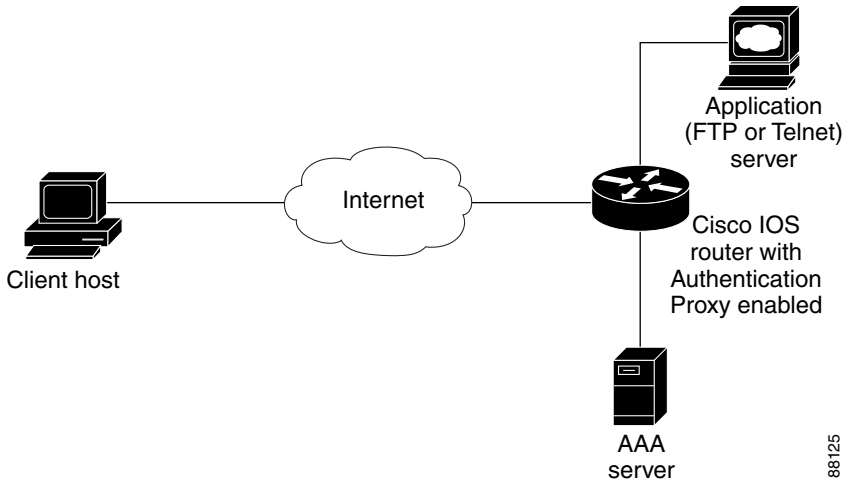
Authentication proxy for FTP and Telnet Sessions functions like authentication proxy for HTTP; that is, FTP and Telnet are independent components in the Cisco IOS software and can be enabled or disabled on the interface of an unauthenticated host.

Many of the authentication proxy for FTP or Telnet functions are similar to those used with HTTP, such as the interaction between the authentication proxy router and the AAA server during authentication. However, because of protocol differences, FTP and Telnet login methods are different from HTTP.

FTP and Telnet Login Methods

Figure 50 displays a typical authentication proxy topology.

Figure 50 Typical Authentication Proxy Topology



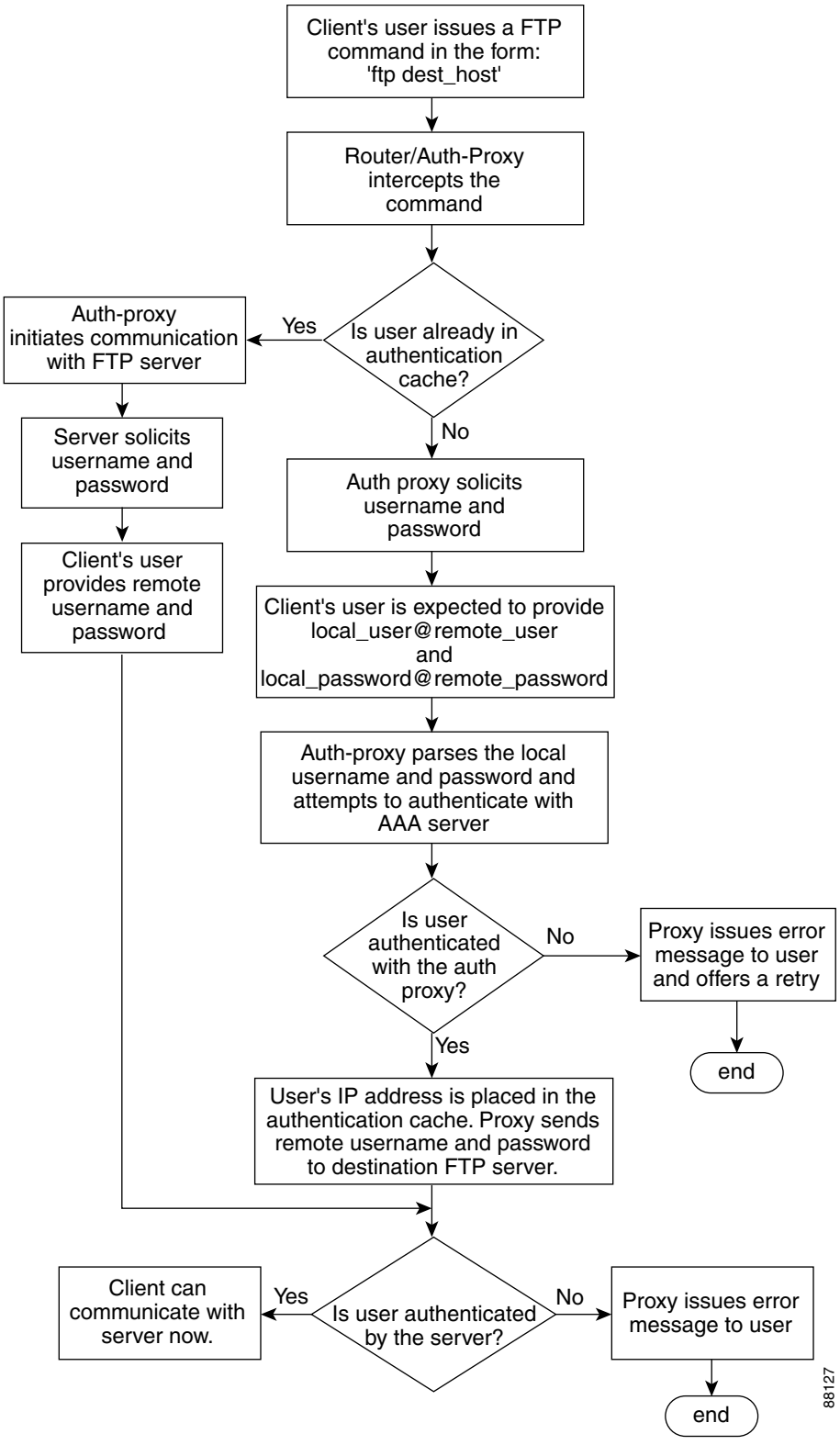
Just as with HTTP, the authentication proxy router intercepts traffic that is sent from the client host. Upon receiving a FTP or Telnet packet, the router will look into its authentication cache to check whether the client host has already been authenticated. If it has been authenticated, the router will forward the client host's traffic to the FTP or Telnet server for additional authentication. If the IP address of the client host is not in the cache of the router, the router will try to authenticate the client host with the AAA server using the username and password of the router.

FTP Login

For FTP login, the client host will be prompted (by the authentication proxy router) for the username and password of the router; the client must respond with the username and password in the following format: "login: proxy_username@ftp_username" and "password: proxy_passwd@ftp_passwd:". The authentication proxy will use the proxy username and password to verify the client's profile against the AAA server's user database. After the client is successfully authenticated with the AAA server, the authentication proxy will pass the FTP (remote) username and password to the FTP server (destination server) for the application server authentication.

A flow chart that depicts an overview of the FTP authentication proxy process is shown in Figure 51.

Figure 51 FTP Authentication Proxy Overview

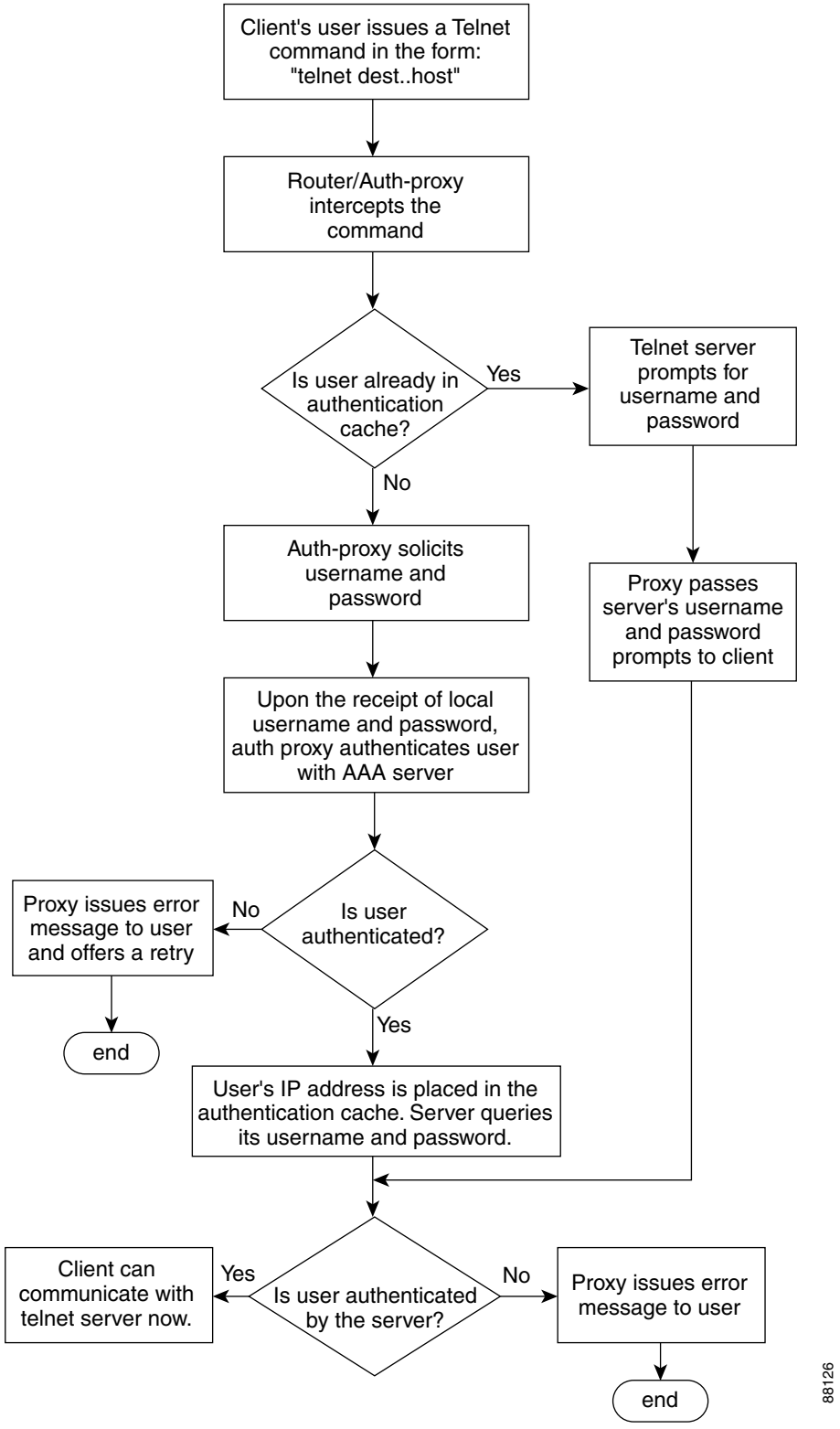


Telnet Login

For Telnet login, the client host will be prompted (by the authentication proxy router) for the username, followed by the password; the client must respond with the username and password in the following format: “login: proxy_username:” and “password: proxy_passwd):”. The username and password will be verified against the AAA server’s user database. After the client is successfully authenticated with the AAA server, the Telnet server (destination server) will prompt the client for the username and password of the Telnet server.

A flow chart that depicts an overview of the Telnet authentication proxy process is shown in Figure 52.

Figure 52 Telnet Authentication Proxy Overview



If authentication with the AAA server fails, the proxy will inform the client accordingly. With Telnet, the proxy does not have any interest in the Telnet server's username and password. If the client is authenticated with the AAA server but fails with the Telnet server, the client will not have to authenticate with the AAA server the next time he or she logs into the network; the client's IP address will be stored in the authentication cache. The client will have to authenticate only with the Telnet server.

Note With FTP, the client must always reenter the local and remote username and password combination every time he or she tries to log into the network—regardless of a successful AAA server authentication.

Absolute Timeout

An absolute timeout value has been added to allow users to configure a window during which the authentication proxy on the enabled interface is active. After the absolute timer expires, the authentication proxy will be disabled regardless of any activity. The absolute timeout value can be configured per protocol (via the **ip auth-proxy name** command) or globally (via the **ip auth-proxy** command). The default value of the absolute timeout is zero; that is, the absolute timer is turned off by default, and the authentication proxy is enabled indefinitely and is subject only to the timeout specified by the **inactivity-timer** keyword.

Note The **inactivity-timer** keyword deprecates the **auth-cache-time** keyword in the **ip auth-proxy name** and the **ip auth-proxy** commands.

How to Configure FTP or Telnet Authentication Proxy

To enable FTP or Telnet authentication proxy, you must enable AAA services, configure the FTP or Telnet server, and enable authentication proxy. This section contains the following procedures:

- Configuring AAA, page 823
- Configuring the Authentication Proxy, page 825
- Verifying FTP or Telnet Authentication Proxy, page 827
- Monitoring and Maintaining FTP or Telnet Authentication Proxy Sessions, page 828

Configuring AAA

To use authentication proxy, you must configure a AAA server for authentication. The authentication proxy service of the AAA server must also be configured for authorization. To configure these tasks, perform the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication login default group tacacs+ group radius**

5. **aaa authorization auth-proxy default** [[**group tacacs+**] [**group radius**]]
6. **aaa authorization exec default** [**group tacacs+**] [**group radius**]
7. **aaa accounting auth-proxy default stop-only** [**group tacacs+**] [**group radius**]
8. **access-list** *access-list-number* {**permit** | **deny**} {**tcp** | **ip** | **icmp**} **host** *source* **eq** *tacacs* **host** *destination*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router (config)# aaa new-model`	Enables the AAA functionality on the router.
Step 4	**aaa authentication login default group tacacs+ group radius** **Example:** `Router (config)# aaa authentication login default group tacacs+ group radius`	Defines the list of authentication methods at login.
Step 5	**aaa authorization auth-proxy default** [[**group tacacs+**] [**group radius**]] **Example:** `Router (config)# aaa authorization auth-proxy default group tacacs+ group radius`	Uses the **auth-proxy** keyword to enable authorization proxy for AAA methods.
Step 6	**aaa authorization exec default** [**group tacacs+**] [**group radius**] **Example:** `Router (config)# aaa authorization exec default group tacacs+ group radius`	Enables authorization for TACACS+ and RADIUS.

	Command or Action	Purpose
Step 7	**aaa accounting auth-proxy default stop-only** [**group tacacs+**] [**group radius**] **Example:** `Router (config)# aaa accounting auth-proxy default stop-only group tacacs+ group radius`	Activates authentication proxy accounting and uses the **auth-proxy** keyword to set up the authorization policy as dynamic access control lists (ACLs) that can be downloaded.
Step 8	**access-list** *access-list-number* {**permit** | **deny**} {**tcp** | **ip** | **icmp**} **host** *source* **eq** *tacacs* **host** *destination* **Example:** `Router (config)# access-list 111 permit tcp host 209.165.200.225 eq tacacs host 209.165.200.254` or `Router (config)# access-list 111 deny ip any any` or `Router (config)#  access-list 111 permit icmp any any`	Creates an ACL entry to allow the AAA server to return traffic to the firewall. The source address is the IP address of the AAA server, and the destination is the IP address of the router interface where the AAA server resides.

What to Do Next

Ensure that your FTP or Telnet server is enabled and that the user credentials of the client (the username and password) are stored in the server's database.

Configuring the Authentication Proxy

To configure the authentication proxy, perform the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip auth-proxy** {**inactivity-timer** *min* | **absolute-timer** *min*}
4. **ip auth-proxy auth-proxy-banner** {**ftp** | **http** | **telnet**} [*banner-text*]
5. **ip auth-proxy name** *auth-proxy-name* {**ftp** | **http** | **telnet**} [**inactivity-timer** *min* | **absolute-timer** *min*] [**list** {*acl* | *acl-name*}]
6. **interface** *type*
7. **ip auth-proxy** *auth-proxy-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip auth-proxy** {**inactivity-timer** *min* | **absolute-timer** *min*} **Example:** `Router (config)# ip auth-proxy inactivity-timer 30`	Sets the global authentication proxy idle timeout values in minutes. • **inactivity-timer** *min*—Specifies the length of time in minutes that an authentication cache entry is managed after a period of inactivity. Enter a value in the range 1 to 2,147,483,647. The default value is 60 minutes. • **absolute-timer** *min*—Specifies a window in which the authentication proxy on the enabled interface is active. Enter a value in the range 1 to 65,535 minutes (45 and a half days). The default value is 0 minutes.
Step 4	**ip auth-proxy auth-proxy-banner** {**ftp** | **http** | **telnet**}[*banner-text*] **Example:** `Router (config)# ip auth-proxy auth-proxy-banner ftp hello`	Optional) Displays the name of the firewall router in the authentication proxy login page. Disabled by default. • **ftp**—Specifies the FTP protocol. • **http**—Specifies the HTTP protocol. • **telnet**—Specifies the Telnet protocol. • *banner-text*—(Optional) A text string that replaces the default banner.
Step 5	**ip auth-proxy name** *auth-proxy-name* {**ftp** | **http** | **telnet**} [**inactivity-timer** *min*] [**absolute-timer** *min*] [**list** {*acl* | *acl-name*}] **Example:** `Router (config)# ip auth-proxy name ftp_list1 ftp absolute-timer 60 ftp list 102`	Configures authentication proxy on an interface. • **ftp**—Specifies FTP to trigger that authentication proxy. • **http**—Specifies HTTP to trigger that authentication proxy. • **telnet**—Specifies Telnet to trigger that authentication proxy. • **inactivity-timer** *min*—Overrides global authentication proxy cache timer for a specific authentication proxy name. • **absolute-timer** *min*— Overrides the global value specified via the **ip auth-proxy** command. • **list** {*acl* | *acl-name*}—Specifies a standard (1–99), extended (1–199), or named access list to use with the authentication proxy.

	Command or Action	Purpose
Step 6	**interface** *type* **Example:** `Router (config)# interface e0`	Enters interface configuration mode by specifying the interface type on which to apply the authentication proxy.
Step 7	**ip auth-proxy** *auth-proxy-name* **Example:** `Router(config-if)# ip auth-proxy authproxyrule`	In interface configuration mode, applies the named authentication proxy rule at the interface. This command enables the authentication proxy rule with that name.

Verifying FTP or Telnet Authentication Proxy

To verify your FTP or Telnet authentication proxy configuration, perform the following optional steps:

SUMMARY STEPS

1. **enable**
2. **show ip auth-proxy configuration**
3. **show ip auth-proxy cache**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**show ip auth-proxy configuration** **Example:** `Router# show ip auth-proxy configuration`	Displays the current authentication proxy configuration.
Step 3	**show ip auth-proxy cache** **Example:** `Router# show ip auth-proxy cache`	Displays the list of user authentication entries. The authentication proxy cache lists the host IP address, the source port number, the timeout value for the authentication proxy, and the state of the connection. If the authentication proxy state is ESTAB or INTERCEPT, the user authentication was successful.

Monitoring and Maintaining FTP or Telnet Authentication Proxy Sessions

To monitor FTP or Telnet authentication proxy sessions, perform the following optional steps:

SUMMARY STEPS

1. **enable**
2. **debug ip auth-proxy** {**detailed** | **ftp** | **function-trace** | **object-creation** | **object-deletion** | **telnet** | **timers**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**debug ip auth-proxy** {**detailed** | **ftp** | **function-trace** | **object-creation** | **object-deletion** | **telnet** | **timers**} **Example:** `Router# debug ip auth-proxy ftp`	Displays the authentication proxy configuration information on the router.

Configuration Examples for FTP and Telnet Authentication Proxy

This section provides the following configuration examples:

- Authentication Proxy Configuration Example, page 828
- AAA Server User Profile Examples, page 829

Authentication Proxy Configuration Example

The following example shows how to configure your router for authentication proxy:

```
aaa new-model
aaa authentication login default group tacacs+
aaa authorization exec default group tacacs+
aaa authorization auth-proxy default group tacacs+
enable password lab
!
ip inspect name pxy_test ftp
ip auth-proxy name pxy auth-cache-time 1
!
interface Ethernet0/0
 ip address 209.165.200.225 255.255.255.224
 ip access-group 105 in
 no ip directed-broadcast
 ip inspect pxy_test in
 ip auth-proxy pxy
```

```
 no shut
!
interface Ethernet0/1
 ip address 209.165.200.225 255.255.255.224
 ip access-group 102 in
 no ip directed-broadcast
no shut
!
ip http authentication aaa
!
access-list 102 permit any
access-list 102 permit tcp host 209.165.200.234 eq tacacs any
access-list 102 deny   tcp any any
access-list 102 deny   udp any any
access-list 102 permit ip any any
access-list 105 permit tcp any any eq www
access-list 105 permit ip any any
access-list 105 deny   tcp any any
access-list 105 deny   udp any any
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
tacacs-server host 209.165.200.234
tacacs-server key cisco
!
line con 0
 transport input none
 login authentication special
line aux 0
line vty 0 4
 password lab
```

AAA Server User Profile Examples

This section includes examples of the authentication proxy user profile entries on the AAA servers. The “proxyacl” entries define the user access privileges. After the user has successfully used the authentication proxy to log in, these entries are transferred to the firewall router. Each entry in the profile must specify “permit” access for the service or application. The source address in each entry is set to “any”, which is replaced with the IP address of the authenticating host when the profile is downloaded to the firewall. The privilege level must be set to 15 for all AAA users.

This section contains the following examples:

- TACACS+ User Profiles Example
- Livingston RADIUS User Profiles Example
- Ascend RADIUS User Profiles Example

TACACS+ User Profiles Example

The following example are sample TACACS+ user profiles:

```
default authorization = permit
key = cisco
user = http_1 {
  default service = permit
      login = cleartext test
         service = exec
        {
```

```
                priv-lvl = 15
                inacl#4="permit tcp any host 209.165.200.234 eq 23"
                inacl#5="permit tcp any host 209.165.200.234 eq 20"
                inacl#6="permit tcp any host 209.165.200.234 eq 21"
                inacl#3="deny -1"

        }
      service = auth-proxy
          {
              priv-lvl=15
              proxyacl#4="permit tcp any host 209.165.201.1 eq 23"
              proxyacl#5="permit tcp any host 209.165.201.1 eq 20"
              proxyacl#6="permit tcp any host 209.165.201.1 eq 21"
              proxyacl#7="permit tcp any host 209.165.201.1 eq 25"
          }

}
user = http {
          login = cleartext test
      service = auth-proxy
         {
              priv-lvl=15
              proxyacl#4="permit tcp any host 209.165.201.1 eq 23"
              proxyacl#5="permit tcp any host 209.165.201.1 eq 20"
              proxyacl#6="permit tcp any host 209.165.201.1 eq 21"
          }
}
 user = proxy_1 {
          login = cleartext test
      service = auth-proxy
          {
              priv-lvl=14
        }
}

user = proxy_3 {
          login = cleartext test
      service = auth-proxy
          {
              priv-lvl=15
        }
}
```

Livingston RADIUS User Profiles Example

The following examples are sample user profiles for the Livingston RADIUS server:

```
#--------------- Proxy user ---------------------------------

http                    Password = "test" User-Service-Type=Outbound-User
       cisco-avpair = "auth-proxy:priv-lvl=15",
       cisco-avpair = "auth-proxy:proxyacl#3=permit tcp any any eq 23"

http_1                     Password = "test"
         User-Service-Type = Shell-User,
         User-Service-Type=Dialout-Framed-User,
         cisco-avpair = "shell:priv-lvl=15",
         cisco-avpair = "shell:inacl#4=permit tcp any host 209.165.200.234
eq 23
         cisco-avpair = "auth-proxy:priv-lvl=15",
         cisco-avpair = "auth-proxy:proxyacl#3=permit tcp any any eq 23"
```

```
http_fail               Password = "test" User-Service-Type=Outbound-User
       cisco-avpair = "auth-proxy:priv-lvl=14",
       cisco-avpair = "auth-proxy:proxyacl#3=permit tcp any any eq 23"

proxy  Password = "cisco" User-Service-Type=Outbound-User       cisco-avpair =
"auth-proxy:proxyacl#4=permit tcp any any eq 20"
```

Ascend RADIUS User Profiles Example

The following examples are sample user profiles for the Ascend RADIUS server:

```
#--------------- Proxy user ---------------------------------

http                   Password = "test" User-Service=Dialout-Framed-User
       cisco-avpair = "auth-proxy:priv-lvl=15",
       cisco-avpair = "auth-proxy:proxyacl#3=permit tcp any any eq 23"

http_2                   Password = "test"
User-Service=Dialout-Framed-User
       cisco-avpair = "auth-proxy:priv-lvl=15",
       cisco-avpair = "auth-proxy:proxyacl#3=permit tcp any any eq 23",
       cisco-avpair = "shell:inacl#4=permit tcp any host 209.165.200.234 eq 25"

http_1                    Password = "test"
         User-Service=Dialout-Framed-User,
         cisco-avpair = "shell:inacl#4=permit tcp any host 209.165.200.234 eq 23",
         cisco-avpair = "auth-proxy:priv-lvl=15",
         cisco-avpair = "auth-proxy:proxyacl#3=permit tcp any any eq 23"

http_fail               Password = "test" User-Service=Dialout-Framed-User
       cisco-avpair = "auth-proxy:priv-lvl=14",
       cisco-avpair = "auth-proxy:proxyacl#3=permit tcp any any eq 23"

        cisco-avpair = "auth-proxy:proxyacl#1=permit tcp any any eq 23",
       cisco-avpair = "auth-proxy:proxyacl#3=permit tcp any any eq ftp",
       cisco-avpair = "auth-proxy:proxyacl#4=permit tcp any any eq 20"


#-------------------------------------------------------------

proxy Password = "cisco" User-Service = Dialout-Framed-User

       cisco-avpair = "auth-proxy:priv-lvl=15",

       cisco-avpair = "auth-proxy:priv-lvl=15",
       cisco-avpair = "auth-proxy:proxyacl#1=permit tcp any any eq 26",
       cisco-avpair = "auth-proxy:proxyacl#3=permit tcp any any eq ftp",
```

Additional References

The following sections provide additional references related to the Firewall Authentication Proxy for FTP and Telnet Sessions feature:

- Related Documents, page 832
- Standards, page 832
- MIBs, page 832

- RFCs, page 832
- Technical Assistance, page 833

Related Documents

Related Topic	Document Title
Additional authentication proxy configuration tasks	The chapter "Configuring Authentication Proxy" in the *Cisco IOS Security Configuration Guide*, Release 12.3
Additional authentication proxy commands	*Cisco IOS Security Command Reference*, Release 12.3
RADIUS and TACACS+ configuration information	The section "Security Server Protocols" in the *Cisco IOS Security Configuration Guide*, Release 12.3
RADIUS and TACACS+ attribute information	The chapters "RADIUS Attributes" and "TACACS+ Attribute-Value Pairs" in the *Cisco IOS Security Configuration Guide*, Release 12.3
Additional authentication proxy information	*Firewall Support of HTTPS Authentication Proxy*, Cisco IOS Release 12.2(15)T feature module

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **debug ip auth-proxy**
- **ip auth-proxy**
- **ip auth-proxy auth-proxy-banner**
- **ip auth-proxy name**

Configuring Port to Application Mapping

This chapter describes the Cisco IOS Firewall Port to Application Mapping (PAM) feature. PAM enables CBAC-supported applications to be run on nonstandard ports. Using PAM, network administrators can customize access control for specific applications and services to meet the distinct needs of their networks.

For a complete description of the PAM commands in this chapter, refer to the chapter "Port to Application Mapping Commands" of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Finding Additional Feature Support Information" section on page cxvii in the chapter Using Cisco IOS Software for Release 12.4.

In This Chapter

This chapter contains the following sections:

- About Port to Application Mapping
- PAM Configuration Task List
- Monitoring and Maintaining PAM
- PAM Configuration Examples

About Port to Application Mapping

Port to Application Mapping (PAM) is a feature of the Cisco IOS Firewall feature set. PAM allows you to customize TCP or UDP port numbers for network services or applications. PAM uses this information to support network environments that run services using ports that are different from the registered or well-known ports associated with an application.

Using the port information, PAM establishes a table of default port-to-application mapping information at the firewall. The information in the PAM table enables Context-based Access Control (CBAC) supported services to run on nonstandard ports. Previously, CBAC was limited to inspecting traffic using only the well-known or registered ports associated with an application. Now, PAM allows network administrators to customize network access control for specific applications and services.

PAM also supports host or subnet specific port mapping, which allows you to apply PAM to a single host or subnet using standard access control lists (ACLs). Host or subnet specific port mapping is done using standard ACLs.

This section contains the following sections:

- How PAM Works
- System-Defined Port Mapping
- PAM and CBAC
- When to Use PAM

How PAM Works

PAM generates a table of information that identifies specific applications with specific TCP or UDP port information. When the firewall router first starts up, the PAM table is populated with system-defined mapping information. As you customize the mapping information, the PAM table is modified with the new information. The information in the PAM table serves as the default port mapping for traffic passing through the firewall.

PAM works with CBAC to identify the applications associated with various port numbers, including services running on non-standard ports, as it inspect traffic passing through the firewall. Previously, CBAC was limited to inspecting traffic using only the well-known or registered ports associated with an application.

Entries in the PAM table provide three types of mapping information:

- System-Defined Port Mapping
- User-Defined Port Mapping
- Host-Specific Port Mapping

System-Defined Port Mapping

PAM creates a table, or database, of system-defined mapping entries using the well-known or registered port mapping information set up during the system start-up. The system-defined entries comprise all the services supported by CBAC, which requires the system-defined mapping information to function properly. The system-defined mapping information cannot be deleted or changed; that is, you cannot map HTTP services to port 21 (FTP) or FTP services to port 80 (HTTP).

Note You can override the system-defined entries for specific hosts using the PAM host-specific option. Refer to the section "Host-Specific Port Mapping" in this chapter.

Table 42 lists the default system-defined services and applications in the PAM table.

Table 42 System-Defined Port Mapping

Application Name	Well-Known or Registered Port Number	Protocol Description
cuseeme	7648	CU-SeeMe Protocol
exec	512	Remote Process Execution
ftp	21	File Transfer Protocol (control port)

Table 42 System-Defined Port Mapping (continued)

Application Name	Well-Known or Registered Port Number	Protocol Description
http	80	Hypertext Transfer Protocol
h323	1720	H.323 Protocol (for example, MS NetMeeting, Intel Video Phone)
login	513	Remote login
mgcp	2427	Media Gateway Control Protocol
msrpc	135	Microsoft Remote Procedure Call
netshow	1755	Microsoft NetShow
real-audio-video	7070	RealAudio and RealVideo
rtsp	8559	Real Time Streaming Protocol
shell	514	Remote command
sip	5060	Session Initiation Protocol
smtp	25	Simple Mail Transfer Protocol
sqlnet	1521	SQL-NET
streamworks	1558	StreamWorks Protocol
sunrpc	111	SUN Remote Procedure Call
telnet	23	Telnet
tftp	69	Trivial File Transfer Protocol
vdolive	7000	VDOLive Protocol

This section has the following sections:

- User-Defined Port Mapping
- Host-Specific Port Mapping

User-Defined Port Mapping

Network services or applications that use non-standard ports require user-defined entries in the PAM table. For example, your network might run HTTP services on the non-standard port 8000 instead of on the system-defined default port (port 80). In this case, you can use PAM to map port 8000 with HTTP services. If HTTP services run on other ports, use PAM to create additional port mapping entries. After you define a port mapping, you can overwrite that entry at a later time by simply mapping that specific port with a different application.

Note If you try to map an application to a system-defined port, a message appears that warns you of a mapping conflict.

User-defined port mapping information can also specify a range of ports for an application by establishing a separate entry in the PAM table for each port number in the range.

User-defined entries are saved with the default mapping information when you save the router configuration.

Host-Specific Port Mapping

User-defined entries in the mapping table can include host-specific mapping information, which establishes port mapping information for specific hosts or subnets. In some environments, it might be necessary to override the default port mapping information for a specific host or subnet.

With host-specific port mapping, you can use the same port number for different services on different hosts. This means that you can map port 8000 with HTTP services for one host, while mapping port 8000 with Telnet services for another host.

Host-specific port mapping also allows you to apply PAM to a specific subnet when that subnet runs a service that uses a port number that is different from the port number defined in the default mapping information. For example, hosts on subnet 192.168.21.0 might run HTTP services on non-standard port 8000, while other traffic through the firewall uses the default port for HTTP services, which is port 80.

Host-specific port mapping allows you to override a system-defined entry in the PAM table. For example, if CBAC finds an entry in the PAM table that maps port 25 (the system-defined port for SMTP) with HTTP for a specific host, CBAC identifies port 25 as HTTP protocol traffic on that host.

Note If the host-specific port mapping information is the same as an existing system-defined or user-defined default entries, host-specific port changes have no effect.

PAM and CBAC

CBAC uses the information in the PAM table to identify a service or application from traffic flowing through the firewall. With PAM, CBAC can associate non-standard port numbers with specific protocols. For example, if you use PAM to map port 8000 with HTTP services, CBAC can determine that traffic using port 8000 is an HTTP application.

When to Use PAM

Here are a few examples of when you might want to use PAM:

- Use PAM to apply a non-standard port numbers for a service or application.
- Use PAM when a specific host or subnet uses a port number for an application that is different than the default port number established in the PAM table.
- Use PAM when different hosts use the same port number for different applications.

PAM Configuration Task List

See the following sections for PAM configuration tasks. Each task in the list indicates if it is optional or required:

- Configuring Standard ACLs (Optional)
- Configuring PAM (Required)
- Verifying PAM (Optional)

Configuring Standard ACLs

If you require PAM for a specific host or subnet, use the **access-list** (standard) command in global configuration mode to define an ACL:

Command	Purpose
`Router(config)# access-list access-list-number permit source [source-wildcard]`	(Optional) Creates a standard ACL that defines the specific host or subnet for host-specific PAM. For complete information on access-list command, refer to the *Cisco IOS IP Command Reference, Volume 1 of 3: Addressing and Services*.

Configuring PAM

To configure PAM, use the **ip port-map** command in global configuration mode:

Command	Purpose
`Router(config)# ip port-map appl_name port port_num [list acl_num]`	Establishes a port mapping entry using the TCP or UDP port number and the application name. (Optional) Use the list option to associate this port mapping to the specific hosts in the ACL. (PAM uses standard access lists only.) If an access list is included, the hosts defined in that ACL have the application *appl_name* running on port *port_num*.

Verifying PAM

To verify the port mapping information, enter the **show ip port-map** command in privileged EXEC mode and review the entries:

```
Router# show ip port-map
```

This command displays all entries in the PAM table, including the system-defined entries.

For PAM configuration examples using the commands in this chapter, refer to the "PAM Configuration Examples" section at the end of this chapter.

Monitoring and Maintaining PAM

The following commands can be used to monitor and maintain PAM:

Command	Purpose
`Router# show ip port-map [appl_name | port port_num]`	Displays the port mapping information, including the system-defined entries. Include the application name to display a list of entries by application. Include the port number to display the entries by port.
`Router(config)# no ip port-map appl_name port port_num [list acl_num]`	Deletes user-defined port mapping information. This command has no effect on the system-defined port mapping information.

PAM Configuration Examples

The following sections provide PAM configuration examples:

- Mapping an Application to a Non-Standard Port Example
- Mapping an Application with a Port Range Example
- Invalid Port Mapping Entry Example
- Mapping an Application to a Port for a Specific Host Example
- Mapping an Application to a Port for a Subnet Example
- Overriding a System-Defined Port Mapping Example
- Mapping Different Applications to the Same Port Example

Mapping an Application to a Non-Standard Port Example

In this example, non-standard port 8000 is established as the user-defined default port mapping for HTTP services:

```
ip port-map http port 8000
```

Mapping an Application with a Port Range Example

The following PAM entries establish a range of non-standard ports for HTTP services:

```
ip port-map http 8001
ip port-map http 8002
ip port-map http 8003
ip port-map http 8004
```

Invalid Port Mapping Entry Example

This example is not valid because it tries to establish port 21, which is the system-defined default port for FTP, as the user-defined port for HTTP services:

```
ip port-map http port 21
```

Mapping an Application to a Port for a Specific Host Example

In this example, a specific host uses port 8000 for FTP services. ACL 10 identifies the server address (192.168.32.43), while port 8000 is mapped with FTP services.

```
access-list 10 permit 192.168.32.43
ip port-map ftp port 8000 list 10
```

Mapping an Application to a Port for a Subnet Example

In this example, a specific subnet runs HTTP services on port 8080. ACL 50 identifies the subnet, while port 8080 is mapped with HTTP services.

```
access-list 50 permit 192.168.92.0 0.0.0.255
ip port-map http 8080 list 50
```

Overriding a System-Defined Port Mapping Example

In this example, a specific host runs HTTP services on port 25, which is the system-defined port number for SMTP services. This requires a host-specific PAM entry that overrides the system-defined default port mapping for HTTP, which is port 80. ACL 15 identifies the host address (192.168.33.33), while port 25 is mapped with HTTP services.

```
access-list 15 permit 192.168.33.33
ip port-map http port 25 list 15
```

Mapping Different Applications to the Same Port Example

In this example, the same port number is required by different services running on different hosts. Port 8000 is required for HTTP services for host 192.168.3.4, while port 8000 is also required for FTP services for host 192.168.5.6. ACL 10 and ACL 20 identify the specific hosts, while the PAM entries map the ports with the services for each ACL.

```
access-list 10 permit 192.168.3.4
access-list 20 permit 192.168.5.6
ip port-map http port 8000 list 10
ip port-map http ftp 8000 list 20
```

Part 4: IPSec and IKE

Internet Key Exchange for IPSec VPNs

This part consists of the following:

- Call Admission Control for IKE
- Certificate to ISAKMP Profile Mapping
- Encrypted Preshared Key
- Configuring Internet Key Exchange for IPSec VPNs

Call Admission Control for IKE

The Call Admission Control for IKE feature describes the application of Call Admission Control (CAC) to the Internet Key Exchange (IKE) protocol in Cisco IOS. CAC limits the number of simultaneous IKE security associations (SAs) (that is, calls to CAC) that a router can establish.

Feature History for Call Admission Control for IKE

Release	Modification
12.3(8)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for Call Admission Control for IKE, page 848
- Information About Call Admission Control for IKE, page 848
- How to Configure Call Admission Control for IKE, page 849
- Verifying the Call Admission Control for IKE Configuration, page 850
- Configuration Examples for Call Admission Control for IKE, page 851
- Additional References, page 852
- Command Reference, page 853

Prerequisites for Call Admission Control for IKE

- Configure IKE on the router. Refer to the *Cisco IOS Security Configuration Guide*, Release 12.3.

Information About Call Admission Control for IKE

To configure CAC for IKE, you need to understand the following concepts:

- IKE Session, page 848
- Security Association Limit, page 848
- System Resource Usage, page 848

IKE Session

There are two ways to limit the number of IKE SAs that a router can establish to or from another router:

- Configure the absolute IKE SA limit by entering the **crypto call admission limit** command. The router drops new IKE SA requests when the value has been reached.
- Configure the system resource limit by entering the **call admission limit** command. The router drops new IKE SA requests when the specified percentage of system resources is being used.

For information about using these commands, see the "Command Reference" section on page 853.

CAC is applied only to new SAs (that is, when an SA does not already exist between the peers). Every effort is made to preserve existing SAs. Only new SA requests will ever be denied due to a lack of system resources or because the configured IKE SA limit has been reached.

Security Association Limit

An SA is a description of how two or more entities will utilize security services to communicate securely on behalf of a particular data flow. IKE requires and uses SAs to identify the parameters of its connections. IKE can negotiate and establish its own SA. An IKE SA is used by IKE only, and it is bidirectional. An IKE SA can not limit IPsec.

IKE drops SA requests based on a user-configured SA limit. To configure an IKE SA limit, enter the **crypto call admission limit** command. When there is a new SA request from a peer router, IKE determines if the number of active IKE SAs plus the number of SAs being negotiated meets or exceeds the configured SA limit. If the number is greater than or equal to the limit, the new SA request is rejected and a syslog is generated. This log contains the source destination IP address of the SA request.

System Resource Usage

CAC polls a global resource monitor so that IKE knows when the router is running short of CPU cycles or memory buffers. You can configure a resource limit, from 1 to 100, that represents a percentage of system resources. When that percentage of the system resources is being used, IKE drops (will not accept new) SA requests. For example, if you specify a resource limit of 90 percent, IKE stops accepting SA requests when 90 percent of the system resources is being used. To configure the system resource usage, enter the **call admission control** command.

How to Configure Call Admission Control for IKE

This section contains the following procedures:

- Configure the IKE Security Association Limit, page 849 (optional)
- Configure the System Resource Limit, page 850 (optional)

Note You must perform one of the procedures.

Configure the IKE Security Association Limit

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto call admission limit ike sa** *number*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto call admission limit ike sa** *number* **Example:** `Router(config)# crypto call admission limit ike sa 25`	Specifies the maximum number of IKE SAs that the router can establish before IKE begins rejecting new SA requests.
Step 4	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.

Configure the System Resource Limit

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **call admission limit** *percent*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**call admission limit** *percent* **Example:** `Router(config)# call admission limit 90`	Instructs IKE to stop accepting new SA requests (that is, calls for CAC) when the specified percentage of system resources is being used.
Step 4	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.

Verifying the Call Admission Control for IKE Configuration

To verify the CAC for IKE configuration, perform the following steps.

SUMMARY STEPS

1. **show call admission statistics**
2. **show crypto call admission statistics**

DETAILED STEPS

Note For detailed field descriptions of the command output, see the "Command Reference" section on page 853.

Step 18 **show call admission statistics**

Use this command to monitor the global CAC configuration parameters and the behavior of CAC.

```
Router# show call admission statistics

Total Call admission charges: 0, limit 25
Total calls rejected 12, accepted 51
Load metric: charge 0, unscaled 0
```

Step 19 **show crypto call admission statistics**

Use this command to monitor Crypto CAC statistics.

```
Router# show crypto call admission statistics

------------------------------------------------------------
               Crypto Call Admission Control Statistics
------------------------------------------------------------
System Resource Limit: 0    Max IKE SAs 0
Total IKE SA Count:    0    active:     0   negotiating: 0
Incoming IKE Requests: 0    accepted:   0   rejected:    0
Outgoing IKE Requests: 0    accepted:   0   rejected:    0
Rejected IKE Requests: 0    rsrc low:   0   SA limit:    0
```

Configuration Examples for Call Admission Control for IKE

This section provides the following configuration examples:

- Configuring the IKE Security Association Limit: Example, page 851
- Configuring the System Resource Limit: Example, page 851

Configuring the IKE Security Association Limit: Example

The following example shows how to specify that there can be a maximum of 25 SAs before IKE starts rejecting new SA requests:

```
Router(config)# crypto call admission limit ike sa 25
```

Configuring the System Resource Limit: Example

The following example shows how to specify that IKE should drop SA requests when 90 percent of system resources are being used:

```
Router(config)# call admission limit 90
```

Additional References

The following sections provide references related to Call Admission Control for IKE.

Related Documents

Related Topic	Document Title
IKE	• *Cisco IOS Security Command Reference*, Release 12.3T • *Cisco IOS Security Configuration Guide*, Release 12.3

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC #2409	*The Internet Key Exchange*

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **call admission limit**
- **clear crypto call admission statistics**
- **crypto call admission limit**
- **show call admission statistics**
- **show crypto call admission statistics**

Certificate to ISAKMP Profile Mapping

The Certificate to ISAKMP Profile Mapping feature enables you to assign an Internet Security Association and Key Management Protocol (ISAKMP) profile to a peer on the basis of the contents of arbitrary fields in the certificate. In addition, this feature allows you to assign a group name to those peers that are assigned an ISAKMP profile.

Feature History for Certificate to ISAKMP Profile Mapping

Release	Modification
12.3(8)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for Certificate to ISAKMP Profile Mapping, page 855
- Restrictions for Certificate to ISAKMP Profile Mapping, page 856
- Information About Certificate to ISAKMP Profile Mapping, page 856
- How to Configure Certificate to ISAKMP Profile Mapping, page 857
- Configuration Examples for Certificate to ISAKMP Profile Mapping, page 860
- Additional References, page 863
- Command Reference, page 864

Prerequisites for Certificate to ISAKMP Profile Mapping

- You should be familiar with configuring certificate maps.
- You should be familiar with configuring ISAKMP profiles.

Restrictions for Certificate to ISAKMP Profile Mapping

This feature will not be applicable if you use Rivest, Shamir, and Adelman- (RSA-) signature or RSA-encryption authentication without certificate exchange. ISAKMP peers must be configured to do RSA-signature or RSA-encryption authentication using certificates.

Information About Certificate to ISAKMP Profile Mapping

To configure the Certificate to ISAKMP Profile Mapping feature, you should understand the following concepts:

- Certificate to ISAKMP Profile Mapping Overview, page 856
- How Certificate to ISAKMP Profile Mapping Works, page 856
- Assigning an ISAKMP Profile and Group Name to a Peer, page 857

Certificate to ISAKMP Profile Mapping Overview

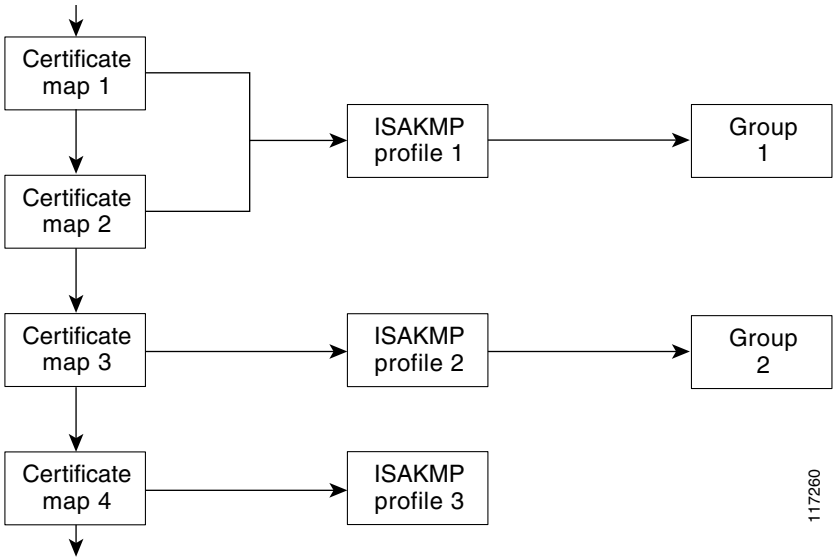
Prior to Cisco IOS Release 12.3(8)T, the only way to map a peer to an ISAKMP profile was as follows. The ISAKMP identity field in the ISAKMP exchange was used for mapping a peer to an ISAKMP profile. When certificates were used for authentication, the ISAKMP identity payload contained the subject name from the certificate. If a certificate authority (CA) did not provide the required group value in the first Organizational Unit (OU) field of a certificate, an ISAKMP profile could not be assigned to a peer.

Effective with Cisco IOS Release 12.3(8)T, a peer can still be mapped as explained above. However, the Certificate to ISAKMP Profile Mapping feature enables you to assign an ISAKMP profile to a peer on the basis of the contents of arbitrary fields in the certificate. You are no longer limited to assigning an ISAKMP profile on the basis of the subject name of the certificate. In addition, this feature allows you to assign a group to a peer to which an ISAKMP profile has been assigned.

How Certificate to ISAKMP Profile Mapping Works

Figure 53 illustrates how certificate maps may be attached to ISAKMP profiles and assigned group names.

Figure 53 Certificate Maps Mapped for Profile Group Assignment



A certificate map can be attached to only one ISAKMP profile although an ISAKMP profile can have several certificate maps attached to it.

Certificate maps provide the ability for a certificate to be matched with a given set of criteria. ISAKMP profiles can bind themselves to certificate maps, and if the presented certificate matches the certificate map present in an ISAKMP profile, the peer will be assigned the ISAKMP profile. If the ISAKMP profile contains a client configuration group name, the same group name will be assigned to the peer. This ISAKMP profile information will override the information in the ID_KEY_ID identity or in the first OU field of the certificate.

Assigning an ISAKMP Profile and Group Name to a Peer

To assign an ISAKMP profile to a peer on the basis of arbitrary fields in the certificate, use the **match certificate** command after the ISAKMP profile has been defined.

To associate a group name with an ISAKMP profile that will be assigned to a peer, use the **client configuration group** command, also after the ISAKMP profile has been defined.

How to Configure Certificate to ISAKMP Profile Mapping

This section contains the following procedures:

- Mapping the Certificate to the ISAKMP Profile, page 858 (required)
- Verifying That the Certificate Has Been Mapped, page 858 (optional)
- Assigning the Group Name to the Peer, page 859 (required)
- Monitoring and Maintaining Your Certificate to ISAKMP Profile Mapping, page 860 (optional)

Mapping the Certificate to the ISAKMP Profile

To map the certificate to the ISAKMP profile, perform the following steps. This configuration will enable you to assign the ISAKMP profile to a peer on the basis of the contents of arbitrary fields in the certificate.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp profile** *profile-name*
4. **match certificate** *certificate-map*

DETAILED STEPS

	Command or Action	Purpose
Step 5	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 6	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 7	**crypto isakmp profile** *profile-name* **Example:** `Router (config)# crypto isakmp profile vpnprofile`	Defines an ISAKMP profile and enters into crypto ISAKMP profile configuration mode.
Step 8	**match certificate** *certificate-map* **Example:** `Router (conf-isa-prof)# match certificate map1`	Accepts the name of a certificate map.

Verifying That the Certificate Has Been Mapped

The following **show** command may be used to verify that the subject name of the certificate map has been properly configured.

SUMMARY

1. **enable**
2. **show crypto ca certificates**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto ca certificates** **Example:** `Router# show crypto ca certificates`	Displays information about your certificate.

Assigning the Group Name to the Peer

To associate a group name with a peer when the peer is mapped to an ISAKMP profile, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp profile** *profile-name*
4. **client configuration group** *group-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp profile** *profile-name* **Example:** `Router (config)# crypto isakmp profile vpnprofile`	Defines an ISAKMP profile and enters into isakmp profile configuration mode.
Step 4	**client configuration group** *group-name* **Example:** `Router (conf-isa-prof)# client configuration group group1`	Accepts the name of a group that will be assigned to a peer when the peer is assigned this crypto ISAKMP profile.

Monitoring and Maintaining Your Certificate to ISAKMP Profile Mapping

To monitor and maintain your certificate to ISAKMP profile mapping, you may use the following **debug** command.

SUMMARY STEPS

1. **enable**
2. **debug crypto isakmp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto isakmp** **Example:** `Router# debug crypto isakmp`	Displays output showing that the certificate has gone through certificate map matching and that the certificate matches the ISAKMP profile. The command may also be used to verify that the peer has been assigned a group.

Configuration Examples for Certificate to ISAKMP Profile Mapping

This section contains the following configuration examples:

- Certificates Mapped to the ISAKMP Profile on the Basis of Arbitary Fields: Example, page 860
- Group Name Assigned to a Peer That Is Associated with an ISAKMP Profile: Example, page 861
- Mapping a Certificate to an ISAKMP Profile Verification: Example, page 861
- Group Name Assigned to a Peer Verfication: Example, page 862

Certificates Mapped to the ISAKMP Profile on the Basis of Arbitary Fields: Example

The following configuration example shows that whenever a certificate contains “ou = green,” the ISAKMP profile “cert_pro” will be assigned to the peer:

```
crypto pki certificate map cert_map 10
 subject-name co ou = green
!
!
crypto isakmp identity dn
crypto isakmp profile cert_pro
   ca trust-point 2315
   ca trust-point LaBcA
```

```
   initiate mode aggressive
   match certificate cert_map
```

Group Name Assigned to a Peer That Is Associated with an ISAKMP Profile: Example

The following example shows that the group "some_group" is to be associated with a peer that has been assigned an ISAKMP profile:

```
crypto isakmp profile id_profile
   ca trust-point 2315
   match identity host domain cisco.com
   client configuration group some_group
```

Mapping a Certificate to an ISAKMP Profile Verification: Example

The following examples show that a certificate has been mapped to an ISAKMP profile. The examples include the configurations for the responder and initiator, **show** command output verifying that the subject name of the certificate map has been configured, and **debug** command output showing that the certificate has gone through certificate map matching and been matched to the ISAKMP profile.

Responder Configuration

```
crypto pki certificate map cert_map 10
! The above line is the certificate map definition.
 subject-name co ou = green
! The above line shows that the subject name must have "ou = green."
!
crypto isakmp profile certpro
! The above line shows that this is the ISAKMP profile that will match if the certificate
of the peer matches cert_map (shown on third line below).
   ca trust-point 2315
   ca trust-point LaBcA
   match certificate cert_map
   initiate mode aggressive
```

Initiator Configuration

```
crypto ca trustpoint LaBcA
 enrollment url http://10.76.82.20:80/cgi-bin/openscep
 subject-name ou=green,c=IN
! The above line ensures that the subject name "ou = green" is set.
 revocation-check none
```

show crypto ca certificates Command Output for the Initiator

```
Router# show crypto ca certificates

Certificate
  Status: Available
  Certificate Serial Number: 21
  Certificate Usage: General Purpose
  Issuer:
    cn=blue-lab CA
    o=CISCO
    c=IN
  Subject:
    Name: Router1.cisco.com
    c=IN
```

```
    ou=green
! The above line is a double check that "ou = green" has been set as the subject name.
    hostname=Router1.cisco.com
  Validity Date:
    start date: 14:34:30 UTC Mar 31 2004
    end   date: 14:34:30 UTC Apr 1 2009
    renew date: 00:00:00 UTC Jan 1 1970
  Associated Trustpoints: LaBcA
```

debug crypto isakmp Command Output for the Responder

```
Router# debug crypto isakmp

6d23h: ISAKMP (0:268435460): received packet from 192.0.0.2 dport 500 sport 500 Global (R)
MM_KEY_EXCH
6d23h: ISAKMP: Main Mode packet contents (flags 1, len 892):
6d23h:           ID payload
6d23h:             FQDN <Router1.cisco.com> port 500 protocol 17
6d23h:           CERT payload
6d23h:           SIG payload
6d23h:           KEEPALIVE payload
6d23h:           NOTIFY payload
6d23h: ISAKMP:(0:4:HW:2):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
6d23h: ISAKMP:(0:4:HW:2):Old State = IKE_R_MM4  New State = IKE_R_MM5

6d23h: ISAKMP:(0:4:HW:2): processing ID payload. message ID = 0
6d23h: ISAKMP (0:268435460): ID payload
        next-payload : 6
        type         : 2
        FQDN name    : Router1.cisco.com
        protocol     : 17
        port         : 500
        length       : 28
6d23h: ISAKMP:(0:4:HW:2):: peer matches *none* of the profiles
6d23h: ISAKMP:(0:4:HW:2): processing CERT payload. message ID = 0
6d23h: ISAKMP:(0:4:HW:2): processing a CT_X509_SIGNATURE cert
6d23h: ISAKMP:(0:4:HW:2): peer's pubkey isn't cached
6d23h: ISAKMP:(0:4:HW:2): OU = green
6d23h: ISAKMP:(0:4:HW:2): certificate map matches certpro profile
! The above line shows that the certificate has gone through certificate map matching and
that it matches the "certpro" profile.
6d23h: ISAKMP:(0:4:HW:2): Trying to re-validate CERT using new profile
6d23h: ISAKMP:(0:4:HW:2): Creating CERT validation list: 2315, LaBcA,
6d23h: ISAKMP:(0:4:HW:2): CERT validity confirmed.
```

Group Name Assigned to a Peer Verfication: Example

The following configuration and debug output show that a group has been assigned to a peer.

Initiator Configuration

```
crypto isakmp profile certpro
   ca trust-point 2315
   ca trust-point LaBcA
   match certificate cert_map
   client configuration group new_group
! The statement on the above line will assign the group "new_group" to any peer that
matches the ISAKMP profile "certpro."
   initiate mode aggressive
!
```

debug crypto isakmp profile Command Output for the Responder

The following debug output example shows that the peer has been matched to the ISAKMP profile named “certpro” and that it has been assigned a group named “new_group.”

```
Router# debug crypto isakmp profile
6d23h: ISAKMP (0:268435461): received packet from 192.0.0.2 dport 500 sport 500 Global (R)
MM_KEY_EXCH
6d23h: ISAKMP: Main Mode packet contents (flags 1, len 892):
6d23h:           ID payload
6d23h:             FQDN <Router1.cisco.com> port 500 protocol 17
6d23h:           CERT payload
6d23h:           SIG payload
6d23h:           KEEPALIVE payload
6d23h:           NOTIFY payload
6d23h: ISAKMP:(0:5:HW:2):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
6d23h: ISAKMP:(0:5:HW:2):Old State = IKE_R_MM4  New State = IKE_R_MM5

6d23h: ISAKMP:(0:5:HW:2): processing ID payload. message ID = 0
6d23h: ISAKMP (0:268435461): ID payload
        next-payload : 6
        type         : 2
        FQDN name    : Router1.cisco.com
        protocol     : 17
        port         : 500
        length       : 28
6d23h: ISAKMP:(0:5:HW:2):: peer matches *none* of the profiles
6d23h: ISAKMP:(0:5:HW:2): processing CERT payload. message ID = 0
6d23h: ISAKMP:(0:5:HW:2): processing a CT_X509_SIGNATURE cert
6d23h: ISAKMP:(0:5:HW:2): peer's pubkey isn't cached
6d23h: ISAKMP:(0:5:HW:2): OU = green
6d23h: ISAKMP:(0:5:HW:2): certificate map matches certpro profile
6d23h: ISAKMP:(0:5:HW:2): Trying to re-validate CERT using new profile
6d23h: ISAKMP:(0:5:HW:2): Creating CERT validation list: 2315, LaBcA,
6d23h: ISAKMP:(0:5:HW:2): CERT validity confirmed.
6d23h: ISAKMP:(0:5:HW:2):Profile has no keyring, aborting key search
6d23h: ISAKMP:(0:5:HW:2): Profile certpro assigned peer the group named new_group
```

Additional References

The following sections provide references related to Certificate to ISAKMP Profile Mapping.

Related Documents

Related Topic	Document Title
Configuring certificate maps	“*Certificate Security Attribute-Based Access Control*,” Release 12.2 T
Configuring ISAKMP profiles	“*VRF-Aware IPSec*,” Release 12.2 T
Security commands	*Cisco IOS Security Command Reference*, Release 12.3 T

Standards

Standards	Title
There are no new or modified standards associated with this feature.	—

MIBs

MIBs	MIBs Link
There are no new or modified MIBs associated with this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
There are no new or modified RFCs associated with this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **client configuration group**
- **match certificate**

Encrypted Preshared Key

The Encrypted Preshared Key feature allows you to securely store plain text passwords in type 6 (encrypted) format in NVRAM.

Feature History for Encrypted Preshared Key

Release	Modification
12.3(2)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Restrictions for Encrypted Preshared Key, page 865
- Information About Encrypted Preshared Key, page 866
- How to Configure an Encrypted Preshared Key, page 867
- Configuration Examples for Encrypted Preshared Key, page 875
- Where to Go Next, page 877
- Additional References, page 877
- Command Reference, page 878

Restrictions for Encrypted Preshared Key

- Old ROM monitors (ROMMONs) and boot images cannot recognize the new type 6 passwords. Therefore, errors are expected if you boot from an old ROMMON.
- For Cisco 836 routers, please note that support for Advanced Encryption Standard (AES) is available only on IP plus images.

Information About Encrypted Preshared Key

Before Using the Encrypted Preshared Key feature, you should understand the following concepts:

- Using the Encrypted Preshared Key Feature to Securely Store Passwords, page 866
- How to Configure an Encrypted Preshared Key, page 867

Using the Encrypted Preshared Key Feature to Securely Store Passwords

Using the Encrypted Preshared Key feature, you can securely store plain text passwords in type 6 format in NVRAM using a command-line interface (CLI). Type 6 passwords are encrypted. Although the encrypted passwords can be seen or retrieved, it is difficult to decrypt them to find out the actual password. Use the **key config-key password-encryption** command with the **password encryption aes** command to configure and enable the password (symmetric cipher AES is used to encrypt the keys). The password (key) configured using the **config-key password-encryption** command is the master encryption key that is used to encrypt all other keys in the router.

If you configure the **password encryption aes** command without configuring the **key config-key password-encryption** command, the following message is printed at startup or during any nonvolatile generation (NVGEN) process, such as when the **show running-config** or **copy running-config startup-config** commands have been configured:

```
"Can not encrypt password. Please configure a configuration-key with 'key config-key'"
```

Changing a Password

If the password (master key) is changed, or reencrypted, using the **key config-key password-encryption** command), the list registry passes the old key and the new key to the application modules that are using type 6 encryption.

Deleting a Password

If the master key that was configured using the **key config-key password-encryption** command is deleted from the system, a warning is printed (and a confirm prompt is issued) that states that all type 6 passwords will become useless. As a security measure, after the passwords have been encrypted, they will never be decrypted in the Cisco IOS software. However, passwords can be reencrypted as explained in the previous paragraph.

Caution If the password configured using the **key config-key password-encryption** command is lost, it cannot be recovered. The password should be stored in a safe location.

Unconfiguring Password Encryption

If you later unconfigure password encryption using the **no password encryption aes** command, all existing type 6 passwords are left unchanged, and as long as the password (master key) that was configured using the **key config-key password-encryption** command exists, the type 6 passwords will be decrypted as and when required by the application.

Storing Passwords

Because no one can "read" the password (configured using the **key config-key password-encryption** command), there is no way that the password can be retrieved from the router. Existing management stations cannot "know" what it is unless the stations are enhanced to include this key somewhere, in which case the password needs to be stored securely within the management system. If configurations are stored using TFTP, the configurations are not standalone, meaning that they cannot be loaded onto a router. Before or after the configurations are loaded onto a router, the password must be manually added (using the **key config-key password-encryption** command). The password can be manually added to the stored configuration but is not recommended because adding the password manually allows anyone to decrypt all passwords in that configuration.

Configuring New or Unknown Passwords

If you enter or cut and paste cipher text that does not match the master key, or if there is no master key, the cipher text is accepted or saved, but an alert message is printed. The alert message is as follows:

```
"ciphertext>[for username bar>] is incompatible with the configured master key."
```

If a new master key is configured, all the plain keys are encrypted and made type 6 keys. The existing type 6 keys are not encrypted. The existing type 6 keys are left as is.

If the old master key is lost or unknown, you have the option of deleting the master key using the **no key config-key password-encryption** command. Deleting the master key using the **no key config-key password-encryption** command causes the existing encrypted passwords to remain encrypted in the router configuration. The passwords will not be decrypted.

Enabling the Encrypted Preshared Key

The **password encryption aes** command is used to enable the encrypted password.

How to Configure an Encrypted Preshared Key

This section contains the following procedures:

- Configuring an Encrypted Preshared Key, page 867 (required)
- Monitoring Encrypted Preshared Keys, page 869 (optional)
- Configuring an ISAKMP Preshared Key, page 870 (optional)
- Configuring an ISAKMP Preshared Key in ISAKMP Keyrings, page 871 (optional)
- Configuring ISAKMP Aggressive Mode, page 872 (optional)
- Configuring a Unity Server Group Policy, page 873 (optional)
- Configuring an Easy VPN Client, page 874 (optional)

Configuring an Encrypted Preshared Key

To configure an encrypted preshared key, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key config-key password-encryption** [*text*]
4. **password encryption aes**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**key config-key password-encryption** [*text*] **Example:** `Router (config)# key config-key password-encryption`	Stores a type 6 encryption key in private NVRAM. • If you want to key in interactively (using the enter key) and an encrypted key already exists, you will be prompted for the following: Old key, New key, and Confirm key. • If you want to key in interactively but an encryption key is not present, you will be prompted for the following: New key and Confirm key. • If you want to remove the password that is already encrypted, you will see the following prompt: "WARNING: All type 6 encrypted keys will become unusable. Continue with master key deletion? [yes/no]:".
Step 4	**password encryption aes** **Example:** `Router (config)# password-encryption aes`	Enables the encrypted preshared key.

Troubleshooting Tips

If you see the warning message "ciphertext >[for username bar>] is incompatible with the configured master key," you have entered or cut and pasted cipher text that does not match the master key or there is no master key. (The cipher text will be accepted or saved.) The warning message will allow you to locate the broken configuration line or lines.

Monitoring Encrypted Preshared Keys

To get logging output for encrypted preshared keys, perform the following steps.

1. **enable**
2. **password logging**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**password logging** **Example:** `Router# password logging`	Provides a log of debugging output for a type 6 password operation.

Examples

The following **password logging** debug output shows that a new master key has been configured and that the keys have been encrypted with the new master key:

```
Router (config)# key config-key password-encrypt
New key:
Confirm key:
Router (config)#
01:40:57: TYPE6_PASS: New Master key configured, encrypting the keys with
the new master keypas

Router (config)# key config-key password-encrypt
Old key:
New key:
Confirm key:
Router (config)#
01:42:11: TYPE6_PASS: Master key change heralded, re-encrypting the keys
with the new master key
01:42:11: TYPE6_PASS: Mac verification successful
01:42:11: TYPE6_PASS: Mac verification successful
01:42:11: TYPE6_PASS: Mac verification successful
```

What To Do Next

You can perform any of the following procedures. Each procedure is independent of the others.

- Configuring an ISAKMP Preshared Key, page 870
- Configuring an ISAKMP Preshared Key in ISAKMP Keyrings, page 871
- Configuring ISAKMP Aggressive Mode, page 872
- Configuring a Unity Server Group Policy, page 873
- Configuring an Easy VPN Client, page 874

Configuring an ISAKMP Preshared Key

To configure an ISAKMP preshared key, perform the following procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp key** *keystring* **address** *peer-address*
4. **crypto isakmp key** *keystring* **hostname** *hostname*

DETAILED STEPS

	Command	Description
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp key** *keystring* **address** *peer-address* **Example:** `Router (config)# crypto isakmp key cisco address 10.2.3.4`	Configures a preshared authentication key. • The *peer-address* argument specifies the IP address of the remote peer.
Step 4	**crypto isakmp key** *keystring* **hostname** *hostname* **Example:** `Router (config)# crypto isakmp key foo hostname foo.com`	Configures a preshared authentication key. • The *hostname* argument specifies the fully qualified domain name (FQDN) of the peer.

Example

The following sample output shows that an encrypted preshared key has been configured:

```
crypto isakmp key 6 _Hg[^^ECgLGGPF^RXTQfDDWQ][YAAB address 10.2.3.4
crypto isakmp key 6 `eR\eTRaKCUZPYYQfDgXRWi_AAB hostname foo.com
```

Configuring an ISAKMP Preshared Key in ISAKMP Keyrings

To configure an ISAKMP preshared key in ISAKMP keyrings, which are used in IPSec Virtual Route Forwarding (VRF) configurations, perform the following procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto keyring** *keyring-name*
4. **pre-shared-key address** *address* **key** *key*
5. **pre-shared-key hostname** *hostname* **key** *key*

DETAILED STEPS

	Command	Description
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto keyring** *keyring-name* **Example:** `Router (config)# crypto keyring foo`	Defines a crypto keyring to be used during Internet Key Exchange (IKE) authentication and enters keyring configuration mode.
Step 4	**pre-shared-key address** *address* **key** *key* **Example:** `Router (config-keyring)# pre-shared-key address 10.2.3.5 key cisco`	Defines a preshared key to be used for IKE authentication. • The *address* argument specifies the IP address of the remote peer.
Step 5	**pre-shared-key hostname** *hostname* **key** *key* **Example:** `Router (config-keyring)# pre-shared-key hostname foo.com key cisco`	Defines a preshared key to be used for IKE authentication. • The *hostname* argument specifies the FQDN of the peer.

Example

The following s**how-running-config** sample output shows that an encrypted preshared key in ISAKMP keyrings has been configured.

```
crypto keyring foo
  pre-shared-key address 10.2.3.5 key 6 `WHCJYR_Z]GRPF^RXTQfDcfZ]GPAAB
  pre-shared-key hostname foo.com key 6 aE_REHDcOfYCPF^RXTQfDJYVVNSAAB
```

Configuring ISAKMP Aggressive Mode

To configure ISAKMP aggressive mode, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp peer ip-address** *ip-address*
4. **set aggressive-mode client-endpoint** *client-endpoint*
5. **set aggressive-mode password** *password*

DETAILED STEPS

	Command	Description
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp peer ip-address** *ip-address* **Example:** `Router (config)# crypto isakmp peer ip-address 10.2.3.4`	To enable an IP Security (IPSec) peer for IKE querying of authentication, authorization, and accounting (AAA) for tunnel attributes in aggressive mode and to enter ISAKMP peer configuration mode.
Step 4	**set aggressive-mode client-endpoint** *client-endpoint* **Example:** `Router (config-isakmp-peer)# set aggressive-mode client-endpoint fqdn cisco.com`	Specifies the Tunnel-Client-Endpoint attribute within an ISAKMP peer configuration.
Step 5	**set aggressive-mode password** *password* **Example:** `Router (config-isakmp-peer)# set aggressive-mode password cisco`	Specifies the Tunnel-Password attribute within an ISAKMP peer configuration.

Example

The following s**how-running-config** sample output shows that an encrypted preshared key in ISAKMP aggressive mode has been configured.

```
crypto isakmp peer address 10.2.3.4
 set aggressive-mode password 6 ^aKPIQ_KJE_PPF^RXTQfDTIaLNeAAB
 set aggressive-mode client-endpoint fqdn cisco.com
```

Configuring a Unity Server Group Policy

To configure a unity server group policy, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp client configuration group** *group-name*
4. **pool** *name*
5. **domain** *name*
6. **key** *name*

DETAILED STEPS

	Command	Description
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp client configuration group** *group-name* **Example:** `Router (config)# crypto isakmp client configuration group foo`	Specifies the policy profile of the group that will be defined and enters ISAKMP group configuration mode.
Step 4	**pool** *name* **Example:** `Router (config-isakmp-group)# pool foopool`	Defines a local pool address.
Step 5	**domain** name **Example:** `Router (config-isakmp-group)# domain cisco.com`	Specifies the Domain Name Service (DNS) domain to which a group belongs.
Step 6	**key** *name* **Example:** `Router (config-isakmp-group)# key cisco`	Specifies the IKE preshared key for group policy attribute definition.

Example

The following **show-running-config** sample output shows that an encrypted key has been configured for a unity server group policy:

```
crypto isakmp client configuration group foo
 key 6 cZZgDZPOE\dDPF^RXTQfDTIaLNeAAB
 domain cisco.com
 pool foopool
```

Configuring an Easy VPN Client

To configure an Easy VPN client, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec client ezvpn** *name*
4. **peer** *ipaddress*
5. **mode client**
6. **group** *group-name* **key** *group-key*
7. **connect manual**

DETAILED STEPS

	Command	Description
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec client ezvpn** *name* **Example:** `Router (config)# crypto ipsec client ezvpn foo`	Creates a Cisco Easy VPN remote configuration and enters Cisco Easy VPN remote configuration mode.
Step 4	**peer** *ipaddress* **Example:** `Router (config-isakmp-peer)# peer 10.2.3.4`	Sets the peer IP address for the VPN connection.

	Command	Description
Step 5	**mode client** **Example:** `Router (config-isakmp-ezpvy)# mode client`	Automatically configures the router for Cisco Easy VPNclient mode operation, which uses Network Address Translation (NAT) or Peer Address Translation (PAT) address translations.
Step 6	**group** *group-name* **key** *group-key* **Example:** `Router (config-isakmp-ezvpn)# group foo key cisco`	Specifies the group name and key value for the VPN connection.
Step 7	**connect manual** **Example:** `Router (config-isakmp-ezvpn)# connect manual`	Specifies the manual setting for directing the Cisco Easy VPN remote client to wait for a command or application program interface (API) call before attenpting to establish the Cisco Easy VPN remote connection.

Example

The following **show-running-config** sample output shows that an Easy VPN client has been configured. The key has been encrypted.

```
crypto ipsec client ezvpn foo
 connect manual
 group foo key 6 gdMI`S^^[GIcPF^RXTQfDFKEO\RAAB
 mode client
 peer 10.2.3.4
```

Configuration Examples for Encrypted Preshared Key

This section provides the following configuration examples:

- Encrypted Preshared Key: Example, page 875
- No Previous Key Present: Example, page 876
- Key Already Exists: Example, page 876
- Key Already Exists But the User Wants to Key In Interactively: Example, page 876
- No Key Present But the User Wants to Key In Interactively: Example, page 876
- Removal of the Password Encryption: Example, page 877

Encrypted Preshared Key: Example

The following is an example of a configuration for which a type 6 preshared key has been encrypted. It includes the prompts and messages that a user might see.

```
Router (config)# crypto isakmp key cisco address 10.0.0.2
Router (config)# exit
Router# show runnning-config | include crypto isakmp key
 crypto isakmp key cisco address 10.0.0.2
Router#
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
Router (config)# password encryption aes
Router (config)# key config-key password-encrypt
New key:
Confirm key:
Router (config)#
01:46:40: TYPE6_PASS: New Master key configured, encrypting the keys with
the new master key
Router (config)# exit
Router # show running-config | include crypto isakmp key
crypto isakmp key 6 CXWdhVTZYB_Vcd^`cIHDOahiFTa address 10.0.0.2
```

No Previous Key Present: Example

In the following configuration example, no previous key is present:

```
Router (config)# key config-key password-encryption testkey 123
```

Key Already Exists: Example

In the following configuration example, a key already exists:

```
Router (config)# key config-key password-encryption testkey123
Old key:
Router (config)#
```

Key Already Exists But the User Wants to Key In Interactively: Example

In the following configuration example, the user wants to key in interactively, but a key already exists. The Old key, New key, and Confirm key prompts will show on your screen if you enter the **key config-key password-encryption** command and press the enter key to get into interactive mode.

```
Router (config)# key config-key password-encryption
Old key:
New key:
Confirm key:
```

No Key Present But the User Wants to Key In Interactively: Example

In the following example, the user wants to key in interactively, but no key is present. The New key and Confirm key prompts will show on your screen if you are in interactive mode.

```
Router (config)# key config-key password-encryption
New key:
Confirm key:
```

Removal of the Password Encryption: Example

In the following configuration example, the user wants to remove the encrypted password. The "WARNING: All type 6 encrypted keys will become unusable. Continue with master key deletion? [yes/no]:" prompt will show on your screen if you are in interactive mode.

```
Router (config)# no key config-key password-encryption

WARNING: All type 6 encrypted keys will become unusable. Continue with master key
deletion ? [yes/no]: y
```

Where to Go Next

Configure any other preshared keys.

Additional References

The following sections provide references related to Encrypted Preshared Key.

Related Documents

Related Topic	Document Title
Configuring passwords	The section "Part 4: IP Security and Encryption" of the *Cisco IOS Security Configuration Guide* *Cisco IOS Security Command Reference*, Release 12.3 T

Standards

Standards	Title
This feature has no new or modified standards.	—

MIBs

MIBs	MIBs Link
This feature has no new or modified MIBs.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
This feature has no new or modified RFCs.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

New Commands

- **key config-key password-encryption**
- **password encryption aes**
- **password logging**

Modified Commands

- **crypto ipsec client ezvpn (global)**
- **crypto isakmp client configuration group**
- **crypto isakmp key**
- **pre-shared-key**
- **set aggressive-mode password**

Configuring Internet Key Exchange for IPSec VPNs

This module describes how to configure the Internet Key Exchange (IKE) protocol for basic IP Security (IPSec) virtual private networks (VPNs). IKE is a key management protocol standard that is used in conjunction with the IPSec standard. IPSec is an IP security feature that provides robust authentication and encryption of IP packets.

IPSec can be configured without IKE, but IKE enhances IPSec by providing additional features, flexibility, and ease of configuration for the IPSec standard.

IKE is a hybrid protocol, which implements the Oakley key exchange and Skeme key exchange inside the Internet Security Association Key Management Protocol (ISAKMP) framework. (ISAKMP, Oakley, and Skeme are security protocols implemented by IKE.)

Module History

This module was first published on May 2, 2005, and last updated on May 2, 2005.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all features. To find information about feature support and configuration, use the "Feature Information for Configuring IKE for IPSec VPNs" section on page 901.

Contents

- Prerequisites for IKE Configuration, page 880
- Restrictions for IKE Configuration, page 880
- Information About Configuring IKE for IPSec VPNs, page 880
- How to Configure IKE for IPSec VPNs, page 882
- Configuration Examples for an IKE Configuration, page 896
- Where to Go Next, page 898
- Additional References, page 898

Prerequisites for IKE Configuration

- You should you should be familiar with the concepts and tasks explained in the module "Configuring Security for VPNs with IPSec."
- Ensure that your access control lists (ACLs) are compatible with IKE. Because IKE negotiation uses User Datagram Protocol (UDP) on port 500, your ACLs must be configured so that UDP port 500 traffic is not blocked at interfaces used by IKE and IPSec. In some cases you might need to add a statement to your ACLs to explicitly permit UDP port 500 traffic.

Restrictions for IKE Configuration

The following restrictions are applicable when configuring IKE negotiation:

- The initiating router *must not* have a certificate associated with the remote peer.
- The preshared key *must* be by fully qualified domain name (FQDN) on both peers. (To configure the preshared key, enter the **crypto isakmp key** command.)
- The communicating routers *must* have a FQDN host entry for each other in their configurations.
- The communicating routers *must* be configured to authenticate by hostname, *not* by IP address; thus, you should use the **crypto isakmp identity hostname** command.

Information About Configuring IKE for IPSec VPNs

To configure IKE for IPSec VPNs, you should understand the following concepts:

- Supported Standards for Use with IKE, page 880
- IKE Benefits, page 882
- IKE Main Mode and Aggressive Mode, page 882

Supported Standards for Use with IKE

Cisco implements the following standards:

- IPSec—IP Security Protocol. IPSec is a framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer; it uses IKE to handle negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPSec. IPSec can be used to protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.
- ISAKMP—Internet Security Association and Key Management Protocol. A protocol framework that defines payload formats, the mechanics of implementing a key exchange protocol, and the negotiation of a security association.
- Oakley—A key exchange protocol that defines how to derive authenticated keying material.
- Skeme—A key exchange protocol that defines how to derive authenticated keying material, with rapid key refreshment.

The component technologies implemented for use by IKE include the following:

- AES—Advanced Encryption Standard. A cryptographic algorithm that protects sensitive, unclassified information. AES is privacy transform for IPSec and IKE and has been developed to replace the DES. AES is designed to be more secure than DES: AES offers a larger key size, while ensuring that the only known approach to decrypt a message is for an intruder to try every possible key. AES has a variable key length—the algorithm can specify a 128-bit key (the default), a 192-bit key, or a 256-bit key.
- DES—Data Encryption Standard. An algorithm that is used to encrypt packet data. IKE implements the 56-bit DES-CBC with Explicit IV standard. Cipher Block Chaining (CBC) requires an initialization vector (IV) to start encryption. The IV is explicitly given in the IPSec packet.

  Cisco IOS software also implements Triple DES (168-bit) encryption, depending on the software versions available for a specific platform. Triple DES (3DES) is a strong form of encryption that allows sensitive information to be transmitted over untrusted networks. It enables customers, particularly in the finance industry, to utilize network-layer encryption.

  Note Cisco IOS images that have strong encryption (including, but not limited to, 56-bit data encryption feature sets) are subject to United States government export controls, and have a limited distribution. Images that are to be installed outside the United States require an export license. Customer orders might be denied or subject to delay because of United States government regulations. Contact your sales representative or distributor for more information, or send e-mail to export@cisco.com.

- SEAL—Software Encryption Algorithm. An alternative algorithm to software-based DES, 3DES, and AES. SEAL encryption uses a 160-bit encryption key and has a lower impact to the CPU when compared to other software-based algorithms.
- Diffie-Hellman—A public-key cryptography protocol that allows two parties to establish a shared secret over an unsecure communications channel. Diffie-Hellman is used within IKE to establish session keys. 768-bit (the default), 1024-bit, and 1536-bit Diffie-Hellman groups are supported.
- MD5 (HMAC variant)—Message Digest 5. A hash algorithm used to authenticate packet data. HMAC is a variant that provides an additional level of hashing.
- SHA (HMAC variant)—Secure Hash Algorithm. A hash algorithm used to authenticate packet data. HMAC is a variant that provides an additional level of hashing.
- RSA signatures and RSA encrypted nonces—RSA is the public key cryptographic system developed by Ron Rivest, Adi Shamir, and Leonard Adleman. RSA signatures provide nonrepudiation, and RSA encrypted nonces provide repudiation. (Repudation and nonrepudation have to do with traceability.)

IKE interoperates with the following standard:

X.509v3 certificates—Used with the IKE protocol when authentication requires public keys. This certificate support allows the protected network to scale by providing the equivalent of a digital ID card to each device. When two devices intend to communicate, they exchange digital certificates to prove their identity (thus removing the need to manually exchange public keys with each peer or to manually specify a shared key at each peer).

IKE Benefits

IKE automatically negotiates IPSec security associations (SAs) and enables IPSec secure communications without costly manual preconfiguration. Specifically, IKE provides the following benefits:

- Eliminates the need to manually specify all the IPSec security parameters in the crypto maps at both peers.
- Allows you to specify a lifetime for the IPSec SA.
- Allows encryption keys to change during IPSec sessions.
- Allows IPSec to provide anti-replay services.
- Permits certification authority (CA) support for a manageable, scalable IPSec implementation.
- Allows dynamic authentication of peers.

IKE Main Mode and Aggressive Mode

IKE has two phases of key negotiation: phase 1 and phase 2. Phase 1 negotiates a security association (a key) between two IKE peers. The key negotiated in phase 1 enables IKE peers to communicate securely in phase 2. During phase 2 negotiation, IKE establishes keys (security associations) for other applications, such as IPSec.

Phase 1 negotiation can occur using main mode or aggressive mode. Main mode tries to protect all information during the negotiation, meaning that no information is available to a potential attacker. When main mode is used, the identities of the two IKE peers are hidden. Although this mode of operation is very secure, it is relatively costly in terms of the time it takes to complete the negotiation. Aggressive mode takes less time to negotiate keys between peers; however, it gives up some of the security provided by main mode negotiation. For example, the identities of the two parties trying to establish a security association are exposed to an eavesdropper.

The two modes serve different purposes and have different strengths. Main mode is slower than aggressive mode, but main mode is more secure and more flexible because it can offer an IKE peer more security proposals than aggressive mode. Aggressive mode is less flexible and not as secure, but much faster.

In Cisco IOS software, the two modes are not configurable. The default action for IKE authentication (rsa-sig, rsa-encr, or preshared) is to initiate main mode; however, in cases where there is no corresponding information to initiate authentication, and there is a preshared key associated with the hostname of the peer, Cisco IOS software can initiate aggressive mode. Cisco IOS software will respond in aggressive mode to an IKE peer that initiates aggressive mode.

How to Configure IKE for IPSec VPNs

If you do not want IKE to be used with your IPSec implementation, you can disable it at all IPSec peers via the **no crypto isakmp** command, skip the rest of this chapter, and begin your IPSec VPN.

Note If you disable IKE, you will have to manually specify all the IPSec SAs in the crypto maps at all peers, the IPSec SAs of the peers will never time out for a given IPSec session, the encryption keys will never change during IPSec sessions between the peers, anti-replay services will not be available between the peers, and public key infrastructure (PKI) support cannot be used.

IKE is enabled by default. IKE does not have to be enabled for individual interfaces, but it is enabled globally for all interfaces at the router.

Perform the following tasks to provide authentication of IPSec peers, negotiate IPSec SAs, and establish IPSec keys:

- Creating IKE Policies: Security Parameters for IKE Negotiation, page 883 (required)
- Configuring IKE Authentication, page 887 (required)
- Configuring IKE Mode Configuration, page 894

Creating IKE Policies: Security Parameters for IKE Negotiation

An IKE policy defines a combination of security parameters to be used during the IKE negotiation. You must create an IKE policy at each peer participating in the IKE exchange.

If you do not configure any IKE policies, your router will use the default policy, which is always set to the lowest priority and which contains the default value of each parameter.

About IKE Policies

Because IKE negotiations must be protected, each IKE negotiation begins by agreement of both peers on a common (shared) IKE policy. This policy states which security parameters will be used to protect subsequent IKE negotiations and mandates how the peers are authenticated.

After the two peers agree upon a policy, the security parameters of the policy are identified by an SA established at each peer, and these SAs apply to all subsequent IKE traffic during the negotiation.

You can configure multiple, prioritized policies on each peer—each with a different combination of parameter values. However, at least one of these policies must contain exactly the same encryption, hash, authentication, and Diffie-Hellman parameter values as one of the policies on the remote peer. For each policy that you create, you assign a unique priority (1 through 10,000, with 1 being the highest priority).

Tip If you are interoperating with a device that supports only one of the values for a parameter, your choice is limited to the value supported by the other device. Aside from this limitation, there is often a trade-off between security and performance, and many of these parameter values represent such a trade-off. You should evaluate the level of security risks for your network and your tolerance for these risks.

IKE Peers Agreeing Upon a Matching IKE Policy

When the IKE negotiation begins, IKE searches for an IKE policy that is the same on both peers. The peer that initiates the negotiation will send all its policies to the remote peer, and the remote peer will try to find a match. The remote peer looks for a match by comparing its own highest priority policy against the policies received from the other peer. The remote peer checks each of its policies in order of its priority (highest priority first) until a match is found.

A match is made when both policies from the two peers contain the same encryption, hash, authentication, and Diffie-Hellman parameter values, and when the remote peer's policy specifies a lifetime that is less than or equal to the lifetime in the policy being compared. (If the lifetimes are not identical, the shorter lifetime—from the remote peer's policy—will be used.)

If a match is found, IKE will complete negotiation, and IPSec security associations will be created. If no acceptable match is found, IKE refuses negotiation and IPSec will not be established.

Note Depending on which authentication method is specified in a policy, additional configuration might be required (as described in the section "Configuring IKE Authentication"). If a peer's policy does not have the required companion configuration, the peer will not submit the policy when attempting to find a matching policy with the remote peer.

Restrictions

If you are configuring an AES IKE policy, note the following restrictions:

- Your router must support IPSec and long keys (the "k9" subsystem).
- AES cannot encrypt IPSec and IKE traffic if an acceleration card is present.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp policy** *priority*
4. **encryption** {**des** | **3des** | **aes** | **aes 192** | **aes 256**}
5. **hash** {**sha** | **md5**}
6. **authentication** {**rsa-sig** | **rsa-encr** | **pre-share**}
7. **group** {**1** | **2** | **5**}
8. **lifetime** *seconds*
9. **exit**
10. **exit**
11. **show crypto isakmp policy**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp policy** *priority* **Example:** `Router(config)# crypto isakmp policy 10`	Defines an IKE policy and enters config-isakmp configuration mode. • *priority*—Uniquely identifies the IKE policy and assigns a priority to the policy. Valid values: 1 to 10,000; 1 is the highest priority.

	Command or Action	Purpose
Step 4	**encryption** {**des** | **3des** | **aes** | **aes 192** | **aes 256**} **Example:** `Router(config-isakmp)# encryption aes 256`	Specifies the encryption algorithm. By default, the **des** keyword is used. • **des**—56-bit DES-CBC • **3des**—168-bit DES • **aes**—128-bit AES • **aes 192**—192-bit AES • **aes 256**—256-bit AES
Step 5	**hash** {**sha** | **md5**} **Example:** `Router(config-isakmp)# hash sha`	Specifies the hash algorithm. By default, SHA-1 (**sha**) is the used. **Note** MD5 has a smaller digest and is considered to be slightly faster than SHA-1.
Step 6	**authentication** {**rsa-sig** | **rsa-encr** | **pre-share**} **Example:** `Router(config-isakmp)# authentication pre-share`	Specifies the authentication method. By default, RSA signatures are used. • **rsa-sig**—RSA signatures require that you configure your peer routers to obtain certificates from a CA. • **rsa-encr**—RSA encrypted nonces require that you ensure each peer has the other peer's RSA public keys. • **pre-share**—Preshared keys require that you separately configure these preshared keys.
Step 7	**group** {**1** | **2** | **5**} **Example:** `Router(config-isakmp)# group 1`	Specifies the Diffie-Hellman group identifier. By default, D-H group 1 is used. • **1**—768-bit Diffie-Hellman • **2**—1024-bit Diffie-Hellman • **5**—1536-bit Diffie-Hellman **Note** The 1024-bit and 1536-bit Diffie-Hellman options are harder to "crack," but require more CPU time to execute.
Step 8	**lifetime** *seconds* **Example:** `Router(config-isakmp)# lifetime 180`	Specifies the lifetime of the IKE SA. • *seconds*—Time, in seconds, before each SA expires. Valid values: 60 to 86,400 seconds; default value: 86,400. **Note** The shorter the lifetime (up to a point), the more secure your IKE negotiations will be. However, with longer lifetimes, future IPSec SAs can be set up more quickly.
Step 9	**exit** **Example:** `Router(config-isakmp)# exit`	Exits config-isakmp configuration mode.

	Command or Action	Purpose
Step 10	**exit** **Example:** `Router(config)# exit`	Exits the global configuration mode.
Step 11	**show crypto isakmp policy** **Example:** `Router# show crypto isakmp policy`	(Optional) Displays all existing IKE policies.
Step 12	—	Repeat these steps for each policy you want to create.

Note These parameters apply to the IKE negotiations after the IKE SA is established.

Examples

The following sample output from the **show crypto isakmp policy** command displays a warning message after a user tries to configure an IKE encryption method that the hardware does not support:

```
Router# show crypto isakmp policy

Protection suite of priority 1
        encryption algorithm:  AES - Advanced Encryption Standard (256 bit keys).
WARNING:encryption hardware does not support the configured
encryption method for ISAKMP policy 1
        hash algorithm:        Secure Hash Standard
        authentication method: Pre-Shared Key
        Diffie-Hellman group:  #1 (768 bit)
        lifetime:              3600 seconds, no volume limit
```

Troubleshooting Tips

- Clear (and reinitialize) IPSec SAs by using the **clear crypto sa** EXEC command.

  Using the **clear crypto sa** command without parameters will clear out the full SA database, which will clear out active security sessions. You may also specify the **peer**, **map**, or **entry** keywords to clear out only a subset of the SA database. For more information,see the **clear crypto sa** command in the *Cisco IOS Security Command Reference*, Release 12.4.

- The default policy and default values for configured policies do not show up in the configuration when you issue the **show running-config** command. To see the default policy and any default values within configured policies, use the **show crypto isakmp policy** command.

- Any IPSec transforms or IKE encryption methods that the current hardware does not support should be disabled; they are ignored whenever an attempt to negotiate with the peer is made.

  If a user enters an IPSec transform or an IKE encryption method that the hardware does not support, a warning message will be generated. These warning messages are also generated at boot time. When an encrypted card is inserted, the current configuration is scanned. If any IPSec transforms or IKE encryption methods are found that are not supported by the hardware, a warning message will be generated.

What to Do Next

Depending on which authentication method you specified in your IKE policies (RSA signatures, RSA encrypted nonces, or preshared keys), you must do certain additional configuration tasks before IKE and IPSec can successfully use the IKE policies. For information on completing these additional tasks, refer to the following section "Configuring IKE Authentication."

To configure an AES-based transform set, see the module "Configuring Security for VPNs with IPSec."

Configuring IKE Authentication

After you have created at least one IKE policy in which you specified an authentication method (or accepted the default method), you need to configure an authentication method. IKE policies cannot be used by IPSec until the authentication method is successfully configured.

To configure IKE authentication, you should perform one of the following tasks, as appropriate:

- Configuring RSA Keys Manually for RSA Encrypted Nonces, page 888
- Configuring Preshared Keys, page 891
- Configuring RSA Keys to Obtain Certificates from a CA. For information on completing this task, see the module "Deploying RSA Keys Within a PKI."

IKE Authentication Methods: Overview

IKE authentication consists of three options—RSA signatures, RSA encrypted nonces, and preshared keys. Each authentication method requires additional configuration as follows:

RSA Signatures

With RSA signatures, you can configure the peers to obtain certificates from a CA. (The CA must be properly configured to issue the certificates.) Using a CA can dramatically improve the manageability and scalability of your IPSec network. Additionally, RSA signature-based authentication uses only two public key operations, whereas RAS encryption uses four public key operations, making it costlier in terms of overall performance. To properly configure CA support, see the chapter "Implementing and Managing a PKI."

The certificates are used by each peer to exchange public keys securely. (RSA signatures requires that each peer has the public signature key of the remote peer.) When both peers have valid certificates, they will automatically exchange public keys with each other as part of any IKE negotiation in which RSA signatures are used.

You can also exchange the public keys manually, as described in the section "Configuring RSA Keys Manually for RSA Encrypted Nonces."

RSA signatures provide nonrepudiation for the IKE negotiation. And, you can prove to a third party after the fact that you did indeed have an IKE negotiation with the remote peer.

RSA Encrypted Nonces

With RSA encrypted nonces, you must ensure that each peer has the public keys of the other peers.

Unlike RSA signatures, the RSA encrypted nonces method can not use certificates to exchange public keys. Instead, you ensure that each peer has the others' public keys by one of the following methods:

- Manually configuring RSA keys as described in the section "Configuring RSA Keys Manually for RSA Encrypted Nonces."

  or

- Ensuring that an IKE exchange using RSA signatures with certificates has already occurred between the peers. (The peers' public keys are exchanged during the RSA-signatures-based IKE negotiations if certificates are used.)

  To make that the IKE exchange happens, specify two policies: a higher-priority policy with RSA encrypted nonces and a lower-priority policy with RSA signatures. When IKE negotiations occur, RSA signatures will be used the first time because the peers do not yet have each other's public keys. Then future IKE negotiations can use RSA encrypted nonces because the public keys will have been exchanged.

Note This alternative requires that you already have CA support configured.

RSA encrypted nonces provide repudiation for the IKE negotiation; however, unlike RSA signatures, you cannot prove to a third party that you had an IKE negotiation with the remote peer.

Preshared Keys

With preshared keys, you must configure them as described in the section "Configuring Preshared Keys."

Preshared keys are clumsy to use if your secured network is large, and they do not scale well with a growing network. However, they do not require use of a CA, as do RSA signatures, and might be easier to set up in a small network with fewer than ten nodes. RSA signatures also can be considered more secure when compared with preshared key authentication.

Note If RSA encryption is configured and signature mode is negotiated (and certificates are used for signature mode), the peer will request both signature and encryption keys. Basically, the router will request as many keys as the configuration will support. If RSA encryption is not configured, it will just request a signature key.

Prerequisites

You must have configured at least one IKE policy, which is where the authentication method was specified (or RSA signatures was accepted by default).

Configuring RSA Keys Manually for RSA Encrypted Nonces

To manually configure RSA keys, perform this task for each IPSec peer that uses RSA encrypted nonces in an IKE policy.

Note This task can be performed only if a CA is not in use.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto key generate rsa** {**general-keys** | **usage-keys**} [**label** *key-label*] [**exportable**] [**modulus** *modulus-size*]
4. **exit**
5. **show crypto key mypubkey rsa**
6. **configure terminal**
7. **crypto key pubkey-chain rsa**
8. **named-key** *key-name* [**encryption** | **signature**]

   or

   addressed-key *key-address* [**encryption** | **signature**]
9. **address** *ip-address*
10. **key-string** *key-string*
11. **quit**
12. Repeat these steps at each peer that uses RSA encrypted nonces in an IKE policy.
13. **exit**
14. **exit**
15. **show crypto key pubkey-chain rsa** [**name** *key-name* | **address** *key-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto key generate rsa** {**general-keys** | **usage-keys**} [**label** *key-label*] [**exportable**] [**modulus** *modulus-size*] **Example:** `Router(config)# crypto key generate rsa general-keys modulus 360`	Generates RSA keys. • If a *key-label* argument is not specified, the default value, which is the fully qualified domain name (FQDN) of the router, is used.
Step 4	**exit** **Example:** `Router(config)# exit`	(Optional) Exits global configuration mode.

	Command or Action	Purpose
Step 5	**show crypto key mypubkey rsa** **Example:** `Router# show crypto key mypubkey rsa`	(Optional) Displays the generated RSA public keys.
Step 6	**configure terminal** **Example:** `Router# configure terminal`	Returns to global configuration mode.
Step 7	**crypto key pubkey-chain rsa** **Example:** `Router(config)# crypto key pubkey-chain rsa`	Enters public key configuration mode (so you can manually specify the RSA public keys of other devices).
Step 8	**named-key** *key-name* [**encryption** | **signature**] **Example:** `Router(config-pubkey-chain)# named-key otherpeer.example.com` or **addressed-key** *key-address* [encryption | signature] **Example:** `Router(config-pubkey-chain)# addressed-key 10.1.1.2 encryption`	Indicates which remote peer's RSA public key you are going to specify and enters public key configuration mode. If the remote peer uses its host name as its ISAKMP identity, use the **named-key** command and specify the remote peer's FQDN, such as somerouter.example.com, as the *key-name*. If the remote peer uses its IP address as its ISAKMP identity, use the **addressed-key** command and specify the remote peer's IP address as the *key-address*.
Step 9	**address** *ip-address* **Example:** `Router(config-pubkey-key)# address 10.5.5.1`	Specifies the IP address of the remote peer. If you use the **named-key** command, you need to use this command to specify the IP address of the peer.
Step 10	**key-string** *key-string* **Example:** `Router(config-pubkey-key)# key-string` `Router(config-pubkey)# 00302017 4A7D385B 1234EF29 335FC973` `Router(config-pubkey)# 2DD50A37 C4F4B0FD 9DADE748 429618D5` `Router(config-pubkey)# 18242BA3 2EDFBDD3 4296142A DDF7D3D8` `Router(config-pubkey)# 08407685 2F2190A0 0B43F1BD 9A8A26DB` `Router(config-pubkey)# 07953829 791FCDE9 A98420F0 6A82045B` `Router(config-pubkey)# 90288A26 DBC64468 7789F76E EE21`	Specifies the RSA public key of the remote peer. (This key was previously viewed by the administrator of the remote peer when the RSA keys of the remote router were generated.)
Step 11	**quit** **Example:** `Router(config-pubkey-k)# quit`	Returns to public key chain configuration mode.

	Command or Action	Purpose
Step 12	—	Repeat these steps at each peer that uses RSA encrypted nonces in an IKE policy.
Step 13	**exit** **Example:** `Router(config-pubkey-c)# exit`	Returns to global configuration mode.
Step 14	**exit** **Example:** `Router(config)# exit`	Returns to EXEC mode.
Step 15	**show crypto key pubkey-chain rsa** [**name** *key-name* | **address** *key-address*] **Example:** `Router# show crypto key pubkey-chain rsa`	(Optional) Displays either a list of all RSA public keys that are stored on your router or details of a particular RSA key that is stored on your router.

Configuring Preshared Keys

To configure preshared keys, perform these steps at each peer that uses preshared keys in an IKE policy.

Setting ISAKMP Identity for Preshared Keys

You should set the ISAKMP identity for each peer that uses preshared keys in an IKE policy.

When two peers use IKE to establish IPSec SAs, each peer sends its identity to the remote peer. Each peer sends either its host name or its IP address, depending on how you have set the ISAKMP identity of the router.

By default, a peer's ISAKMP identity is the IP address of the peer. If appropriate, you could change the identity to be the peer's host name instead. As a general rule, set the identities of all peers the same way—either all peers should use their IP addresses or all peers should use their hostnames. If some peers use their host names and some peers use their IP addresses to identify themselves to each other, IKE negotiations could fail if the identity of a remote peer is not recognized and a DNS lookup is unable to resolve the identity.

Mask Preshared Keys

A mask preshared key allows a group of remote users with the same level of authentication to share an IKE preshared key. The preshared key of the remote peer must match the preshared key of the local peer for IKE authentication to occur.

A mask preshared key is usually distributed through a secure out-of-band channel. In a remote peer-to-local peer scenario, any remote peer with the IKE preshared key configured can establish IKE SAs with the local peer.

If you specify the **mask** keyword with the **crypto isakmp key** command, it is up to you to use a subnet address, which will allow more peers to share the same key. That is, the preshared key is no longer restricted to use between two users.

Note Using 0.0.0.0 as a subnet address is not recommended because it encourages group preshared keys, which allow all peers to have the same group key, thereby reducing the security of your user authentication.

Disable Xauth on a Specific IPSec Peer

Disabling Extended Authentication (Xauth) for static IPSec peers prevents the routers from being prompted for Xauth information—username and password.

Without the ability to disable Xauth, a user cannot select which peer on the same crypto map should use Xauth. That is, if a user has router-to-router IPSec on the same crypto map as a VPN-client-to-Cisco-IOS IPSec, both peers are prompted for a username and password. In addition, a remote static peer (a Cisco IOS router) cannot establish an IKE SA with the local Cisco IOS router. (Xauth is not an optional exchange, so if a peer does not respond to an Xauth request, the IKE SA is deleted.) Thus, the same interface cannot be used to terminate IPSec to VPN clients (that need Xauth) as well as other Cisco IOS routers (that cannot respond to Xauth) unless this feature is implemented.

Note Xauth can be disabled only if preshared keys are used as the authentication mechanism for the given crypto map.

Restrictions

- Preshared do not scale well with a growing network.
- Mask preshared keys have the following restrictions:
  - The SA cannot be established between the IPSec peers until all IPSec peers are configured for the same preshared key.
  - The mask preshared key must be distinctly different for remote users requiring varying levels of authorization. You must configure a new preshared key for each level of trust and assign the correct keys to the correct parties. Otherwise, an untrusted party may obtain access to protected data.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp identity** {**address** | **hostname**}
4. **ip host** *hostname address1* [*address2...address8*]
5. **crypto isakmp key** *keystring* **address** *peer-address* [**mask**] [**no-xauth**]

   or

   crypto isakmp key *keystring* **hostname** *hostname* [**no-xauth**]
6. **crypto isakmp key** *keystring* **address** *peer-address* [**mask**] [**no-xauth**]

   or

   crypto isakmp key *keystring* **hostname** *hostname* [**no-xauth**]
7. Repeat these steps for each peer that uses preshared keys.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp identity** {**address** | **hostname**} **Example:** `Router(config)# crypto isakmp identity address`	Specifies the peer's ISAKMP identity by IP address or by hostname at the local peer. • **address**—Typically used when there is only one interface (and therefore only one IP address) that will be used by the peer for IKE negotiations, and the IP address is known. • **hostname**—Should be used if there is more than one interface on the peer that might be used for IKE negotiations, or if the interface's IP address is unknown (such as with dynamically assigned IP addresses).
Step 4	**ip host** *hostname address1* [*address2...address8*] **Example:** `Router(config)# ip host RemoteRouter.example.com 192.168.0.1`	If the local peer's ISAKMP identity was specified using a hostname, maps the peer's host name to its IP address(es) at all the remote peers. (This step might be unnecessary if the hostname or address is already mapped in a DNS server.)
Step 5	**crypto isakmp key** *keystring* **address** *peer-address* [**mask**] [**no-xauth**] **Example:** `Router(config)# crypto isakmp key sharedkeystring address 192.168.1.33 no-xauth` or **crypto isakmp key** *keystring* **hostname** *hostname* [**no-xauth**] **Example:** `Router(config) crypto isakmp key sharedkeystring hostname RemoteRouter.example.com`	Specifies at the local peer the shared key to be used with a particular remote peer. If the remote peer specified its ISAKMP identity with an address, use the **address** keyword in this step; otherwise use the **hostname** keyword in this step. • **no-xauth—**Prevents the router from prompting the peer for Xauth information. Use this keyword if router-to-router IPSec is on the same crypto map as VPN-client-to-Cisco IOS IPSec. **Note** According to the design of preshared key authentication in IKE main mode, preshared keys must be based on the IP address of the peers. Although you can send hostname as the identity of preshared key authentication, the key is searched on the IP address of the peer; if the key is not found (based on the IP address) the negotiation will fail.

	Command or Action	Purpose
Step 6	**crypto isakmp key** *keystring* **address** *peer-address* [**mask**] [**no-xauth**] **Example:** `Router(config) crypto isakmp key sharedkeystring address 10.0.0.1` or **crypto isakmp key** *keystring* **hostname** *hostname* [**no-xauth**] **Example:** `Router(config) crypto isakmp key sharedkeystring hostname LocalRouter.example.com`	Specifies at the remote peer the shared key to be used with the local peer. This is the same key you just specified at the local peer. If the local peer specified its ISAKMP identity with an address, use the **address** keyword in this step; otherwise use the **hostname** keyword in this step.
Step 7	—	Repeat these steps at each peer that uses preshared keys in an IKE policy.

Configuring IKE Mode Configuration

Perform the following task to configure IKE mode configuration.

About IKE Mode Configuration

IKE mode configuration, as defined by the Internet Engineering Task Force (IETF) , allows a gateway to download an IP address (and other network level configuration) to the client as part of an IKE negotiation. Using this exchange, the gateway gives IP addresses to the IKE client to be used as an "inner" IP address encapsulated under IPSec. This method provides a known IP address for the client that can be matched against IPSec policy.

To implement IPSec VPNs between remote access clients that have dynamic IP addresses and a corporate gateway, you have to dynamically administer scalable IPSec policy on the gateway once each client is authenticated. With IKE Mode Configuration, the gateway can set up scalable policy for a very large set of clients irrespective of the IP addresses of those clients.

There are two types of IKE Mode Configuration:

- Gateway initiation—Gateway initiates the configuration mode with the client. Once the client responds, the IKE modifies the identity of the sender, the message is processed, and the client receives a response.
- Client initiation—Client initiates the configuration mode with the gateway. The gateway responds with an IP address that it has allocated for the client.

Restrictions

IKE Mode Configuration has the following restrictions:

- Interfaces with crypto maps that are configured for IKE Mode Configuration may experience a slightly longer connection setup time, which is true even for IKE peers that refuse to be configured or do not respond to the configuration mode request. In both cases, the gateway initiates the configuration of the client.
- This feature was not designed to enable the configuration mode for every IKE connection by default. Configure this feature at the global crypto map level.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip local pool** *pool-name start-addr end-addr*
4. **crypto isakmp client configuration address-pool local** *pool-name*
5. **crypto map** *tag* **client configuration address** [**initiate** | **respond**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip local pool** *pool-name start-addr end-addr* **Example:** `Router(config) ip local pool ire 172.16.23.0 172.16.23.255`	Defines an existing local address pool that defines a set of addresses.
Step 4	**crypto isakmp client configuration address-pool local** *pool-name* **Example:** `Router(config) crypto isakmp client configuration address-pool local ire`	References the local address pool in the IKE configuration.
Step 5	**crypto map** *tag* **client configuration address** [**initiate** | **respond**] **Example:** `Router(config)# crypto map dyn client configuration address initiate`	Configures IKE Mode Configuration in global crypto map configuration mode.

Configuration Examples for an IKE Configuration

This section contains the following configuration examples:

- Creating IKE Policies: Examples, page 896
- Configuring IKE Authentication: Example, page 897

Creating IKE Policies: Examples

This section contains the following examples, which show how to configure a 3DES IKE policy and an AES IKE policy:

- Creating 3DES IKE Policies: Example, page 896
- Creating an AES IKE Policy: Example, page 897

Creating 3DES IKE Policies: Example

This example creates two IKE policies, with policy 15 as the highest priority, policy 20 as the next priority, and the existing default priority as the lowest priority. It also creates a preshared key to be used with policy 20 with the remote peer whose IP address is 192.168.224.33.

```
crypto isakmp policy 15
 encryption 3des
 hash md5
 authentication rsa-sig
 group 2
 lifetime 5000
!
crypto isakmp policy 20
 authentication pre-share
 lifetime 10000
!
crypto isakmp key 1234567890 address 192.168.224.33
```

In the example, the encryption des of policy 15 would not appear in the written configuration because this is the default value for the encryption algorithm parameter.

If the **show crypto isakmp policy** command is issued with this configuration, the output is as follows:

```
Protection suite priority 15
encryption algorithm:3DES - Triple Data Encryption Standard (168 bit keys)
hash algorithm:Message Digest 5
authentication method:Rivest-Shamir-Adleman Signature
Diffie-Hellman group:#2 (1024 bit)
lifetime:5000 seconds, no volume limit
Protection suite priority 20
encryption algorithm:DES - Data Encryption Standard (56 bit keys)
hash algorithm:Secure Hash Standard
authentication method:preshared Key
Diffie-Hellman group:#1 (768 bit)
lifetime:10000 seconds, no volume limit
Default protection suite
encryption algorithm:DES - Data Encryption Standard (56 bit keys)
hash algorithm:Secure Hash Standard
authentication method:Rivest-Shamir-Adleman Signature
Diffie-Hellman group:#1 (768 bit)
lifetime:86400 seconds, no volume limit
```

Note that although the output shows “no volume limit” for the lifetimes, you can configure only a time lifetime (such as 86,400 seconds); volume-limit lifetimes are not configurable.

Creating an AES IKE Policy: Example

The following example is sample output from the **show running-config** command. In this example, the AES 256-bit key is enabled.

```
Current configuration : 1665 bytes
!
version 12.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname "Router1"
!
!
ip subnet-zero
!
!
no ip domain lookup
!
ip audit notify log
ip audit po max-events 100
!
crypto isakmp policy 10
 encryption aes 256
 authentication pre-share
 lifetime 180
crypto isakmp key cisco123 address 10.0.110.1
!
!
crypto ipsec transform-set aesset esp-aes 256 esp-sha-hmac
 mode transport
!
crypto map aesmap 10 ipsec-isakmp
 set peer 10.0.110.1
 set transform-set aesset
 match address 120
!
.
.
.
```

Configuring IKE Authentication: Example

The following example shows how to manually specify the RSA public keys of two IPSec peer— the peer at 10.5.5.1 uses general-purpose keys, and the other peer uses special-usage keys:

```
crypto key pubkey-chain rsa
 named-key otherpeer.example.com
 address 10.5.5.1
 key-string
 005C300D 06092A86 4886F70D 01010105
 00034B00 30480241 00C5E23B 55D6AB22
 04AEF1BA A54028A6 9ACC01C5 129D99E4
 64CAB820 847EDAD9 DF0B4E4C 73A05DD2
 BD62A8A9 FA603DD2 E2A8A6F8 98F76E28
 D58AD221 B583D7A4 71020301 0001
```

```
quit
exit
addressed-key 10.1.1.2 encryption
key-string
00302017 4A7D385B 1234EF29 335FC973
2DD50A37 C4F4B0FD 9DADE748 429618D5
18242BA3 2EDFBDD3 4296142A DDF7D3D8
08407685 2F2190A0 0B43F1BD 9A8A26DB
07953829 791FCDE9 A98420F0 6A82045B
90288A26 DBC64468 7789F76E EE21
quit
exit
addressed-key 10.1.1.2 signature
key-string
0738BC7A 2BC3E9F0 679B00FE 53987BCC
01030201 42DD06AF E228D24C 458AD228
58BB5DDD F4836401 2A2D7163 219F882E
64CE69D4 B583748A 241BED0F 6E7F2F16
0DE0986E DF02031F 4B0B0912 F68200C4
C625C389 0BFF3321 A2598935 C1B1
quit
exit
exit
```

Where to Go Next

After you have successfully configured IKE negotiation, you can begin configuring IPSec. For information on completing these tasks, see the module "Configuring Security for VPNs With IPSec."

Additional References

The following sections provide references related to configuring IKE for IPSec VPNs.

Related Documents

Related Topic	Document Title
IPSec configuration	"Configuring Security for VPNs with IPSec" module
Configuring RSA keys to obtain certificates from a CA	"Deploying RSA Keys Within a PKI" module
IKE, IPSec, and PKI configuration commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*, Release 12.4

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2408	*Internet Security Association and Key Management Protocol (ISAKMP)*
RFC 2409	*The Internet Key Exchange (IKE)*
RFC 2412	*The OAKLEY Key Determination Protocol*

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Glossary

anti-replay—Security service in which the receiver can reject old or duplicate packets to protect itself against replay attacks. IPSec provides optional anti-replay services by use of a sequence number and the use of authentication.

data authentication—Verification of the integrity and origin of the data.

Data authentication can refer either to integrity alone or to both of these concepts (although data origin authentication is dependent upon data integrity).

peer—In the context of this chapter, a "peer"is a router or other device that participates in IPSec and IKE.

PFS—perfect forward secrecy. Cryptographic characteristic associated with a derived shared secret value. With PFS, if one key is compromised, previous and subsequent keys are not also compromised, because subsequent keys are not derived from previous keys.

repudiation—Quality that prevents a third party from being able to prove that a communication between two other parties ever took place. Repudiation is a desirable quality if you do not want your communications to be traceable.

nonrepudiation—Quality that allows a third party to prove that a communication between two other parties took place. Nonrepudiation is desirable if you want to be able to trace your communications and prove that they occurred.

SA—security association. How two or more entities utilize security services to communicate securely.

For example, an IPSec SA defines the encryption algorithm (if used), the authentication algorithm, and the shared session key to be used during the IPSec connection. Both IPSec and IKE require and use SAs to identify the parameters of their connections. IKE can negotiate and establish its own SA. The IPSec SA is established either by IKE or by manual user configuration.

Note Refer to *Internetworking Terms and Acronyms* for terms not included in this glossary.

Feature Information for Configuring IKE for IPSec VPNs

Table 43 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for a specific command was introduced, see the command reference documentation.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Note Table 43 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 43 Feature Information for Configuring IKE for IPSec VPNs

Feature Name	Software Releases	Feature Configuration Information
Ability to Disable Extended Authentication for Static IPSec Peers	12.2(4)T	This feature allows a user to disable Xauth while configuring the preshared key for router-to-router IPSec. Thus, the router will not prompt the peer for a username and password, which are transmitted when Xauth occurs for VPN-client-to-Cisco-IOS IPSec. The following section provides information about this feature: • Configuring Preshared Keys The following command was modified by this feature: **crypto isakmp key**

Table 43 Feature Information for Configuring IKE for IPSec VPNs (continued)

Feature Name	Software Releases	Feature Configuration Information
Advanced Encryption Standard (AES)	12.2(8)T	This feature adds support for the new encryption standard AES, which is a privacy transform for IPSec and IKE and has been developed to replace DES. The following sections provide information about this feature: • Supported Standards for Use with IKE • Creating IKE Policies: Security Parameters for IKE Negotiation The following commands were modified by this feature: **crypto ipsec transform-set**, **encryption (IKE policy)**, **show crypto isakmp policy**, **show crypto ipsec transform-set**
SEAL Encryption	12.3(7)T	This feature adds support for SEAL encryption in IPSec. The following section provides information about this feature: • Supported Standards for Use with IKE The following command was modified by this feature: **crypto ipsec transform-set**

Security for VPNs with IPSec

This part consists of the following:

- Configuring Security for VPNs with IPSec
- Ability to Disable Extended Authentication for Static IPSec Peers
- Cisco Easy VPN Remote
- Crypto Access Check on Clear-Text Packets
- DF Bit Override Functionality with IPSec Tunnels
- Distinguished Name Based Crypto Maps
- Dynamic Multipoint VPN (DMVPN)
- Easy VPN Remote RSA Signature Support
- Easy VPN Server
- Invalid Security Parameter Index Recovery
- IP Security VPN Monitoring
- IPSec and Quality of Service
- IPSec Anti-Replay Window: Expanding and Disabling
- IPSec Dead Peer Detection Periodic Message Option
- IPSec NAT Transparency
- IPSec Preferred Peer
- IPSec Security Association Idle Timers
- IPSec—SNMP Support
- IPSec Virtual Tunnel Interface
- IPSec VPN Accounting
- IPSec VPN High Availability Enhancements
- L2TP Security
- Low Latency Queueing (LLQ) for IPSec Encryption Engines
- L2TP—IPSec Support for NAT and PAT Windows Clients
- Pre-Fragmentation for IPSec VPNs
- Real-Time Resolution for IPSec Tunnel Peer
- Reverse Route Injection

- SafeNet IPSec VPN Client Support
- Stateful Failover for IPSec
- VRF-Aware IPSec

Configuring Security for VPNs with IPSec

This module describes how to configure basic IP Security (IPSec) virtual private networks (VPNs). IPSec is a framework of open standards developed by the Internet Engineering Task Force (IETF). It provides security for transmission of sensitive information over unprotected networks such as the Internet. IPSec acts at the network layer, protecting and authenticating IP packets between participating IPSec devices ("peers"), such as Cisco routers.

Module History

This module was first published on May 2, 2005, and last updated on May 2, 2005.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all features. To find information about feature support and configuration, use the "Feature Information for Security for VPNs with IPSec" section on page 939.

Contents

- Prerequisites for Configuring Security for VPNs with IPSec, page 905
- Restrictions for Configuring Security for VPNs with IPSec, page 906
- Information About Configuring Security for VPNs with IPSec, page 906
- How to Configure IPSec VPNs, page 912
- Configuration Examples for Configuring an IPSec VPN, page 935
- Additional References, page 937
- Glossary, page 938
- Feature Information for Security for VPNs with IPSec, page 939

Prerequisites for Configuring Security for VPNs with IPSec

IKE Configuration

You must configure Internet Key Exchange (IKE) as described in the module "Configuring Internet Key Exchange Security for IPSec VPNs."

Even if you decide to not use IKE, you still must disable it as described in the module "Configuring Internet Key Exchange for IPSec VPNs."

Ensure Access Lists Are Compatible with IPSec

IKE uses UDP port 500. The IPSec Encapsulating Security Payload (ESP) and Authentication Header (AH) protocols use protocol numbers 50 and 51. Ensure that your access lists are configured so that protocol 50, 51, and User Datagram Protocol (UDP) port 500 traffic is not blocked at interfaces used by IPSec. In some cases you might need to add a statement to your access lists to explicitly permit this traffic.

Restrictions for Configuring Security for VPNs with IPSec

Unicast IP Datagram Application Only

At this time, IPSec can be applied to unicast IP datagrams only. Because the IPSec Working Group has not yet addressed the issue of group key distribution, IPSec does not currently work with multicasts or broadcast IP datagrams.

NAT Configuration

If you use Network Address Translation (NAT), you should configure static NAT translations so that IPSec will work properly. In general, NAT translation should occur before the router performs IPSec encapsulation; in other words, IPSec should be working with global addresses.

Information About Configuring Security for VPNs with IPSec

To configure basic IPSec VPNs, you should understand the following concepts:

- Supported Standards, page 906
- Supported Hardware, Switching Paths, and Encapsulation, page 907
- IPSec Functionality Overview, page 910
- IPSec Traffic Nested to Multiple Peers, page 912

Supported Standards

Cisco implements the following standards with this feature:

- IPSec—IP Security Protocol. IPSec is a framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer; it uses IKE to handle negotiation of protocols and algorithms based on local policy, and to generate the encryption and authentication keys to be used by IPSec. IPSec can be used to protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

  Note The term IPSec is sometimes used to describe the entire protocol of IPSec data services and IKE security protocols and is also sometimes used to describe only the data services.

  IPSec is documented in a series of Internet Drafts, all available at http://www.ietf.org/html.charters/ipsec-charter.html.

- IKE—A hybrid protocol that implements Oakley and SKEME key exchanges inside the ISAKMP framework. While IKE can be used with other protocols, its initial implementation is with the IPSec protocol. IKE provides authentication of the IPSec peers, negotiates IPSec security associations, and establishes IPSec keys.

The component technologies implemented for IPSec include:

- AES—Advanced Encryption Standard. A cryptographic algorithm that protects sensitive, unclassified information. AES is privacy transform for IPSec and IKE and has been developed to replace the DES. AES is designed to be more secure than DES: AES offers a larger key size, while ensuring that the only known approach to decrypt a message is for an intruder to try every possible key. AES has a variable key length—the algorithm can specify a 128-bit key (the default), a 192-bit key, or a 256-bit key.
- DES—Data Encryption Standard. An algorithm that is used to encrypt packet data. Cisco IOS implements the mandatory 56-bit DES-CBC with Explicit IV. Cipher Block Chaining (CBC) requires an initialization vector (IV) to start encryption. The IV is explicitly given in the IPSec packet. For backwards compatibility, Cisco IOS IPSec also implements the RFC 1829 version of ESP DES-CBC.

  Cisco IOS also implements Triple DES (168-bit) encryption, depending on the software versions available for a specific platform. Triple DES (3DES) is a strong form of encryption that allows sensitive information to be transmitted over untrusted networks. It enables customers to utilize network layer encryption.

  Note Cisco IOS images with strong encryption (including, but not limited to, 56-bit data encryption feature sets) are subject to United States government export controls, and have a limited distribution. Images to be installed outside the United States require an export license. Customer orders might be denied or subject to delay due to United States government regulations. Contact your sales representative or distributor for more information, or send e-mail to export@cisco.com.

- SEAL—Software Encryption Algorithm. An alternative algorithm to software-based DES, 3DES, and AES. SEAL encryption uses a 160-bit encryption key and has a lower impact to the CPU when compared to other software-based algorithms.
- MD5 (HMAC variant)—MD5 (Message Digest 5) is a hash algorithm. HMAC is a keyed hash variant used to authenticate data.
- SHA (HMAC variant)—SHA (Secure Hash Algorithm) is a hash algorithm. HMAC is a keyed hash variant used to authenticate data.

IPSec as implemented in Cisco IOS software supports the following additional standards:

- AH—Authentication Header. A security protocol which provides data authentication and optional anti-replay services. AH is embedded in the data to be protected (a full IP datagram).
- ESP—Encapsulating Security Payload. A security protocol which provides data privacy services and optional data authentication, and anti-replay services. ESP encapsulates the data to be protected.

Supported Hardware, Switching Paths, and Encapsulation

IPSec has certain requirements for hardware, switching paths, and encapsulation methods as follows:

- Supported Hardware
- Supported Switching Paths

- Supported Encapsulation

Supported Hardware

This section contains the following subsections:

- ISA and ISM Support
- VPN Accelerator Module (VAM) Support
- AIMs and NM Support

ISA and ISM Support

For 7100 and 7200 hardware platforms, IPSec support requires the following adaptors or modules:

- Integrated Services Adapter (ISA) for the Cisco 7100 and 7200 series.
- Integrated Services Modules (ISM) for the Cisco 7100 series.

Note A VPN accelerator card and controller is also available on a Cisco 7100 and a Cisco 7200 series routers with an ISA and a Cisco 7100 series router with and ISM.

For more information on ISAs and ISMs, see the *Integrated Services Adapter and Integrated Services Module Installation and Configuration* publication.

VPN Accelerator Module (VAM) Support

The VAM is a single-width acceleration module. It provides high-performance, hardware-assisted tunneling and encryption services suitable for VPN remote access, site-to-site intranet, and extranet applications. It also provides platform scalability and security while working with all services necessary for successful VPN deployments—security, quality of service (QoS), firewall and intrusion detection, service-level validation, and management. The VAM off-loads IPSec processing from the main processor, thus freeing resources on the processor engines for other tasks.

The VAM provides hardware-accelerated support for the following multiple encryption functions:

- 56-bit DES standard mode: CBC
- 3-Key Triple DES (168-bit)
- SHA-1 and MD5
- Rivest, Shamir, Adelman (RSA) public-key algorithm
- Diffie-Hellman key exchange RC4-40

For more information on VAMs, see the document “VPN Acceleration Module (VAM).”

AIMs and NM Support

The data encryption Advanced Integration Module (AIM) and Network Module (NM) provide hardware-based encryption.

The data encryption AIMs and NM are hardware Layer 3 (IPSec) encryption modules and provide DES and Triple DES IPSec encryption for multiple T1s or E1s of bandwidth. These products also have hardware support for Diffie-Hellman, RSA, and DSA key generation.

Before using either module, note that RSA manual keying is not supported.

See Table 44 to determine which VPN encryption module to use.

IPPCP Software for Use with AIMs and NMs in Cisco 2600 and Cisco 3600 Series Routers

Software IPPCP with AIMs and NMs allow customers to use Lempel-Ziv-Stac (LZS) software compression with IPSec when a VPN module is in Cisco 2600 and Cisco 3600 series routers, allowing users to effectively increase the bandwidth on their interfaces.

Without IPPCP software, compression is not supported with the VPN encryption hardware AIM and NM; that is, a user had to remove the VPN module from the router and run software encryption with software compression. IPPCP enables all VPN modules to support LZS compression in software when the VPN module is in the router, thereby, allowing users to configure data compression and increase their bandwidth, which is useful for a low data link.

Without IPPCP, compression occurs at Layer 2, and encryption occurs at Layer 3. After a data stream is encrypted, it is passed on for compression services. When the compression engine receives the encrypted data streams, the data expands and does not compress. This feature enables both compression and encryption of the data to occur at Layer 3 by selecting LZS with the IPSec transform set; that is, LZS compression occurs before encryption, and it is able to get better compression ratio.

Table 44 AIM/VPN Encryption Module Support by Cisco IOS Release

<table>
<tr><th rowspan="2">Platform</th><th colspan="5">Encryption Module Support by Cisco IOS Release</th></tr>
<tr><th>12.2(13)T</th><th>12.3(4)T</th><th>12.3(5)</th><th>12.3(6)</th><th>12.3(7)T</th></tr>
<tr><td>Cisco 831</td><td colspan="5">Software-based AES</td></tr>
<tr><td>Cisco 1710
Cisco 1711
Cisco 1721
Cisco 1751
Cisco 1760</td><td colspan="5">Software-based AES</td></tr>
<tr><td>Cisco 2600 XM</td><td colspan="3">—</td><td colspan="2">AIM-VPN/BPII-Plus Hardware Encryption Module</td></tr>
<tr><td>Cisco 2611 XM
Cisco 2621 XM
Cisco 2651 XM</td><td>—</td><td colspan="3">AIM-VPN/BPII Hardware Encryption Module</td><td>AIM-VPN/BPII-Plus Hardware Encryption Module</td></tr>
<tr><td>Cisco 2691 XM</td><td colspan="4">AIM-VPN/EPII Hardware Encryption Module</td><td>AIM-VPN/EPII-Plus Hardware Encryption Module</td></tr>
<tr><td>Cisco 3735</td><td colspan="2">AIM-VPN/EPII Hardware Encryption Module</td><td colspan="3">AIM-VPN/EPII-Plus Hardware Encryption Module</td></tr>
<tr><td>Cisco 3660
Cisco 3745</td><td colspan="2">AIM-VPN/HPII Hardware Encryption Module</td><td colspan="3">AIM-VPN/HPII-Plus Hardware Encryption Module</td></tr>
</table>

For more information on AIMs and NM, see *Installing Advanced Integration Modules in Cisco 2600 Series, Cisco 3600 Series, and Cisco 3700 Series Routers*.

Supported Switching Paths

Table 45 lists the supported switching paths that work with IPSec.

Table 45 Supported Switching Paths for IPSec

Switching Paths	Examples
Process switching	`interface ethernet0/0` ` no ip route-cache`
Fast switching	`interface ethernet0/0` ` ip route-cache` `! Ensure that you will not hit flow switching.` ` no ip route-cache flow` `! Disable CEF for the interface, which supercedes global CEF.` ` no ip route-cache cef`
Cisco Express Forwarding (CEF)	`ip cef` `interface ethernet0/0` ` ip route-cache` `! Ensure that you will not hit flow switching.` ` no ip route-cache flow`
Fast-flow switching	`interface ethernet0/0` ` ip route-cache` `! Enable flow switching` `p route-cache flow` `! Disable CEF for the interface.` `  no ip route-cache cef`
CEF-flow switching	`! Enable global CEF.` `ip cef` `interface ethernet0/0` ` ip route-cache` ` ip route-cache flow` `! Enable CEF for the interface` ` ip route-cache cef`

Supported Encapsulation

IPSec works with the following serial encapsulations: High-Level Data-Links Control (HDLC), PPP, and Frame Relay.

IPSec also works with the Generic Routing Encapsulation (GRE) and IPinIP Layer 3, Layer 2 Forwarding (L2F), Layer 2 Tunneling Protocol (L2TP), Data Link Switching+ (DLSw+), and SRB tunneling protocols; however, multipoint tunnels are not supported. Other Layer 3 tunneling protocols may not be supported for use with IPSec.

Because the IPSec Working Group has not yet addressed the issue of group key distribution, IPSec currently cannot be used to protect group traffic (such as broadcast or multicast traffic).

IPSec Functionality Overview

IPSec provides the following network security services. (In general, local security policy will dictate the use of one or more of these services.)

- Data Confidentiality—The IPSec sender can encrypt packets before transmitting them across a network.

- Data Integrity—The IPSec receiver can authenticate packets sent by the IPSec sender to ensure that the data has not been altered during transmission.
- Data Origin Authentication—The IPSec receiver can authenticate the source of the IPSec packets sent. This service is dependent upon the data integrity service.
- Anti-Replay—The IPSec receiver can detect and reject replayed packets.

IPSec provides secure *tunnels* between two peers, such as two routers. You define which packets are considered sensitive and should be sent through these secure tunnels, and you define the parameters that should be used to protect these sensitive packets, by specifying characteristics of these tunnels. Then, when the IPSec peer recognizes such a sensitive packet, the peer sets up the appropriate secure tunnel and sends the packet through the tunnel to the remote peer. (The use of the term *tunnel* in this chapter does not refer to using IPSec in tunnel mode.)

More accurately, these *tunnels* are sets of security associations (SAs) that are established between two IPSec peers. The SAs define which protocols and algorithms should be applied to sensitive packets and specify the keying material to be used by the two peers. SAs are unidirectional and are established per security protocol (AH or ESP).

With IPSec you define what traffic should be protected between two IPSec peers by configuring access lists and applying these access lists to interfaces by way of crypto map sets. Therefore, traffic may be selected on the basis of source and destination address, and optionally Layer 4 protocol, and port. (The access lists used for IPSec are used only to determine which traffic should be protected by IPSec, not which traffic should be blocked or permitted through the interface. Separate access lists define blocking and permitting at the interface.)

A crypto map set can contain multiple entries, each with a different access list. The crypto map entries are searched in order—the router attempts to match the packet to the access list specified in that entry.

When a packet matches a **permit** entry in a particular access list, and the corresponding crypto map entry is tagged as **cisco**, connections are established if necessary. If the crypto map entry is tagged as **ipsec-isakmp**, IPSec is triggered. If no SA exists that IPSec can use to protect this traffic to the peer, IPSec uses IKE to negotiate with the remote peer to set up the necessary IPSec SAs on behalf of the data flow. The negotiation uses information specified in the crypto map entry as well as the data flow information from the specific access list entry. (The behavior is different for dynamic crypto map entries. See the section "Creating Dynamic Crypto Maps" section later in this module.)

If the crypto map entry is tagged as **ipsec-manual**, IPSec is triggered. If no SA exists that IPSec can use to protect this traffic to the peer, the traffic is dropped. In this case, the SAs are installed via the configuration, without the intervention of IKE. If the SAs did not exist, IPSec did not have all of the necessary pieces configured.

Once established, the set of SAs (outbound, to the peer) is then applied to the triggering packet and to subsequent applicable packets as those packets exit the router. "Applicable" packets are packets that match the same access list criteria that the original packet matched. For example, all applicable packets could be encrypted before being forwarded to the remote peer. The corresponding inbound SAs are used when processing the incoming traffic from that peer.

Multiple IPSec tunnels can exist between two peers to secure different data streams, with each tunnel using a separate set of SAs. For example, some data streams might be authenticated only while other data streams must both be encrypted and authenticated.

Access lists associated with IPSec crypto map entries also represent which traffic the router requires to be protected by IPSec. Inbound traffic is processed against the crypto map entries—if an unprotected packet matches a **permit** entry in a particular access list associated with an IPSec crypto map entry, that packet is dropped because it was not sent as an IPSec-protected packet.

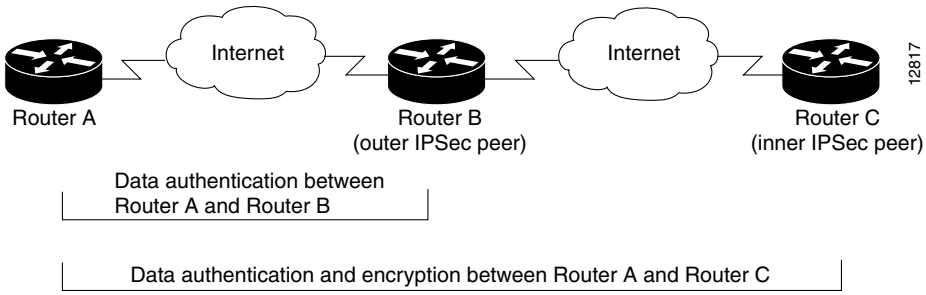
Crypto map entries also include transform sets. A transform set is an acceptable combination of security protocols, algorithms, and other settings to apply to IPSec protected traffic. During the IPSec SA negotiation, the peers agree to use a particular transform set when protecting a particular data flow.

IPSec Traffic Nested to Multiple Peers

You can nest IPSec traffic to a series of IPSec peers. For example, in order for traffic to traverse multiple firewalls (these firewalls have a policy of not letting through traffic that they have not authenticated), the router must establish IPSec tunnels with each firewall in turn. The "nearer" firewall becomes the "outer" IPSec peer.

In the example shown in Figure 54, Router A encapsulates the traffic destined for Router C in IPSec (Router C is the inner IPSec peer). However, before Router A can send this traffic, it must first reencapsulate this traffic in IPSec in order to send it to Router B (Router B is the "outer" IPSec peer).

Figure 54 Nesting Example of IPSec Peers



It is possible for the traffic between the "outer" peers to have one kind of protection (such as data authentication) and for traffic between the "inner" peers to have different protection (such as both data authentication and encryption).

How to Configure IPSec VPNs

Perform the tasks in the following sections to create IPSec VPNs:

- Creating Crypto Access Lists, page 912
- Defining Transform Sets: A Combination of Security Protocols and Algorithms, page 917
- Creating Crypto Map Sets, page 920
- Applying Crypto Map Sets to Interfaces, page 933

Creating Crypto Access Lists

To create crypto access lists that define which traffic is protected via IPSec tunnels, you should understand the following concepts:

- Crypto Access List Overview
- When to Use the permit and deny Keywords in Crypto Access Lists
- Mirror Image Crypto Access Lists at Each IPSec Peer

- When to Use the any Keyword in Crypto Access Lists

Crypto Access List Overview

Crypto access lists are used to define which IP traffic will be protected by crypto and which traffic will not be protected by crypto. (These access lists are *not* the same as regular access lists, which determine what traffic to forward or block at an interface.) For example, access lists can be created to protect all IP traffic between Subnet A and Subnet Y or Telnet traffic between Host A and Host B.

The access lists themselves are not specific to IPSec. It is the crypto map entry referencing the specific access list that defines whether IPSec processing is applied to the traffic matching a **permit** in the access list.

Crypto access lists associated with IPSec crypto map entries have four primary functions:

- Select outbound traffic to be protected by IPSec (permit = protect).
- Indicate the data flow to be protected by the new SAs (specified by a single **permit** entry) when initiating negotiations for IPSec security associations.
- Process inbound traffic in order to filter out and discard traffic that should have been protected by IPSec.
- Determine whether or not to accept requests for IPSec security associations on behalf of the requested data flows when processing IKE negotiation from the IPSec peer.
- Negotiation is performed only for **ipsec-isakmp** crypto map entries. In order to be accepted, if the peer initiates the IPSec negotiation, it must specify a data flow that is "permitted" by a crypto access list associated with an **ipsec-isakmp** crypto map entry.

If you want certain traffic to receive one combination of IPSec protection (for example, authentication only) and other traffic to receive a different combination of IPSec protection (for example, both authentication and encryption), you need to create two different crypto access lists to define the two different types of traffic. These different access lists are then used in different crypto map entries which specify different IPSec policies.

When to Use the permit and deny Keywords in Crypto Access Lists

Using the **permit** keyword causes all IP traffic that matches the specified conditions to be protected by crypto, using the policy described by the corresponding crypto map entry. Using the **deny** keyword prevents traffic from being protected by crypto in the context of that particular crypto map entry. (In other words, it does not allow the policy as specified in that crypto map entry to be applied to this traffic.) If this traffic is denied in all of the crypto map entries for that interface, the traffic is not protected by crypto.

The crypto access list you define will be applied to an interface after you define the corresponding crypto map entry and apply the crypto map set to the interface. Different access lists must be used in different entries of the same crypto map set. However, both inbound and outbound traffic will be evaluated against the same "outbound" IPSec access list. Therefore, the access list's criteria is applied in the forward direction to traffic exiting your router, and the reverse direction to traffic entering your router.

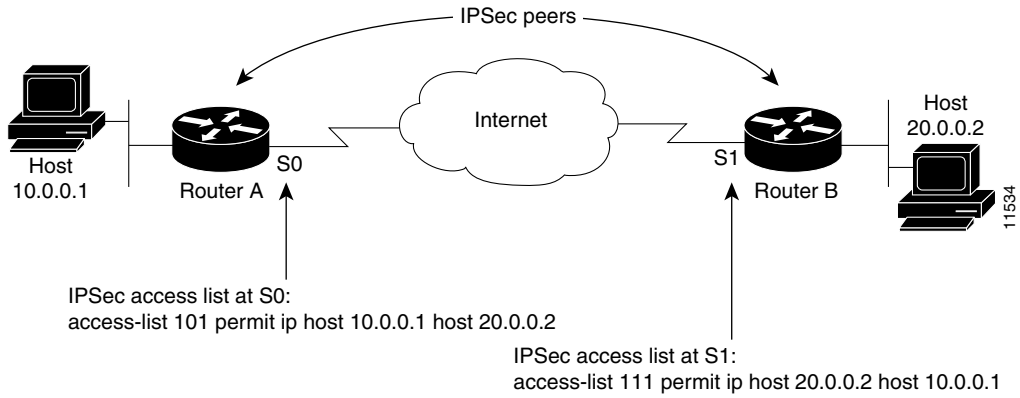
In Figure 55, IPSec protection is applied to traffic between Host 10.0.0.1 and Host 20.0.0.2 as the data exits Router A's S0 interface en route to Host 20.0.0.2. For traffic from Host 10.0.0.1 to Host 20.0.0.2, the access list entry on Router A is evaluated as follows:

```
source = host 10.0.0.1
dest = host 20.0.0.2
```

For traffic from Host 20.0.0.2 to Host 10.0.0.1, that same access list entry on Router A is evaluated as follows:

```
source = host 20.0.0.2
dest = host 10.0.0.1
```

Figure 55 How Crypto Access Lists Are Applied for Processing IPSec



Traffic exchanged between hosts 10.0.0.1 and 20.0.0.2
is protected between Router A S0 and Router B S1

If you configure multiple statements for a given crypto access list which is used for IPSec, in general the first **permit** statement that is matched will be the statement used to determine the scope of the IPSec SA. That is, the IPSec SA will be set up to protect traffic that meets the criteria of the matched statement only. Later, if traffic matches a different **permit** statement of the crypto access list, a new, separate IPSec SA will be negotiated to protect traffic matching the newly matched access list statement.

Any unprotected inbound traffic that matches a **permit** entry in the crypto access list for a crypto map entry flagged as IPSec will be dropped, because this traffic was expected to be protected by IPSec.

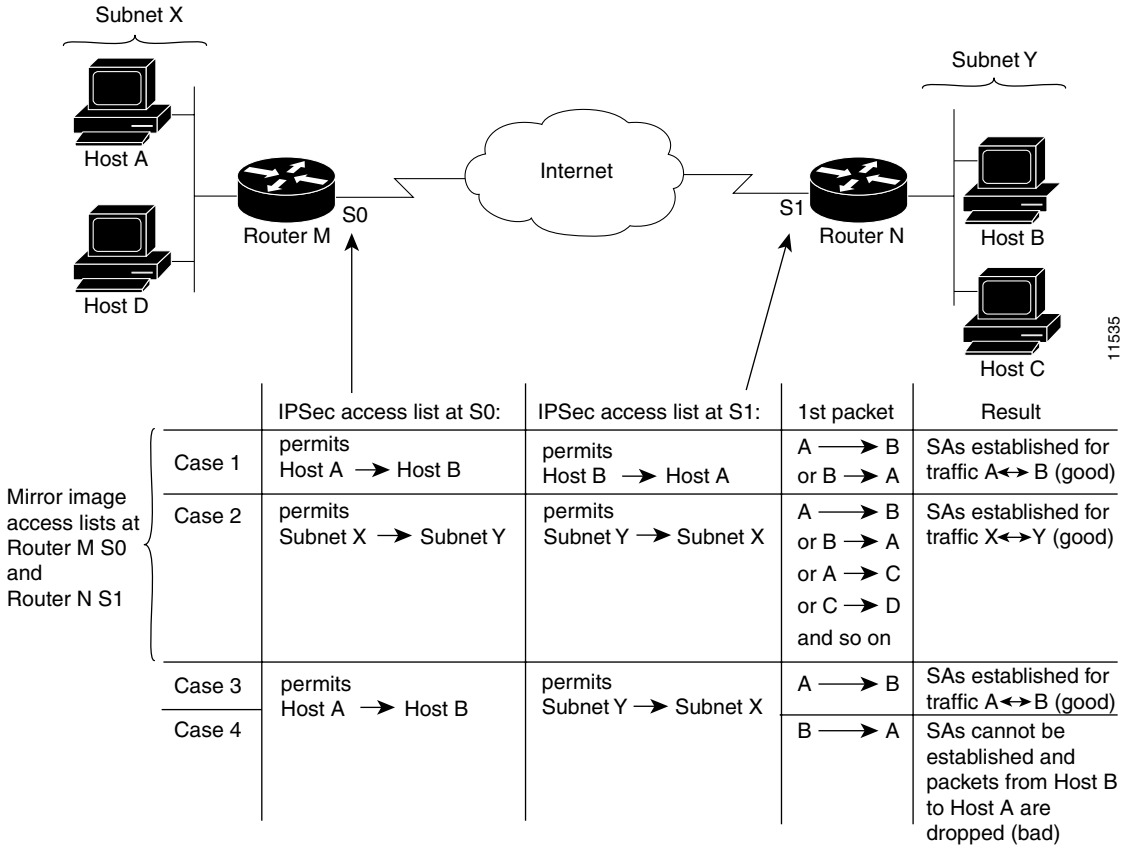
Note If you view your router's access lists by using a command such as **show ip access-lists**, *all* extended IP access lists will be shown in the command output. This includes extended IP access lists that are used for traffic filtering purposes as well as those that are used for crypto. The **show** command output does not differentiate between the different uses of the extended access lists.

Mirror Image Crypto Access Lists at Each IPSec Peer

Cisco recommends that for every crypto access list specified for a static crypto map entry that you define at the local peer, you define a "mirror image" crypto access list at the remote peer. This ensures that traffic that has IPSec protection applied locally can be processed correctly at the remote peer. (The crypto map entries themselves must also support common transforms and must refer to the other system as a peer.)

Figure 56 shows some sample scenarios when you have mirror image access lists and when you do not have mirror image access lists.

Figure 56 Mirror Image vs. Nonmirror Image Crypto Access Lists (for IPSec)



As Figure 56 indicates, IPSec SAs can be established as expected whenever the two peers' crypto access lists are mirror images of each other. However, an IPSec SA can be established only some of the time when the access lists are not mirror images of each other. This can happen in the case where an entry in one peer's access list is a subset of an entry in the other peer's access list, such as shown in Cases 3 and 4 of Figure 56. IPSec SA establishment is critical to IPSec—without SAs, IPSec does not work, causing any packets matching the crypto access list criteria to be silently dropped instead of being forwarded with IPSec.

In Figure 56, an SA cannot be established in Case 4. This is because SAs are always requested according to the crypto access lists at the initiating packet's end. In Case 4, Router N requests that all traffic between Subnet X and Subnet Y be protected, but this is a superset of the specific flows permitted by the crypto access list at Router M so the request is therefore not permitted. Case 3 works because Router M's request is a subset of the specific flows permitted by the crypto access list at Router N.

Because of the complexities introduced when crypto access lists are not configured as mirror images at peer IPSec devices, Cisco strongly encourages you to use mirror image crypto access lists.

When to Use the any Keyword in Crypto Access Lists

When you create crypto access lists, using the **any** keyword could cause problems. Cisco discourages the use of the **any** keyword to specify source or destination addresses.

The **any** keyword in a **permit** statement is discouraged when you have multicast traffic flowing through the IPSec interface; the **any** keyword can cause multicast traffic to fail.

The **permit any any** statement is strongly discouraged, because this will cause all outbound traffic to be protected (and all protected traffic sent to the peer specified in the corresponding crypto map entry) and will require protection for all inbound traffic. Then, all inbound packets that lack IPSec protection will be silently dropped, including packets for routing protocols, Network Time Protocol (NTP), echo, echo response, and so on.

You need to be sure you define which packets to protect. If you *must* use the **any** keyword in a **permit** statement, you must preface that statement with a series of **deny** statements to filter out any traffic (that would otherwise fall within that **permit** statement) that you do not want to be protected.

Also, use of **any** keyword in access control lists (ACLs) with reverse route injection (RRI) is not supported. (For more information on RRI, see the section "Creating Crypto Map Sets.")

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**log**]

   or

   ip access-list extended *name*
4. Repeat Step 3 for each crypto access list you want to create.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**log**] **Example:** `Router(config)# access-list 100 permit ip 10.0.68.0 0.0.0.255 10.1.1.0 0.0.0.255` or **ip access-list extended** *name* **Example:** `Router(config)# ip access-list extended vpn-tunnel`	Specifies conditions to determine which IP packets will be protected.[1] Enable or disable crypto for traffic that matches these conditions. **Tip** Cisco recommends that you configure "mirror image" crypto access lists for use by IPSec and that you avoid using the **any** keyword.
Step 4	—	Repeat Step 3 for each crypto access list you want to create.

1. You specify conditions using an IP access list designated by either a number or a name. The **access-list** command designates a numbered extended access list; the **ip access-list extended** command designates a named access list.

What to Do Next

After you have created at least one crypto access list, you should begin defining a transform set as described in the section "Defining Transform Sets: A Combination of Security Protocols and Algorithms."

Later, you will associate the crypto access lists to particular interfaces when you configure and apply crypto map sets to the interfaces (following instructions in the sections "Creating Crypto Map Sets" and "Applying Crypto Map Sets to Interfaces").

Defining Transform Sets: A Combination of Security Protocols and Algorithms

Perform this task to define a transform set that is to be used by the IPSec peers during IPSec security association negotiations with IKE.

Restrictions

If you are specifying SEAL encryption, note the following restrictions:

- Your router and the other peer must not have hardware IPSec encryption.
- Your router and the other peer must support IPSec.
- Your router and the other peer must support the k9 subsystem.
- SEAL encryption is available only on Cisco equipment. Therefore, interoperability is not possible.

About Transform Sets

A transform set represents a certain combination of security protocols and algorithms. During the IPSec SA negotiation, the peers agree to use a particular transform set for protecting a particular data flow.

You can specify multiple transform sets, and then specify one or more of these transform sets in a crypto map entry. The transform set defined in the crypto map entry will be used in the IPSec SA negotiation to protect the data flows specified by that crypto map entry's access list.

During IPSec security association negotiations with IKE, the peers search for a transform set that is the same at both peers. When such a transform set is found, it is selected and will be applied to the protected traffic as part of both peers' IPSec SAs. (With manually established SAs, there is no negotiation with the peer, so both sides must specify the same transform set.)

If you change a transform set definition, the change is only applied to crypto map entries that reference the transform set. The change will not be applied to existing security associations, but will be used in subsequent negotiations to establish new SAs. If you want the new settings to take effect sooner, you can clear all or part of the SA database by using the **clear crypto sa** command

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec transform-set** *transform-set-name transform1* [*transform2* [*transform3*]]

4. **mode** [**tunnel** | **transport**]
5. **exit**
6. **clear crypto sa** [**peer** {*ip-address* | **peer-name**} | **sa map** *map-name* | **sa entry** *destination-address protocol spi*]
7. **show crypto ipsec transform-set** [**tag** *transform-set-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec transform-set** *transform-set-name transform1* [*transform2* [*transform3*]] **Example:** `Router(config)# crypto ipsec transform-set aesset esp-aes 256 esp-sha-hmac`	Defines a transform set and enters crypto transform configuration mode. There are complex rules defining which entries you can use for the transform arguments. These rules are explained in the command description for the **crypto ipsec transform-set** command, and Table 46 provides a list of allowed transform combinations.
Step 4	**mode** [**tunnel** | **transport**] **Example:** `Router(cfg-crypto-tran)# mode transport`	(Optional) Changes the mode associated with the transform set. The mode setting is applicable only to traffic whose source and destination addresses are the IPSec peer addresses; it is ignored for all other traffic. (All other traffic is in tunnel mode only.)
Step 5	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

	Command or Action	Purpose
Step 6	**clear crypto sa** [**peer** {*ip-address* | *peer-name*} | **sa map** *map-name* | **sa entry** *destination-address protocol spi*] **Example:** `Router# clear crypto sa`	(Optional) Clears existing IPSec security associations so that any changes to a transform set will take effect on subsequently established security associations. Manually established SAs are reestablished immediately. • Using the **clear crypto sa** command without parameters will clear out the full SA database, which will clear out active security sessions. • You may also specify the **peer**, **map**, or **entry** keywords to clear out only a subset of the SA database.
Step 7	**show crypto ipsec transform-set** [**tag** *transform-set-name*] **Example:** `Router# show crypto ipsec transform-set`	(Optional) Displays the configured transform sets.

Table 46 shows allowed transform combinations.

Table 46 Allowed Transform Combinations

Transform Type	Transform	Description
AH Transform	**ah-md5-hmac**	AH with the MD5 (a Hash-based Message Authentication Code [HMAC] variant) authentication algorithm
	ah-sha-hmac	AH with the SHA (an HMAC variant) authentication algorithm
ESP Encryption Transform	**esp-aes**	ESP with the 128-bit AES encryption algorithm
	esp-aes 192	ESP with the 192-bit AES encryption algorithm
	esp-aes 256	ESP with the 256-bit AES encryption algorithm
	esp-des	ESP with the 56-bit DES encryption algorithm
	esp-3des	ESP with the 168-bit DES encryption algorithm (3DES or Triple DES)
	esp-null	Null encryption algorithm
	esp-seal	ESP with the 160-bit SEAL encryption algorithm.

Table 46 Allowed Transform Combinations (continued)

Transform Type	Transform	Description
ESP Authentication Transform	**esp-md5-hmac**	ESP with the MD5 (HMAC variant) authentication algorithm
	esp-sha-hmac	ESP with the SHA (HMAC variant) authentication algorithm
IP Compression Transform	**comp-lzs**	IP compression with the LZS algorithm

What to Do Next

After you have defined a transform set, you should create a crypto map as specified in the section "Creating Crypto Map Sets."

Creating Crypto Map Sets

See one of the following sections, as appropriate, to help create crypto map sets:

- Creating Static Crypto Maps
- Creating Dynamic Crypto Maps
- Creating Crypto Map Entries to Establish Manual SAs

Prerequisites

Before you create crypto map entries, you should determine which type of crypto map—static, dynamic, or manual—best addresses the needs of your network. You should also understand the following concepts:

- About Crypto Maps
- Load Sharing Among Crypto Maps
- Crypto Map Guidelines

About Crypto Maps

Crypto map entries created for IPSec pull together the various parts used to set up IPSec SAs, including:

- Which traffic should be protected by IPSec (per a crypto access list)
- The granularity of the flow to be protected by a set of SAs
- Where IPSec-protected traffic should be sent (who the remote IPSec peer is)
- The local address to be used for the IPSec traffic (See the section "Applying Crypto Map Sets to Interfaces" for more details.)
- What IPSec SA should be applied to this traffic (selecting from a list of one or more transform sets)
- Whether SAs are manually established or are established via IKE
- Other parameters that might be necessary to define an IPSec SA

How Crypto Maps Work

Crypto map entries with the same crypto map name (but different map sequence numbers) are grouped into a crypto map set. Later, you will apply these crypto map sets to interfaces; then, all IP traffic passing through the interface is evaluated against the applied crypto map set. If a crypto map entry sees outbound IP traffic that should be protected and the crypto map specifies the use of IKE, a SA is negotiated with the remote peer according to the parameters included in the crypto map entry; otherwise, if the crypto map entry specifies the use of manual SAs, an SA should have already been established via configuration. (If a dynamic crypto map entry sees outbound traffic that should be protected and no security association exists, the packet is dropped.)

The policy described in the crypto map entries is used during the negotiation of SAs. If the local router initiates the negotiation, it will use the policy specified in the static crypto map entries to create the offer to be sent to the specified IPSec peer. If the IPSec peer initiates the negotiation, the local router will check the policy from the static crypto map entries, as well as any referenced dynamic crypto map entries to decide whether to accept or reject the peer's request (offer).

For IPSec to succeed between two IPSec peers, both peers' crypto map entries must contain compatible configuration statements.

Compatible Crypto Maps: Establishing an SA

When two peers try to establish a SA, they must each have at least one crypto map entry that is compatible with one of the other peer's crypto map entries. For two crypto map entries to be compatible, they must at least meet the following criteria:

- The crypto map entries must contain compatible crypto access lists (for example, mirror image access lists). In the case where the responding peer is using dynamic crypto maps, the entries in the local crypto access list must be "permitted" by the peer's crypto access list.
- The crypto map entries must each identify the other peer (unless the responding peer is using dynamic crypto maps).
- The crypto map entries must have at least one transform set in common.

Load Sharing Among Crypto Maps

You can define multiple remote peers using crypto maps to allow for load sharing. Load sharing is useful because if one peer fails, there will still be a protected path. The peer that packets are actually sent to is determined by the last peer that the router heard from (received either traffic or a negotiation request from) for a given data flow. If the attempt fails with the first peer, IKE tries the next peer on the crypto map list.

If you are not sure how to configure each crypto map parameter to guarantee compatibility with other peers, you might consider configuring dynamic crypto maps as described in the section "Creating Dynamic Crypto Maps." Dynamic crypto maps are useful when the establishment of the IPSec tunnels is initiated by the remote peer (such as in the case of an IPSec router fronting a server). They are not useful if the establishment of the IPSec tunnels is locally initiated, because the dynamic crypto maps are policy templates, not complete statements of policy. (Although the access lists in any referenced dynamic crypto map entry are used for crypto packet filtering.)

Crypto Map Guidelines

You can apply only one crypto map set to a single interface. The crypto map set can include a combination of IPSec/IKE and IPSec/manual entries. Multiple interfaces can share the same crypto map set if you want to apply the same policy to multiple interfaces.

If you create more than one crypto map entry for a given interface, use the *seq-num* argument of each map entry to rank the map entries: the lower the *seq-num* argument, the higher the priority. At the interface that has the crypto map set, traffic is evaluated against higher priority map entries first.

You must create multiple crypto map entries for a given interface if any of the following conditions exist:

- If different data flows are to be handled by separate IPSec peers.
- If you want to apply different IPSec security to different types of traffic (to the same or separate IPSec peers); for example, if you want traffic between one set of subnets to be authenticated, and traffic between another set of subnets to be both authenticated and encrypted. In this case the different types of traffic should have been defined in two separate access lists, and you must create a separate crypto map entry for each crypto access list.
- If you are not using IKE to establish a particular set of security associations, and want to specify multiple access list entries, you must create separate access lists (one per **permit** entry) and specify a separate crypto map entry for each access list.

Creating Static Crypto Maps

When IKE is used to establish SAs, the IPSec peers can negotiate the settings they will use for the new security associations. This means that you can specify lists (such as lists of acceptable transforms) within the crypto map entry.

Perform this task to create crypto map entries that will use IKE to establish the SAs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* **ipsec-isakmp**
4. **match address** *access-list-id*
5. **set peer** {*hostname* | *ip-address*}
6. **set transform-set** *transform-set-name1* [*transform-set-name2...transform-set-name6*]
7. **set security-association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes*}
8. **set security-association level per-host**
9. **set pfs** [**group1** | **group2** | **group5**]
10. **exit**
11. **exit**
12. **show crypto map** [**interface** *interface* | **tag** *map-name*]

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* **ipsec-isakmp** **Example:** `Router(config)# crypto map static-map 1 ipsec-isakmp`	Names the crypto map entry to create (or modify), and enters crypto map configuration mode.
Step 4	**match address** *access-list-id* **Example:** `Router(config-crypto-m)# match address vpn-tunnel`	Names an extended access list. This access list determines which traffic should be protected by IPSec and which traffic should not be protected by IPSec security in the context of this crypto map entry.
Step 5	**set peer** {*hostname* | *ip-address*} **Example:** `Router(config-crypto-m)# set-peer 192.168.101.1`	Specifies a remote IPSec peer, the peer to which IPSec protected traffic can be forwarded. Repeat for multiple remote peers.
Step 6	**set transform-set** *transform-set-name1* [*transform-set-name2...transform-set-name6*] **Example:** `Router(config-crypto-m)# set transform-set aesset`	Specifies which transform sets are allowed for this crypto map entry. List multiple transform sets in order of priority (highest priority first).
Step 7	**set security-association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes*} **Example:** `Router (config-crypto-m)# set security-association lifetime seconds 2700`	(Optional) Specifies a SA lifetime for the crypto map entry. By default, the SAs of the crypto map are negotiated according to the global lifetimes.
Step 8	**set security-association level per-host** **Example:** `Router(config-crypto-m)# set security-association level per-host`	(Optional) Specifies that separate SAs should be established for each source and destination host pair. By default, a single IPSec "tunnel" can carry traffic for multiple source hosts and multiple destination hosts. ⚠ **Caution** Use this command with care, because multiple streams between given subnets can rapidly consume resources.

	Command	Purpose
Step 9	**set pfs** [**group1** | **group2** | **group 5**] **Example:** `Router(config-crypto-m)# set pfs group2`	(Optional) Specifies that IPSec either should ask for perfect forward secrecy (PFS) when requesting new SAs for this crypto map entry or should demand PFS in requests received from the IPSec peer. By default, PFS is not requested. If no group is specified with this command, group1 is used as the default.
Step 10	**exit** **Example:** `Router(config-crypto-m)# exit`	Exits crypto-map configuration mode.
Step 11	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 12	**show crypto map** [**interface** *interface* | **tag** *map-name*] **Example:** `Router# show crypto map`	Displays your crypto map configuration.

Troubleshooting Tips

Certain configuration changes will only take effect when negotiating subsequent SAs. If you want the new settings to take immediate effect, you must clear the existing SAs so that they will be re-established with the changed configuration. If the router is actively processing IPSec traffic, it is desirable to clear only the portion of the SA database that would be affected by the configuration changes (that is, clear only the SAs established by a given crypto map set). Clearing the full SA database should be reserved for large-scale changes, or when the router is processing very little other IPSec traffic.

To clear IPSec SAs, use the **clear crypto sa** command with appropriate parameters. (Omitting all parameters will clear out the full SA database, which will clear out active security sessions.)

What to Do Next

After you have successfully created a static crypto map, you must apply the crypto map set to each interface through which IPSec traffic flows. To complete this task, see the section "Applying Crypto Map Sets to Interfaces."

Creating Dynamic Crypto Maps

Dynamic crypto maps can ease IPSec configuration and are recommended for use with networks where the peers are not always predetermined. To create dynamic crypto maps, you should understand the following concepts:

- Dynamic Crypto Maps Overview
- Tunnel Endpoint Discovery (TED)

Dynamic Crypto Maps Overview

Dynamic crypto maps are only available for use by IKE.

A dynamic crypto map entry is essentially a static crypto map entry without all the parameters configured. It acts as a policy template where the missing parameters are later dynamically configured (as the result of an IPSec negotiation) to match a remote peer's requirements. This allows remote peers to exchange IPSec traffic with the router even if the router does not have a crypto map entry specifically configured to meet all of the remote peer's requirements.

Dynamic crypto maps are not used by the router to initiate new IPSec security associations with remote peers. Dynamic crypto maps are used when a remote peer tries to initiate an IPSec security association with the router. Dynamic crypto maps are also used in evaluating traffic.

A dynamic crypto map set is included by reference as part of a crypto map set. Any crypto map entries that reference dynamic crypto map sets should be the lowest priority crypto map entries in the crypto map set (that is, have the highest sequence numbers) so that the other crypto map entries are evaluated first; that way, the dynamic crypto map set is examined only when the other (static) map entries are not successfully matched.

If the router accepts the peer's request, at the point that it installs the new IPSec security associations it also installs a temporary crypto map entry. This entry is filled in with the results of the negotiation. At this point, the router performs normal processing, using this temporary crypto map entry as a normal entry, even requesting new security associations if the current ones are expiring (based upon the policy specified in the temporary crypto map entry). Once the flow expires (that is, all of the corresponding security associations expire), the temporary crypto map entry is then removed.

For both static and dynamic crypto maps, if unprotected inbound traffic matches a **permit** statement in an access list, and the corresponding crypto map entry is tagged as "IPSec," then the traffic is dropped because it is not IPSec-protected. (This is because the security policy as specified by the crypto map entry states that this traffic must be IPSec-protected.)

For static crypto map entries, if outbound traffic matches a **permit** statement in an access list and the corresponding SA is not yet established, the router will initiate new SAs with the remote peer. In the case of dynamic crypto map entries, if no SA existed, the traffic would simply be dropped (because dynamic crypto maps are not used for initiating new SAs).

Note Use care when using the **any** keyword in **permit** entries in dynamic crypto maps. If it is possible for the traffic covered by such a **permit** entry to include multicast or broadcast traffic, the access list should include **deny** entries for the appropriate address range. Access lists should also include **deny** entries for network and subnet broadcast traffic, and for any other traffic that should not be IPSec protected.

Restrictions for Dynamic Crypto Maps

Dynamic crypto map entries specify crypto access lists that limit traffic for which IPSec SAs can be established. A dynamic crypto map entry that does not specify an access list will be ignored during traffic filtering. A dynamic crypto map entry with an empty access list causes traffic to be dropped. If there is only one dynamic crypto map entry in the crypto map set, it must specify acceptable transform sets.

Tunnel Endpoint Discovery (TED)

Defining a dynamic crypto map allows only the receiving router to dynamically determine an IPSec peer. TED allows the initiating router to dynamically determine an IPSec peer for secure IPSec communications.

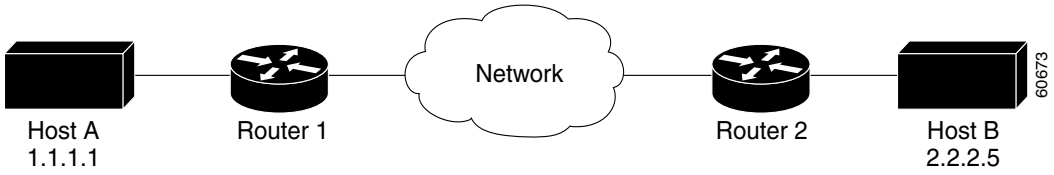
Dynamic TED helps to simplify IPSec configuration on the individual routers within a large network. Each node has a simple configuration that defines the local network that the router is protecting and the required IPSec transforms.

To have a large, fully-meshed network *without* TED, each peer needs to have static crypto maps to every other peer in the network. For example, if there are 100 peers in a large, fully-meshed network, each router needs 99 static crypto maps for each of its peers. With TED, only a single dynamic crypto map with TED enabled is needed because the peer is discovered dynamically. Thus, static crypto maps do not need to be configured for each peer.

Note TED helps only in discovering peers; otherwise, TED does not function any differently than normal IPSec. TED does not improve the scalability of IPSec (in terms of performance or the number of peers or tunnels).

Figure 57 and the corresponding steps explain a sample TED network topology.

Figure 57 Tunnel Endpoint Discovery Sample Network Topology



Step 1 Host A sends a packet that is destined for Host B.

Step 2 Router 1 intercepts and reads the packet. According to the IKE policy, Router 1 contains the following information: the packet must be encrypted, there are no SAs for the packet, and TED is enabled. Thus, Router 1 drops the packet and sends a TED probe into the network. (The TED probe contains the IP address of Host A (as the source IP address) and the IP address of Host B (as the destination IP address) embedded in the payload.

Step 3 Router 2 intercepts the TED probe and checks the probe against the ACLs that it protects; after the probe matches an ACL, it is recognized as a TED probe for proxies that the router protects. It then sends a TED reply with the IP address of Host B (as the source IP address) and the IP address of Host A (as the destination IP address) embedded in the payload.

Step 4 Router 1 intercepts the TED reply and checks the payloads for the IP address and half proxy of Router 2. It then combines the source side of its proxy with the proxy found in the second payload and initiates an IKE session with Router 2; thereafter, Router 1 initiates an IPSec session with Router 2.

Note IKE cannot occur until the peer is identified.

TED Versions

The following table lists the available TED versions:

Version	First Available Release	Description
TEDv1	12.0(5)T	Performs basic TED functionality on nonredundant networks.
TEDv2	12.1M	Enhanced to work with redundant networks with paths through multiple security gateways between the source and the destination.
TEDv3	12.2M	Enhanced to allow non-IP-related entries to be used in the access list.

TED Restrictions

TED has the following restrictions:

- It is Cisco proprietary.
- It is available only on dynamic crypto maps. (The dynamic crypto map template is based on the dynamic crypto map performing peer discovery. Although there are no access-list restrictions on the dynamic crypto map template, the dynamic crypto map template should cover data sourced from the protected traffic and the receiving router using the **any** keyword. When using the **any** keyword, include explicit **deny** statements to exempt routing protocol traffic prior to entering the **permit any** command.)
- TED works only in tunnel mode; that is, it does not work in transport mode.
- It is limited by the performance and scalability of limitation of IPSec on each individual platform.

Note Enabling TED slightly decreases the general scalability of IPSec because of the set-up overhead of peer discovery, which involves an additional "round-trip" of IKE messages (TED probe and reply). Although minimal, the additional memory used to store data structures during the peer discovery stage adversely affects the general scalability of IPSec.

- The IP addresses must be able to be routed within the network.
- The access list used in the crypto map for TED can only contain IP-related entries—TCP, UDP, or any other protocol cannot be used in the access list.

Note This restriction is no longer applicable in TEDv3.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto dynamic-map** *dynamic-map-name dynamic-seq-num*
4. **set transform-set** *transform-set-name1* [*transform-set-name2...transform-set-name6*]
5. **match address** *access-list-id*
6. **set peer** {*hostname* | *ip-address*}

7. **set security-association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes*}
8. **set pfs** [**group1** | **group2** | **group5**]
9. **exit**
10. **exit**
11. **show crypto dynamic-map** [**tag** *map-name*]
12. **configure terminal**
13. **crypto map** *map-name seq-num* **ipsec-isakmp dynamic** *dynamic-map-name* [**discover**]

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto dynamic-map** *dynamic-map-name dynamic-seq-num* **Example:** `Router(config)# crypto dynamic-map test-map 1`	Creates a dynamic crypto map entry and enters crypto map configuration mode.
Step 4	**set transform-set** *transform-set-name1* [*transform-set-name2...transform-set-name6*] **Example:** `Router(config-crypto-m)# set transform-set aesset`	Specifies which transform sets are allowed for the crypto map entry. List multiple transform sets in order of priority (highest priority first). This is the only configuration statement required in dynamic crypto map entries.

	Command	Purpose
Step 5	**match address** *access-list-id* **Example:** `Router(config-crypto-m)# match address 101`	(Optional) Accesses list number or name of an extended access list. This access list determines which traffic should be protected by IPSec and which traffic should not be protected by IPSec security in the context of this crypto map entry. **Note** Although access-lists are optional for dynamic crypto maps, they are highly recommended. If this is configured, the data flow identity proposed by the IPSec peer must fall within a **permit** statement for this crypto access list. If this is not configured, the router will accept any data flow identity proposed by the IPSec peer. However, if this is configured but the specified access list does not exist or is empty, the router will drop all packets. This is similar to static crypto maps because they also require that an access list be specified. Care must be taken if the **any** keyword is used in the access list, because the access list is used for packet filtering as well as for negotiation. You must configure a match address; otherwise, the behavior is not secure, and you cannot enable TED because packets are sent in the clear (unencrypted.)
Step 6	**set peer** {*hostname* | *ip-address*} **Example:** `Router(config-crypto-m)# set peer 192.168.101.1`	(Optional) Specifies a remote IPSec peer. Repeat for multiple remote peers. **Note** This is rarely configured in dynamic crypto map entries. Dynamic crypto map entries are often used for unknown remote peers.
Step 7	**set security-association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes*} **Example:** `Router (config-crypto-m)# set security-association lifetime seconds 7200`	(Optional) Overrides (for a particular crypto map entry) the global lifetime value, which is used when negotiating IP Security SAs.
Step 8	**set pfs** [**group1** | **group2** | **group5**] **Example:** `Router(config-crypto-m)# set pfs group2`	(Optional) Specifies that IPSec should ask for PFS when requesting new security associations for this crypto map entry or should demand PFS in requests received from the IPSec peer. By default, PFS is not requested. If no group is specified with this command, **group1** is used as the default.

	Command	Purpose
Step 9	**exit** **Example:** `Router(config-crypto-m)# exit`	Exits crypto-map configuration mode and returns to global configuration mode.
Step 10	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 11	**show crypto dynamic-map** [**tag** *map-name*] **Example:** `Router# show crypto dynamic-map`	(Optional) Displays information about dynamic crypto maps.
Step 12	**configure terminal** **Example:** `Router# configure terminal`	Returns to global configuration mode.
Step 13	**crypto map** *map-name seq-num* **ipsec-isakmp dynamic** *dynamic-map-name* [**discover**] **Example:** `Router(config)# crypto map static-map 1 ipsec-isakmp dynamic test-map discover`	(Optional) Adds a dynamic crypto map to a crypto map set. You should set the crypto map entries referencing dynamic maps to the lowest priority entries in a crypto map set. **Note** You must issue the **discover** keyword to enable TED.

Troubleshooting Tips

Certain configuration changes will only take effect when negotiating subsequent SAs. If you want the new settings to take immediate effect, you must clear the existing SAs so that they will be reestablished with the changed configuration. If the router is actively processing IPSec traffic, it is desirable to clear only the portion of the SA database that would be affected by the configuration changes (that is, clear only the SAs established by a given crypto map set). Clearing the full SA database should be reserved for large-scale changes, or when the router is processing very little other IPSec traffic.

To clear IPSec SAs, use the **clear crypto sa** command with appropriate parameters. (Omitting all parameters will clear out the full SA database, which will clear out active security sessions.)

What to Do Next

After you have successfully created a crypto map set, you must apply the crypto map set to each interface through which IPSec traffic flows. To complete this task, see the section "Applying Crypto Map Sets to Interfaces."

Creating Crypto Map Entries to Establish Manual SAs

The use of manual security associations is a result of a prior arrangement between the users of the local router and the IPSec peer. The two parties may begin with manual SAs and then move to using SAs established via IKE, or the remote party's system may not support IKE. If IKE is not used for establishing the SAs, there is no negotiation of SAs, so the configuration information in both systems must be the same in order for traffic to be processed successfully by IPSec.

The local router can simultaneously support manual and IKE-established SAs, even within a single crypto map set.

There is very little reason to disable IKE on the local router (unless the router only supports manual SAs, which is unlikely).

Note Access lists for crypto map entries tagged as **ipsec-manual** are restricted to a single **permit** entry and subsequent entries are ignored. In other words, the SAs established by that particular crypto map entry are only for a single data flow. To be able to support multiple manually established SAs for different kinds of traffic, define multiple crypto access lists, and then apply each one to a separate **ipsec-manual** crypto map entry. Each access list should include one **permit** statement defining what traffic to protect.

To create crypto map entries to establish manual SAs (that is, when IKE is not used to establish the SAs), perform this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* **ipsec-manual**
4. **match address** *access-list-id*
5. **set peer** {*hostname* | *ip-address*}
6. **set transform-set** *transform-set-name*
7. **set session-key inbound ah** *spi hex-key-string*

   or

   set session-key outbound ah *spi hex-key-string*
8. **set session-key inbound esp** *spi* **cipher** *hex-key-string* [**authenticator** *hex-key-string*]

   or

   set session-key outbound esp *spi* **cipher** *hex-key-string* [**authenticator** *hex-key-string*]
9. **exit**
10. **exit**
11. **show crypto map** [**interface** *interface* | **tag** *map-name*]

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* **ipsec-manual** **Example:** `Router(config)# crypto map mymap 10 ipsec-manual`	Specifies the crypto map entry to create or modify and enters crypto map configuration mode.
Step 4	**match address** *access-list-id* **Example:** `Router(config-crypto-m)# match address 102`	Names an IPSec access list that determines which traffic should be protected by IPSec and which traffic should not be protected by IPSec in the context of this crypto map entry. (The access list can specify only one **permit** entry when IKE is not used.)
Step 5	**set peer** *{hostname | ip-address}* **Example:** `Router(config-crypto-m)# set peer 10.0.0.5`	Specifies the remote IPSec peer. This is the peer to which IPSec protected traffic should be forwarded. (Only one peer can be specified when IKE is not used.)
Step 6	**set transform-set** *transform-set-name* **Example:** `Router(config-crypto-m)# set transform-set someset`	Specifies which transform set should be used. This must be the same transform set that is specified in the remote peer's corresponding crypto map entry. **Note** Only one transform set can be specified when IKE is not used.
Step 7	**set session-key inbound ah** *spi hex-key-string* **Example:** `Router(config-crypto-m)# set session-key inbound ah 256 9876543210987654987654321098765 4` and **set session-key outbound ah** *spi hex-key-string* **Example:** `Router(config-crypto-m)# set session-key outbound ah 256 fedcbafedcbafedcfedcbafedcbafedc`	Sets the AH security parameter indexes (SPIs) and keys to apply to inbound and outbound protected traffic if the specified transform set includes the AH protocol. (This manually specifies the AH security association to be used with protected traffic.)

	Command	Purpose
Step 8	**set session-key inbound esp** *spi* **cipher** *hex-key-string* [**authenticator** *hex-key-string*] **Example:** `Router(config-crypto-m)# set session-key inbound esp 256 cipher 0123456789012345` and **set session-key outbound esp** *spi* **cipher** *hex-key-string* [**authenticator** *hex-key-string*] **Example:** `Router(config-crypto-m)# set session-key outbound esp 256 cipher abcdefabcdefabcd`	Sets the ESP SPIs and keys to apply to inbound and outbound protected traffic if the specified transform set includes the ESP protocol. Specifies the cipher keys if the transform set includes an ESP cipher algorithm. Specifies the authenticator keys if the transform set includes an ESP authenticator algorithm. (This manually specifies the ESP security association to be used with protected traffic.)
Step 9	**exit** **Example:** `Router(config-crypto-m)# exit`	Exits crypto-map configuration mode and returns to global configuration mode.
Step 10	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 11	**show crypto map** [**interface** *interface* | **tag** *map-name*] **Example:** `Router# show crypto map`	Displays your crypto map configuration.

Troubleshooting Tips

For manually established SAs, you must clear and reinitialize the SAs for the changes to take effect. To clear IPSec SAs, use the **clear crypto sa** command with appropriate parameters. (Omitting all parameters will clear out the full SA database, which will clear out active security sessions.)

What to Do Next

After you have successfully created a crypto map set, you must apply the crypto map set to each interface through which IPSec traffic flows. To complete this task, see the section "Applying Crypto Map Sets to Interfaces."

Applying Crypto Map Sets to Interfaces

You need to apply a crypto map set to each interface through which IPSec traffic will flow. Applying the crypto map set to an interface instructs the router to evaluate all the interface's traffic against the crypto map set and to use the specified policy during connection or security association negotiation on behalf of traffic to be protected by crypto.

Perform this task to apply a crypto map to an interface.

Redundant Interfaces Sharing the Same Crypto Map

For redundancy, you could apply the same crypto map set to more than one interface. The default behavior is as follows:

- Each interface will have its own piece of the security association database.
- The IP address of the local interface will be used as the local address for IPSec traffic originating from or destined to that interface.

If you apply the same crypto map set to multiple interfaces for redundancy purposes, you need to specify an identifying interface. One suggestion is to use a loopback interface as the identifying interface. This has the following effects:

- The per-interface portion of the IPSec security association database will be established one time and shared for traffic through all the interfaces that share the same crypto map.
- The IP address of the identifying interface will be used as the local address for IPSec traffic originating from or destined to those interfaces sharing the same crypto map set.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **crypto map** *map-name*
5. **exit**
6. **crypto map** *map-name* **local-address** *interface-id*
7. **exit**
8. **show crypto map** [**interface** *interface*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# Interface FastEthernet 0/0`	Configures an interface and enters interface configuration mode.
Step 4	**crypto map** *map-name* **Example:** `Router(config-if)# crypto map mymap`	Applies a crypto map set to an interface.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**crypto map** *map-name* **local-address** *interface-id* **Example:** `Router(config)# crypto map mymap local-address loopback0`	(Optional) Permits redundant interfaces to share the same crypto map using the same local identity.
Step 7	**exit** **Example:** `Router(config)# exit`	(Optional) Exits global configuration mode.
Step 8	**show crypto map** [**interface** *interface*] **Example:** `Router# show crypto map`	(Optional) Displays your crypto map configuration

Configuration Examples for Configuring an IPSec VPN

This section contains the following configuration example:

- AES-Based Static Crypto Map: Example, page 935

AES-Based Static Crypto Map: Example

The following example is a portion of the **show running-config** command. This example shows how to configure a static crypto map and define AES as the encryption method.

```
crypto isakmp policy 10
 encryption aes 256
 authentication pre-share
 lifetime 180

crypto isakmp key cisco123 address 10.0.110.1
!
!
crypto ipsec transform-set aesset esp-aes 256 esp-sha-hmac
 mode transport
!
crypto map aesmap 10 ipsec-isakmp
 set peer 10.0.110.1
 set transform-set aesset
 match address 120
!
!
!
voice call carrier capacity active
!
!
```

```
!
!
!
!
!
!
!
mta receive maximum-recipients 0
!
!
!
!
interface FastEthernet0/0
 ip address 10.0.110.2 255.255.255.0
 ip nat outside
 no ip route-cache
 no ip mroute-cache
 duplex auto
 speed auto
 crypto map aesmap
!
interface Serial0/0
 no ip address
 shutdown
!
interface FastEthernet0/1
 ip address 11.0.110.1 255.255.255.0
 ip nat inside
 no ip route-cache
 no ip mroute-cache
 duplex auto
 speed auto
!
ip nat inside source list 110 interface FastEthernet0/0 overload
ip classless
ip route 0.0.0.0 0.0.0.0 10.5.1.1
ip route 12.0.110.0 255.255.255.0 FastEthernet0/0
ip route 172.18.124.0 255.255.255.0 10.5.1.1
ip route 172.18.125.3 255.255.255.255 10.5.1.1
ip http server
!
!
access-list 110 deny   ip 11.0.110.0 0.0.0.255 12.0.110.0 0.0.0.255
access-list 110 permit ip 11.0.110.0 0.0.0.255 any
access-list 120 permit ip 11.0.110.0 0.0.0.255 12.0.110.0 0.0.0.255
!
```

Additional References

The following sections provide references related to IPSec VPN configuration.

Related Documents

Related Topic	Document Title
IKE configuration	"Configuring IKE for IPSec VPNs" module
IKE, IPSec, and PKI configuration commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*, Release 12.4

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
• CISCO-IPSEC-FLOW-MONITOR- MIB • CISCO-IPSEC-MIB • CISCO-IPSEC-POLICY-MAP-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2401	*Security Architecture for the Internet Protocol*
RFC 2402	*IP Authentication Header*
RFC 2403	*The Use of HMAC-MD5-96 within ESP and AH*
RFC 2404	*The Use of HMAC-SHA-1-96 within ESP and AH*
RFC 2405	*The ESP DES-CBC Cipher Algorithm With Explicit IV*
RFC 2406	*IP Encapsulating Security Payload (ESP)*
RFC 2407	*The Internet IP Security Domain of Interpretation for ISAKMP*
RFC 2408	*Internet Security Association and Key Management Protocol (ISAKMP)*

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Glossary

anti-replay—Security service where the receiver can reject old or duplicate packets to protect itself against replay attacks. IPSec provides this optional service by use of a sequence number combined with the use of data authentication. Cisco IOS IPSec provides this service whenever it provides the data authentication service, except for manually established SAs (that is, SAs established by configuration and not by IKE).

data authentication—Verification of the integrity and origin of the data.

Data authentication can refer either to integrity alone or to both of these concepts (although data origin authentication is dependent upon data integrity).

data confidentiality—Security service in which the protected data cannot be observed.

data flow—Grouping of traffic, identified by a combination of source address or mask; destination address or mask; IP next protocol field, and source and destination ports, where the protocol and port fields can have the values of **any**. IPSec protection is applied to data flows.

peer—In the context of this module, a "peer" is a router or other device that participates in IPSec.

PFS—perfect forward secrecy. Cryptographic characteristic associated with a derived shared secret value. With PFS, if one key is compromised, previous and subsequent keys are not compromised, because subsequent keys are not derived from previous keys.

SA—security association. Description of how two or more entities will use security services in the context of a particular security protocol (AH or ESP) to communicate securely on behalf of a particular data flow. The transform and the shared secret keys are used for protecting the traffic.

SPI—security parameter index. A number which, together with a destination IP address and security protocol, uniquely identifies a particular security association. Without IKE, the SPI is manually specified for each security association.

transform—List of operations performed on a dataflow to provide data authentication, data confidentiality, and data compression. For example, one transform is the ESP protocol with the HMAC-MD5 authentication algorithm; another transform is the AH protocol with the 56-bit DES encryption algorithm and the ESP protocol with the HMAC-SHA authentication algorithm.

tunnel—In the context of this module, "tunnel" is a secure communication path between two peers, such as two routers. It does not refer to using IPSec in tunnel mode.

Note Refer to *Internetworking Terms and Acronyms* for terms not included in this glossary.

Feature Information for Security for VPNs with IPSec

Table 47 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for a specific command was introduced, see the command reference documentation.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Table 47 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 47 Feature Information for Configuring Security for IPSec VPNs

Feature Name	Software Releases	Feature Configuration Information
Advanced Encryption Standard (AES)	12.2(8)T	This feature adds support for the new encryption standard AES, which is a privacy transform for IPSec and IKE and has been developed to replace DES. The following sections provide information about this feature: • Supported Standards • Defining Transform Sets: A Combination of Security Protocols and Algorithms The following commands were modified by this feature: **crypto ipsec transform-set**, **encryption (IKE policy)**, **show crypto ipsec transform-set**, **show crypto isakmp policy**
DES/3DES/AES VPN Encryption Module (AIM-VPN/EPII, AIM-VPN/HPII, AIM-VPN/BPII Family)	12.3(7)T	This feature describes which VPN encryption hardware AI) and NM are supported in certain Cisco IOS software releases. The following section provides information about this feature: • AIMs and NM Support

Table 47 Feature Information for Configuring Security for IPSec VPNs (continued)

Feature Name	Software Releases	Feature Configuration Information
SEAL Encryption	12.3(7)T	This feature adds support for SEAL encryption in IPSec. The following section provides information about this feature: • Supported Standards • Defining Transform Sets: A Combination of Security Protocols and Algorithms The following command was modified by this feature: **crypto ipsec transform-set**
Software IPPCP (LZS) with Hardware Encryption	12.2(13)T	This feature allows customers to use LZS software compression with IPSec when a VPN module is in Cisco 2600 and Cisco 3600 series routers. The following section provides information about this feature: • AIMs and NM Support

Ability to Disable Extended Authentication for Static IPSec Peers

Feature History

Release	Modification
12.2(4)T	This feature was introduced.

This feature module describes the Ability to Disable Extended Authentication for Static IPSec Peers feature in Cisco IOS Release 12.2(4)T. It includes the following sections:

- Feature Overview, page 941
- Supported Platforms, page 942
- Supported Standards, MIBs, and RFCs, page 943
- Configuration Tasks, page 944
- Configuration Examples, page 945
- Command Reference, page 945

Feature Overview

The Ability to Disable Extended Authentication for Static IPSec Peers feature allows users to disable extended authentication (Xauth), preventing the routers from being prompted for Xauth information—username and password.

Without the ability to disable Xauth, a user cannot select which peer on the same crypto map should use Xauth. That is, if a user has router-to-router IP security (IPSec) on the same crypto map as a virtual private network (VPN)-client-to-Cisco-IOS IPSec, both peers are prompted for a username and password. In addition, a remote static peer (a Cisco IOS router) cannot establish an Internet Key Exchange (IKE) security association (SA) with the local Cisco IOS router. (Xauth is not an optional exchange, so if a peer does not respond to an Xauth request, the IKE SA is deleted.) Thus, the same interface cannot be used to terminate IPSec to VPN clients (that need Xauth) as well as other Cisco IOS routers (that cannot respond to Xauth) unless this feature is implemented.

Benefits

If VPN-client-to-Cisco-IOS IPSec and router-to-router IPSec exist on a single interface, the Ability to Disable Extended Authentication for Static IPSec Peers feature allows a user to disable Xauth while configuring the preshared key for router-to-router IPSec. Thus, the router will not prompt the peer for a username and password, which are transmitted when Xauth occurs for VPN-client-to-Cisco-IOS IPSec.

Restrictions

Xauth can be disabled only if preshared keys are used as the authentication mechanism for the given crypto map.

Related Documents

- "Configuring Internet Key Exchange Security Protocol" chapter in the *Cisco IOS Security Configuration Guide*, Release 12.2.
- "Internet Key Exchange Security Protocol Commands" chapter in the *Cisco IOS Security Command Reference*, Release 12.2.

Supported Platforms

- Cisco 800 series
- Cisco 1700 series
- Cisco 2500 series
- Cisco 2600 series
- Cisco 3620
- Cisco 3640
- Cisco 3660
- Cisco 7100 series
- Cisco 7200 series
- Cisco uBR905 Cable Access Router
- Cisco uBR925 Cable Access Router

Platform Support Through Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Feature Navigator. Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image.

To access Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Feature Navigator is updated when major Cisco IOS software releases and technology releases occur. As of May 2001, Feature Navigator supports M, T, E, S, and ST releases. You can access Feature Navigator at the following URL:

http://www.cisco.com/go/fn

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

No new or modified RFCs are supported by this feature.

Prerequisites

Before you can disable Xauth for static IPSec peers, you must complete the following tasks:

- Configure authentication, authorization, and accounting (AAA).

  For information on completing this task, refer to the AAA chapters of the *Cisco IOS Security Configuration Guide*, Release 12.2.

  Note Configuring AAA is required only if the VPN-client-to-Cisco-IOS is using AAA authentication.

- Configure an IPSec transform.

  For information on completing this task, refer to the section “Defining Transform Sets” in the chapter “Configuring IPSec Network Security” of the *Cisco IOS Security Configuration Guide*, Release 12.2.

- Configure a static crypto map.

  For information on completing this task, refer to the section "Creating Crypto Map Entries" in the chapter "Configuring IPSec Network Security" of the *Cisco IOS Security Configuration Guide*, Release 12.2.

- Configure ISAKMP policy.

  For information on completing this task, refer to the section "Creating Policies" in the chapter "Configuring Internet Key Exchange Security Protocol" of the *Cisco IOS Security Configuration Guide*, Release 12.2.

Configuration Tasks

See the following sections for configuration tasks for the Ability to Disable Extended Authentication for Static IPSec Peers feature. Each task in the list is identified as either required or optional.

- Disabling Xauth for Static IPSec Peers (required)
- Verifying Disabled Xauth for Static IPSec Peers (optional)

Disabling Xauth for Static IPSec Peers

To disable Xauth for router-to-router IPSec, use the following command in global configuration mode:

Command	Purpose
`Router(config)# crypto isakmp key` *keystring* `address` *peer-address* `[`*mask*`] [no-xauth]`	Configures a preshared authentication key. Use the **no-xauth** keyword if router-to-router IPSec is on the same crypto map as VPN-client-to-Cisco IOS IPSec. This keyword prevents the router from prompting the peer for Xauth information. You must configure the local and remote peer for preshared keys. **Note** According to the design of preshared key authentication in IKE main mode, preshared keys *must* be based on the IP address of the peers. Although you can send **hostname** as the identity of preshared key authentication, the key is searched on the IP address of the peer; if the key is not found (based on the IP address) the negotiation will fail.

Verifying Disabled Xauth for Static IPSec Peers

To verify your configuration, use the **show running-config** command in EXEC mode.

Configuration Examples

This section provides the following configuration example:

- Disabling Xauth for Static IPSec Peers Configuration

Disabling Xauth for Static IPSec Peers Configuration

The following example shows how the local peer specifies the preshared key, designates the remote peer by its IP address, and disables Xauth:

```
crypto isakmp key sharedkeystring address 172.21.230.33 no-xauth
```

Command Reference

The following modified command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **crypto isakmp key**

Cisco Easy VPN Remote

This document provides information on configuring and monitoring the Cisco Easy VPN Remote feature to create IPSec Virtual Private Network (VPN) tunnels between a supported router and an Easy VPN server (Cisco IOS router, VPN 3000 concentrator, or Cisco PIX Firewall) that supports this form of IPSec encryption and decryption.

For the benefits of this feature, see the section "Benefits of the Cisco Easy VPN Remote Feature."

Feature History for Cisco Easy VPN Remote

Release	Modification
12.2(4)YA	Support for Cisco Easy VPN Remote (Phase I) of this feature was introduced for Cisco 806, Cisco 826, Cisco 827, and Cisco 828 routers; Cisco 1700 series routers; and Cisco uBR905 and Cisco uBR925 cable access routers.
12.2(13)T	Cisco Easy VPN Remote was integrated into Cisco IOS Release 12.2(13)T.
12.2(8)YJ	Support for Cisco Easy VPN Remote (Phase II) of this feature was introduced for Cisco 806, Cisco 826, Cisco 827, and Cisco 828 routers; Cisco 1700 series routers; and Cisco uBR905 and Cisco uBR925 cable access routers.
12.2(15)T	The Cisco Easy VPN Remote (Phase II) feature was integrated into Cisco IOS Release 12.2(15)T. Support for the Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers was added.
12.3(2)T	The Type 6 Password in the IOS Configuration feature was added.
12.3(4)T	The Save Password and Multiple Peer Backup features were added.
12.3(7)T	The IPSec Dead Peer Detection Periodic Message Option feature was added.
12.3(7)XR	The following features were introduced: Dead Peer Detection with Stateless Failover (Object Tracking with Easy VPN)—Backup Server List Local Configuration and Backup Server List Auto Configuration, Management Enhancements, Load Balancing, VLAN Support, Multiple Subnet Support, Traffic-Triggered Activation, Perfect Forward Secrecy (PFS) Via Policy Push, 802.1x Authentication, Certificate (PKI) Support, Easy VPN Remote and Server on the Same Interface, and Easy VPN Remote and Site to Site on the Same Interface. **Note** Cisco 800 series routers are not supported in Cisco IOS Release 12.3(7)XR. **Note** These features are available only in Cisco Release 12.3(7)XR2.

12.3(7)XR2	The features in Cisco IOS Release 12.3(7)XR were introduced on Cisco 800 series routers.
12.3(8)YH	The Dial Backup, Traffic-Triggered Activation, and Web-Based Activation features were introduced on the Cisco 1812 router.
12.3(11)T	Except for the Dial Backup and Traffic-Triggered Activation features, all features introduced in Cisco IOS Releases 12.3(7)XR and 12.3(7)XR2 were integrated into Cisco IOS Release 12.3(11)T.
12.3(14)T	Dial Backup and Traffic-Triggered Activation features were integrated into Cisco IOS Release 12.3(14)T. In addition, the Web-Based Activation feature was introduced in this release.
12.3(8)YI	The Dial Backup, Traffic-Triggered Activation, and Web-Based Activation features were introduced on the Cisco 1800 series fixed configuration routers.
12.3(8)YI1	The Dial Backup, Traffic-Triggered Activation, and Web-Based Activation features were introduced on the Cisco 850 series and Cisco 870 series routers.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for Cisco Easy VPN Remote, page 948
- Restrictions for Cisco Easy VPN Remote, page 949
- Information About Cisco Easy VPN Remote, page 950
- How to Configure Cisco Easy VPN Remote, page 972
- Configuration Examples for Cisco Easy VPN Remote, page 996
- Additional References, page 1021
- Command Reference, page 1025
- Appendix A: Supported Mode Configuration Attributes, page 1026

Prerequisites for Cisco Easy VPN Remote

The following requirements are necessary to use the Cisco Easy VPN Remote feature:

- A Cisco 800 series router running Cisco IOS Release 12.2(15)T, 12.3(2)T, 12.3(4)T, 12.3(7)T, or 12.3(7)XR2 configured as a Cisco Easy VPN remote.
- A Cisco 1700 series router running Cisco IOS Release 12.2(15)T, 12.3(2)T, 12.3(4)T, 12.3(7)T, or 12.3(7)XR, configured as a Cisco Easy VPN remote.
- A Cisco uBR905 or Cisco uBR925 cable access router running Cisco IOS Release 12.2(15)T, configured as a Cisco Easy VPN remote.

- A Cisco 1800 series fixed configuration router running Cisco IOS Release 12.3(8)YI.
- Another Cisco router or VPN concentrator that supports the Cisco Easy VPN Server feature and that is configured as a Cisco IOS Easy VPN server. See the "Required Easy VPN Servers" section for a detailed list.

Restrictions for Cisco Easy VPN Remote

Required Easy VPN Servers

The Cisco Easy VPN Remote feature requires that the destination peer be a Cisco IOS Easy VPN server or VPN concentrator that supports the Cisco Easy VPN Server feature. At the time of publication, this includes the following platforms when running the indicated software releases:

- Cisco 806, Cisco 826, Cisco 827, Cisco 828, Cisco 831, Cisco 836, and Cisco 837 routers—Cisco IOS Release 12.2(8)T or later release. Cisco 800 series routers are not supported in Cisco IOS Release 12.3(7)XR, but they are supported in Cisco IOS Release 12.3(7)XR2.
- Cisco 850 series and Cisco 870 series—Cisco IOS Release 12.3(8)YI1.
- Cisco 1700 series—Cisco IOS Release 12.2(8)T or later release.
- Cisco 1800 series fixed configuration router—Cisco IOS Release 12.3(8)YI release.
- Cisco 1812 router—Cisco IOS Release 12.3(8)YH release.
- Cisco 2600 series—Cisco IOS Release 12.2(8)T or later release.
- Cisco 3620—Cisco IOS Release 12.2(8)T or later release.
- Cisco 3640—Cisco IOS Release 12.2(8)T or later release.
- Cisco 3660—Cisco IOS Release 12.2(8)T or later release.
- Cisco 7100 series VPN routers—Cisco IOS Release 12.2(8)T or later release.
- Cisco 7200 series routers—Cisco IOS Release 12.2(8)T or later release.
- Cisco 7500 series routers—Cisco IOS Release 12.2(8)T or later release.
- Cisco PIX 500 series—Software Release 6.2 or later release.
- Cisco VPN 3000 series—Software Release 3.11 or later release.

Only ISAKMP Policy Group 2 Supported on Easy VPN Servers

The Unity Protocol supports only Internet Security Association Key Management Protocol (ISAKMP) policies that use group 2 (1024-bit Diffie-Hellman) Internet Key Exchange (IKE) negotiation, so the Easy VPN server being used with the Cisco Easy VPN Remote feature must be configured for a group 2 ISAKMP policy. The Easy VPN server cannot be configured for ISAKMP group 1 or group 5 when being used with a Cisco Easy VPN client.

Transform Sets Supported

To ensure a secure tunnel connection, the Cisco Easy VPN Remote feature does not support transform sets that provide encryption without authentication (ESP-DES and ESP-3DES) or transform sets that provide authentication without encryption (ESP-NULL ESP-SHA-HMAC and ESP-NULL ESP-MD5-HMAC).

Note The Cisco Unity Client Protocol does not support Authentication Header (AH) authentication, but Encapsulation Security Protocol (ESP) is supported.

Dial Backup for Easy VPN Remotes

Line-status–based backup is not supported in this feature.

Network Address Translation Interoperability Support

Network Addess Translation (NAT) interoperability is not supported in client mode with split tunneling.

Information About Cisco Easy VPN Remote

To configure the Cisco Easy VPN Remote features, you should understand the following concepts:

- Benefits of the Cisco Easy VPN Remote Feature, page 950
- Cisco Easy VPN Remote Overview, page 950
- Modes of Operation, page 951
- Authentication, page 954
- Tunnel Activation Options, page 963
- Dead Peer Detection Stateless Failover Support, page 964
- Cisco Easy VPN Remote Features, page 965

Benefits of the Cisco Easy VPN Remote Feature

- Allows dynamic configuration of end-user policy, requiring less manual configuration by end users and field technicians, thus reducing errors and further service calls.
- Allows the provider to change equipment and network configurations as needed, with little or no reconfiguration of the end-user equipment.
- Provides for centralized security policy management.
- Enables large-scale deployments with rapid user provisioning.
- Eliminates the need for end users to purchase and configure external VPN devices.
- Eliminates the need for end users to install and configure Easy VPN Client software on their PCs.
- Offloads the creation and maintenance of the VPN connections from the PC to the router.
- Reduces interoperability problems between the different PC-based software VPN clients, external hardware-based VPN solutions, and other VPN applications.

Cisco Easy VPN Remote Overview

Cable modems, xDSL routers, and other forms of broadband access provide high-performance connections to the Internet, but many applications also require the security of VPN connections that perform a high level of authentication and that encrypt the data between two particular endpoints. However, establishing a VPN connection between two routers can be complicated and typically requires tedious coordination between network administrators to configure the VPN parameters of the two routers.

The Cisco Easy VPN Remote feature eliminates much of this tedious work by implementing Cisco Unity Client Protocol, which allows most VPN parameters to be defined at a Cisco IOS Easy VPN server. This server can be a dedicated VPN device, such as a Cisco VPN 3000 concentrator or a Cisco PIX Firewall or a Cisco IOS router that supports the Cisco Unity Client Protocol.

After the Cisco Easy VPN server has been configured, a VPN connection can be created with minimal configuration on an Easy VPN remote, such as a Cisco 800 series router or a Cisco 1700 series router. When the Easy VPN remote initiates the VPN tunnel connection, the Cisco Easy VPN server pushes the IPSec policies to the Easy VPN remote and creates the corresponding VPN tunnel connection.

The Cisco Easy VPN Remote feature provides for automatic management of the following details:

- Negotiating tunnel parameters, such as addresses, algorithms, and lifetime.
- Establishing tunnels according to the parameters that were set.
- Automatically creating the NAT or Port Address Translation (PAT) and associated access lists that are needed, if any.
- Authenticating users, that is, ensuring that users are who they say they are by way of usernames, group names, and passwords.
- Managing security keys for encryption and decryption.
- Authenticating, encrypting, and decrypting data through the tunnel.

Modes of Operation

The Cisco Easy VPN Remote feature supports three modes of operation: client, network extension, and network extension plus:

- Client—Specifies that NAT or PAT be done so that the PCs and other hosts at the remote end of the VPN tunnel form a private network that does not use any IP addresses in the IP address space of the destination server.

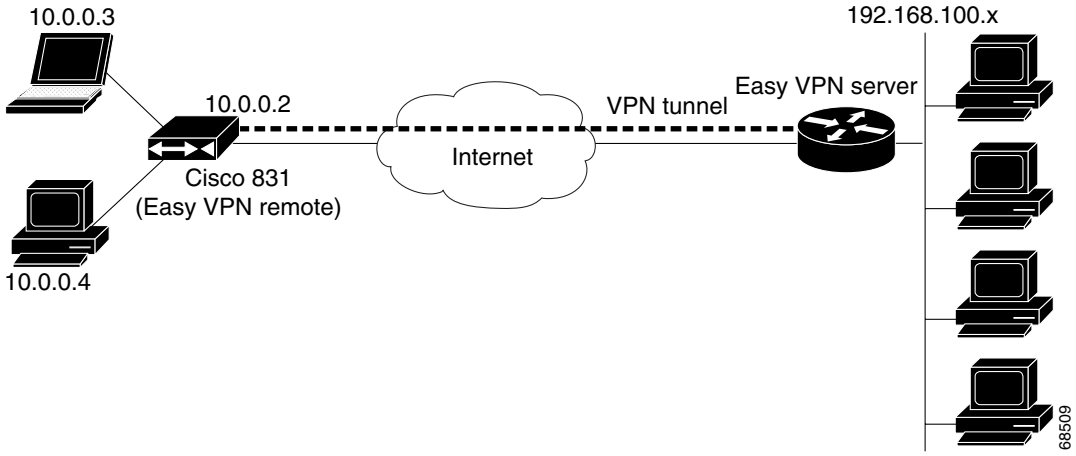
  An enhancement has been made so that the IP address that is received via mode configuration is automatically assigned to an available loopback interface. The IPSec Security Associations (SAs) for this IP address are automatically created by Easy VPN Remote. The IP address is typically used for troubleshooting (using ping, Telnet, and Secure Shell).
- Network extension—Specifies that the PCs and other hosts at the client end of the VPN tunnel should be given IP addresses that are fully routable and reachable by the destination network over the tunneled network so that they form one logical network. PAT is not used, which allows the client PCs and hosts to have direct access to the PCs and hosts at the destination network.
- Network extension plus (mode network-plus)—Identical to network extension mode with the additional capability of being able to request an IP address via mode configuration and automatically assign it to an available loopback interface. The IPSec SAs for this IP address are automatically created by Easy VPN Remote. The IP address is typically used for troubleshooting (using ping, Telnet, and Secure Shell).

All modes of operation also optionally support split tunneling, which allows secure access to corporate resources through the VPN tunnel while also allowing Internet access through a connection to an Internet service provider (ISP) or other service—thereby eliminating the corporate network from the path for web access.

Client Mode and Network Extension Mode Scenarios

Figure 58 illustrates the client mode of operation. In this example, the Cisco 831 router provides access to two PCs, which have IP addresses in the 10.0.0.0 private network space. These PCs connect to the Ethernet interface on the Cisco 831 router, which also has an IP address in the 10.0.0.0 private network space. The Cisco 831 router performs NAT or PAT translation over the VPN tunnel so that the PCs can access the destination network.

Figure 58 Cisco Easy VPN Remote Connection



Note The diagram in Figure 58 could also represent a split tunneling connection, in which the client PCs can access public resources in the global Internet without including the corporate network in the path for the public resources.

Figure 59 also illustrates the client mode of operation, in which a VPN concentrator provides destination endpoints to multiple xDSL clients. In this example, Cisco 800 series routers provide access to multiple small business clients, each of which uses IP addresses in the 10.0.0.0 private network space. The Cisco 800 series routers perform NAT or PAT translation over the VPN tunnel so that the PCs can access the destination network.

Figure 59 Cisco Easy VPN Remote Connection (using a VPN concentrator)

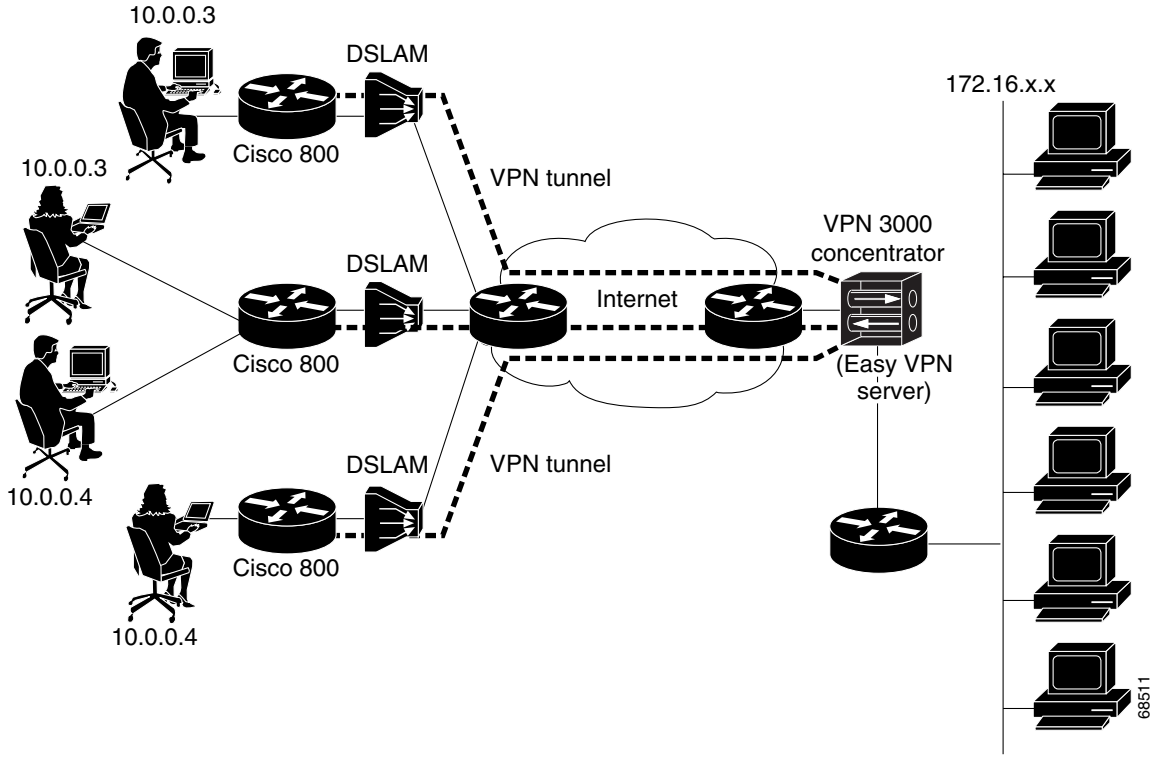
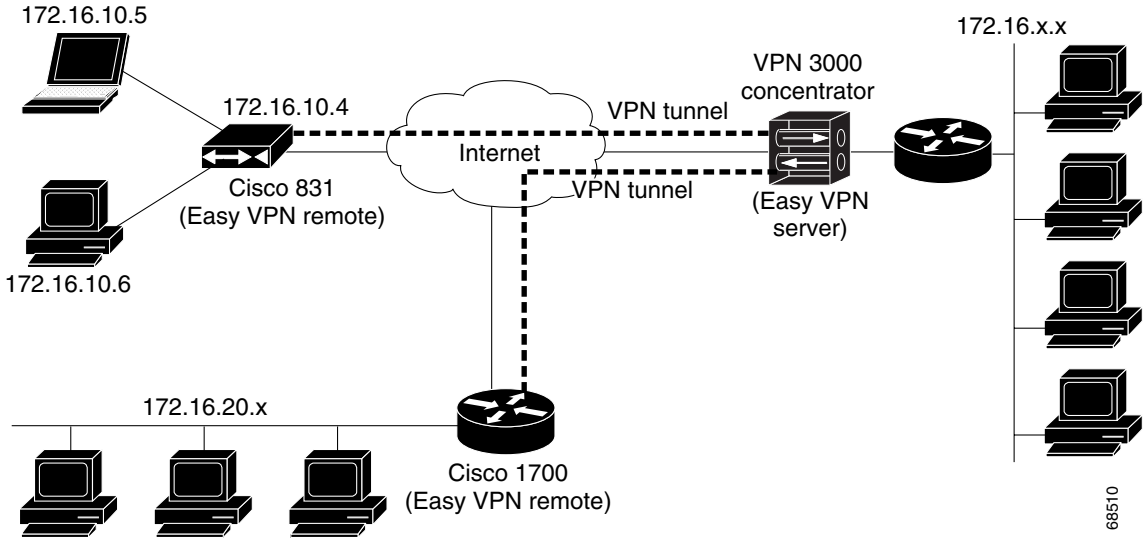


Figure 60 illustrates the network extension mode of operation. In this example, the Cisco 831 router and Cisco 1700 series router both act as Cisco Easy VPN remote devices, connecting to a Cisco VPN 3000 concentrator.

The client hosts are given IP addresses that are fully routable by the destination network over the tunnel. These IP addresses could be either in the same subnet space as the destination network, or in separate subnets, assuming that the destination routers are configured to properly route those IP addresses over the tunnel.

In this example, the PCs and hosts attached to the two routers have IP addresses that are in the same address space as the destination enterprise network. The PCs connect to the Ethernet interface of the Cisco 831 router, which also has an IP address in the enterprise address space. This scenario provides a seamless extension of the remote network.

Figure 60 Cisco Easy VPN Network Extension Connection



Authentication

The Cisco Easy VPN Remote feature supports a two-stage process for authenticating the remote router to the central concentrator. The first step is Group Level Authentication and is part of the control channel creation. In this first stage, two types of authentication credentials can be used: either preshared keys or digital certificates. The following paragraphs provide details about these options.

The second authentication step is called Extended Authentication or Xauth. In this step the remote side (in this case the Easy VPN router) submits a username and password to the central site router. This is the same process as when a user who is using the Cisco VPN software client on a PC enters his or her username and password to activate his or her VPN tunnel. The difference when using the router is that the router itself is being authenticated to the network, not a PC with Cisco VPN Client software. Xauth is an optional step (it can be disabled) but is normally enabled to improve security. It is important to note that after Xauth is successful and the tunnel comes up, all PCs behind the Easy VPN remote router have access to the tunnel.

If Xauth is enabled, the key issue to be decided is how to input the username and password. There are two options. The first option is to store the Xauth username and password in the configuration file of the router. This option is typically used if the router is shared between several PCs and the goal is to keep the VPN tunnel up all the time (see the section "Automatic Activation") or to have the router automatically bring up the tunnel whenever there is data to be sent (see the section "Traffic-Triggered Activation"). An example of this application is a branch office situation, in which the users in the branch office want the VPN tunnel to be available whenever they have data to send and do not want to have to do anything special to activate the VPN tunnel. If the PCs in the branch office need to be individually authenticated on the basis of the ID of each user, the correct configuration is to put the Easy VPN router in Automatic Activation mode to keep the tunnel "up" all the time and to use Cisco IOS Authentication Proxy or 802.1x to authenticate the individual PCs. Because the tunnel is always up, Authentication Proxy or 802.1x can access a central site user database such as AAA/RADIUS to authenticate the individual user requests as they are submitted by PC users. (See the "Related Documents" sections "General information on IPSec and VPN" for a reference to configuring Authentication Proxy and "802.1x authentication" for a reference to configuring 802.1x authentication.)

The second option for entry of the Xauth username and password is not to store it on the router. Instead, a PC user who is connected to the router is presented with a special web page that allows the user to manually enter the username and password (see the section "Manual Activation"). The router sends the username and password to the central site concentrator, and if the username and password are correct, the tunnel comes up. The typical application for this configuration is a teleworker network. The teleworker wants to be able to control when the tunnel is up and has to enter his or her personal user credentials (which could include one-time passwords) to activate the tunnel. Also, the network administrator may only want teleworker tunnels up when someone is using them to conserve resources on the central concentrators. (See the section "Web-Based Activation" for details about this configuration.)

The Xauth username and password can also be manually entered from the command line interface (CLI) of the router. This method is not recommended for most situations because the user must first log in to the router (and needs a user ID on the router to do so). However, it can be useful for network administrators during troubleshooting.

Using Preshared Keys

Using preshared keys, each peer is aware of the key of the other peer. Preshared keys are displayed in running configurations, so they can be seen by anyone (referred to as clear format). When a more secure type of authentication is required, Cisco software also supports another type of preshared key: the encrypted preshared key.

Using an encrypted preshared key for authentication allows you to securely store plain-text passwords in type 6 (encrypted) format in NVRAM. A group preshared key can be preconfigured on both VPN-tunnel peers. The encrypted form of the keyword can be seen in the running configuration, but the actual keyword is not visible. (For more information about encrypted preshared keys, see *Encrypted Preshared Key*.)

Using Digital Certificates

Digital certificates provide for the support of Rivest, Shamir, and Adelman (RSA) signatures on Easy VPN remote devices. The support is provided through a RSA certificate that can be stored on or off the remote device.

Note The recommended timeout for Easy VPN using digital certificates is 40 seconds.

For more information about digital certificates, see the *Easy VPN Remote RSA Signature Support* feature guide, Release 12.3(7)T1.

Using Xauth

Xauth is an additional level of authentication that can be used. Xauth is applicable when either group preshared keys or digital certificates are used. Xauth credentials can be entered using a web interface manager, such as Security Device Manager (SDM), or using the CLI. (See the section "Cisco Easy VPN Remote Web Managers.")

The Save Password feature allows the Xauth username and password to be saved in the Easy VPN Remote configuration so that you are not required to enter the username and password manually. One-Time Passwords (OTPs) are not supported by the Save Password feature and must be entered

manually when Xauth is requested. The Easy VPN server must be configured to "Allow Saved Passwords." (For more information about how to configure the Save Password feature, see the section "Dead Peer Detection Periodic Message Option.")

Xauth is controlled by the Easy VPN server. When the Cisco IOS Easy VPN server requests Xauth authentication, the following messages are displayed on the console of the router:

```
EZVPN: Pending XAuth Request, Please enter the following command:
crypto ipsec client ezvpn xauth
```

When you see this message, you can provide the necessary user ID, password, and other information by entering the **crypto ipsec client ezvpn connect** command and responding to the prompts that follow.

The recommended Xauth timeout is 50 seconds or fewer.

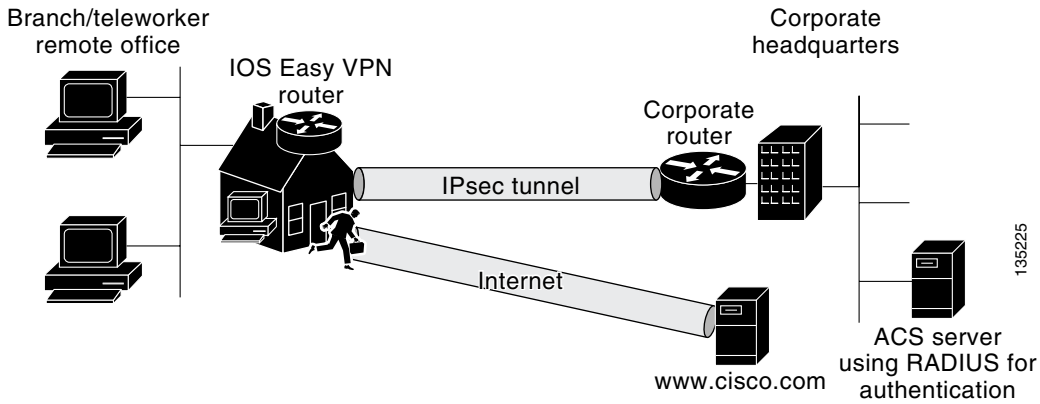
Note The timeout for entering the username and password is determined by the configuration of the Cisco IOS Easy VPN server. For servers running Cisco IOS software, this timeout value is specified by the **crypto isakmp xauth timeout** command.

Web-Based Activation

Web-Based Activation provides a user-friendly method for a remote teleworker to authenticate the VPN tunnel between his or her remote Easy VPN router and the central site router. This feature allows administrators to set up their remote LANs so that the initial HTTP request that is coming from any of the remote PCs is intercepted by the remote Easy VPN router. A login page is returned to the user, whereby the user may enter credentials to authenticate the VPN tunnel. After the VPN tunnel comes up, all users behind this remote site can access the corporate LAN without getting reprompted for the username and password. Alternatively, the user may choose to bypass the VPN tunnel and connect only to the Internet, in which case a password is not required.

A typical application for web-based activation is a home teleworker who brings up the Easy VPN tunnel only when he or she needs to connect to the corporate LAN. If the remote teleworker is not present, other members of the household (such as a spouse or children) can use the "Internet Only" option to browse the Internet without activating the VPN tunnel. Figure 61 shows a typical scenario for web-based activation.

Figure 61 Typical Web-Based Activation Scenario



Note Entering the Xauth credentials brings up the tunnel for all users who are behind this remote site. After the tunnel is up, any additional PCs that are behind the remote site do not get prompted for Xauth credentials. Web-Based Activation is an authentication to bring up the VPN tunnel for all remote PCs and cannot be considered individual user authentication. Individual user authentication for VPN tunnel access is available using the Cisco IOS Authentication Proxy or 802.1x features, which can be configured on the Remote Easy VPN router. (See the "Related Documents" sections "General information on IPSec and VPN" for a reference to configuring Authentication Proxy and "802.1x authentication" for a reference to configuring 802.1x authentication.)

To configure web-based activation, see the section "Configuring Web-Based Activation."

The following sections show the various screen shots that a remote teleworker sees when the Web-Based Activation feature is turned on:

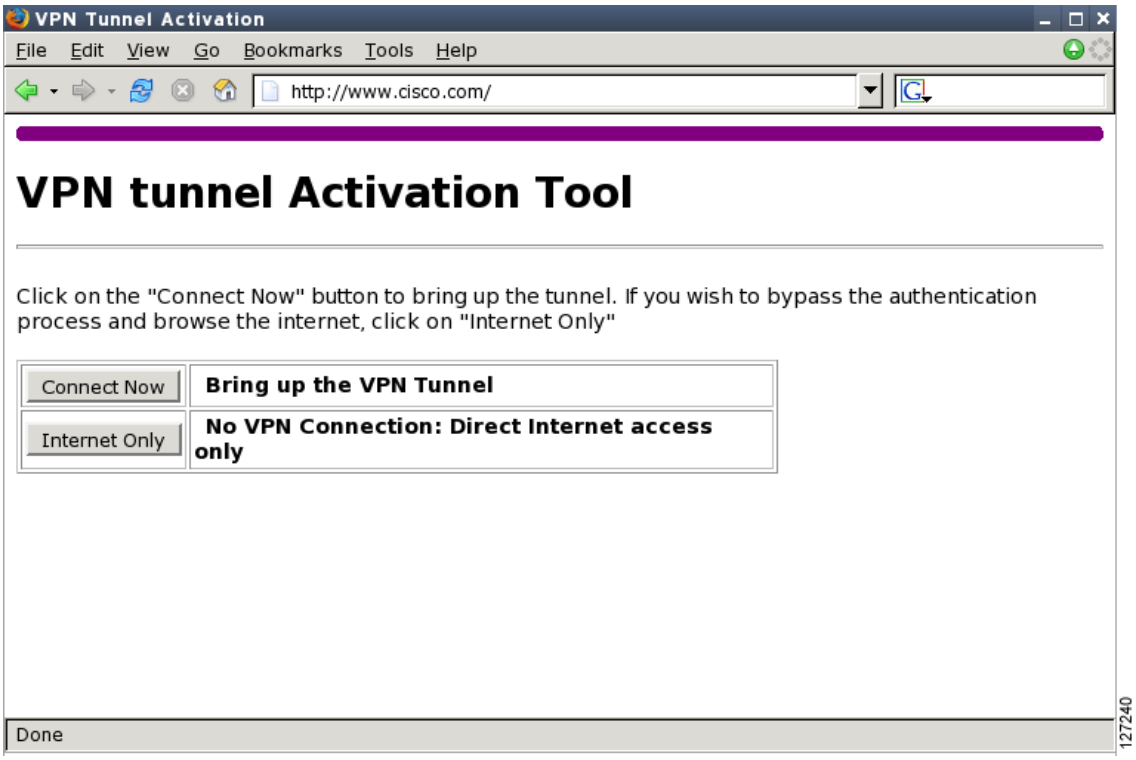
- Web-Based Activation Portal Page, page 957
- VPN Authentication Bypass, page 958
- VPN Tunnel Authentication, page 959
- Successful Authentication, page 960
- Deactivation, page 961

Web-Based Activation Portal Page

Figure 62 is an example of a Web-based activation portal page. The user may choose to connect to the corporate LAN by clicking Connect Now or he or she may choose to connect only to the Internet by clicking Internet Only.

Note If the user chooses to connect only to the Internet, a password is not needed.

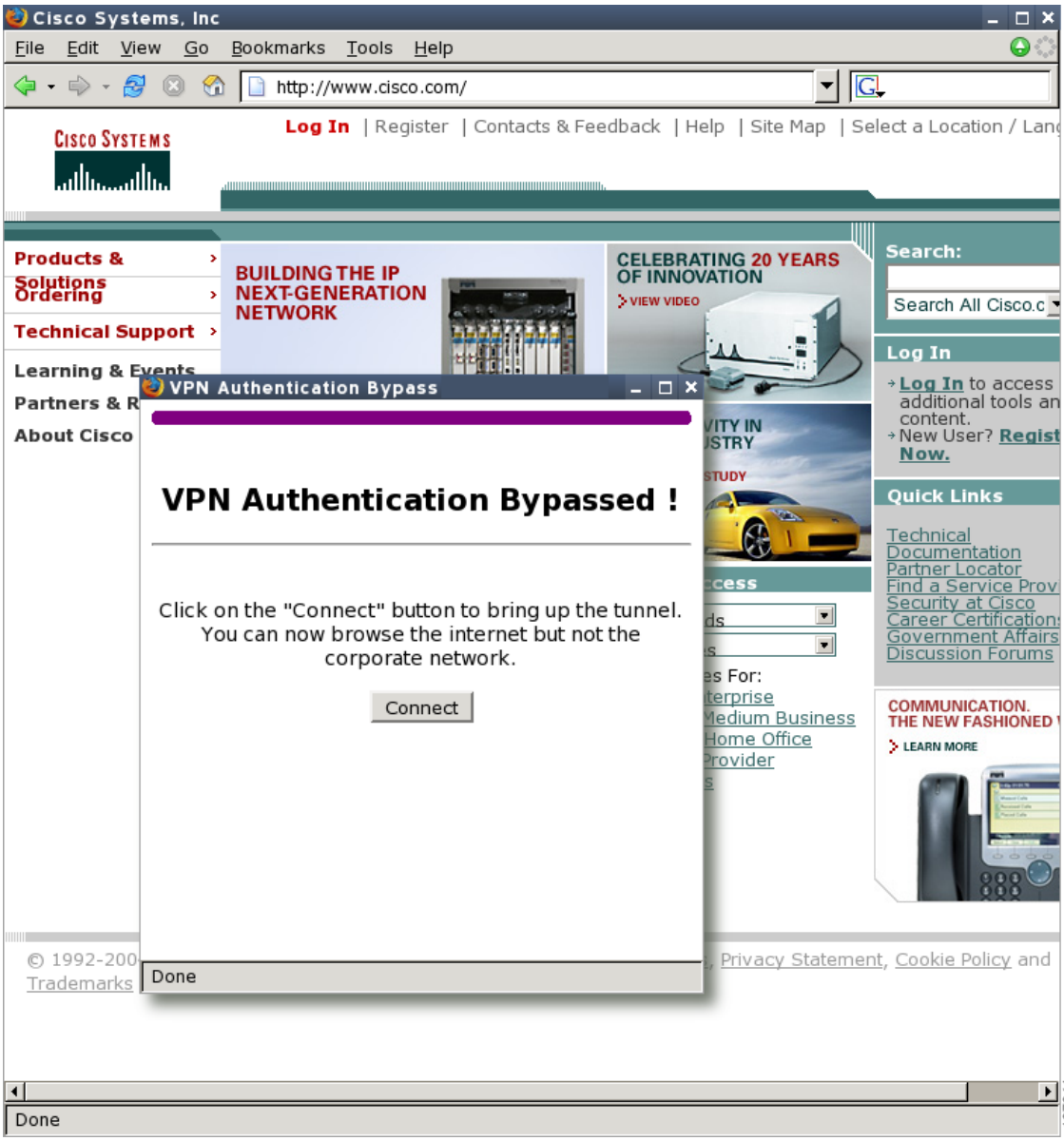
Figure 62 Portal Page



VPN Authentication Bypass

Figure 63 is an example of a web-based activation in which the user chose to connect only to the Internet by clicking the “Internet Only” option. This option will be most useful for household members who need to browse the Internet while the remote teleworker is not available to authenticate the VPN tunnel for corporate use.

Figure 63 VPN Authentication Bypass Page

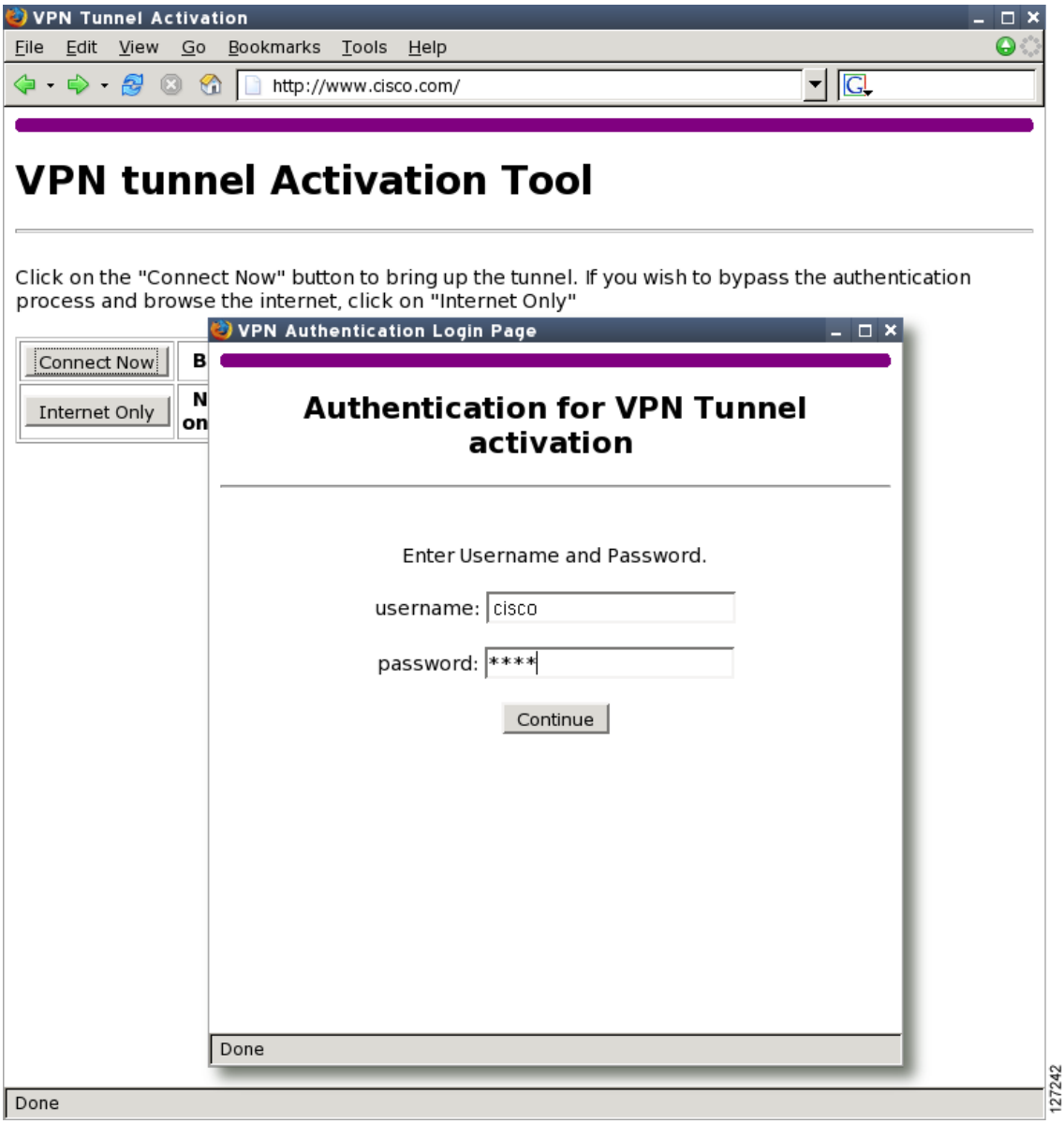


Note If a user mistakenly closes the Web-Based Activation window, the window can be reopened by accessing the remote router (by typing http://routeripaddress/ezvpn/connect). After the Web-Based Activation window opens, the Easy VPN tunnel can be authenticated.

VPN Tunnel Authentication

Figure 64 is an example of a web-based activation in which the user chose to connect to the corporate LAN by entering a username and password. After the user successfully authenticates, the Easy VPN tunnel is brought up for this remote site. If there are multiple PCs behind this remote site, none of the additional users who are connecting to the corporate LAN will be requested for the Xauth credentials because the tunnel is already up.

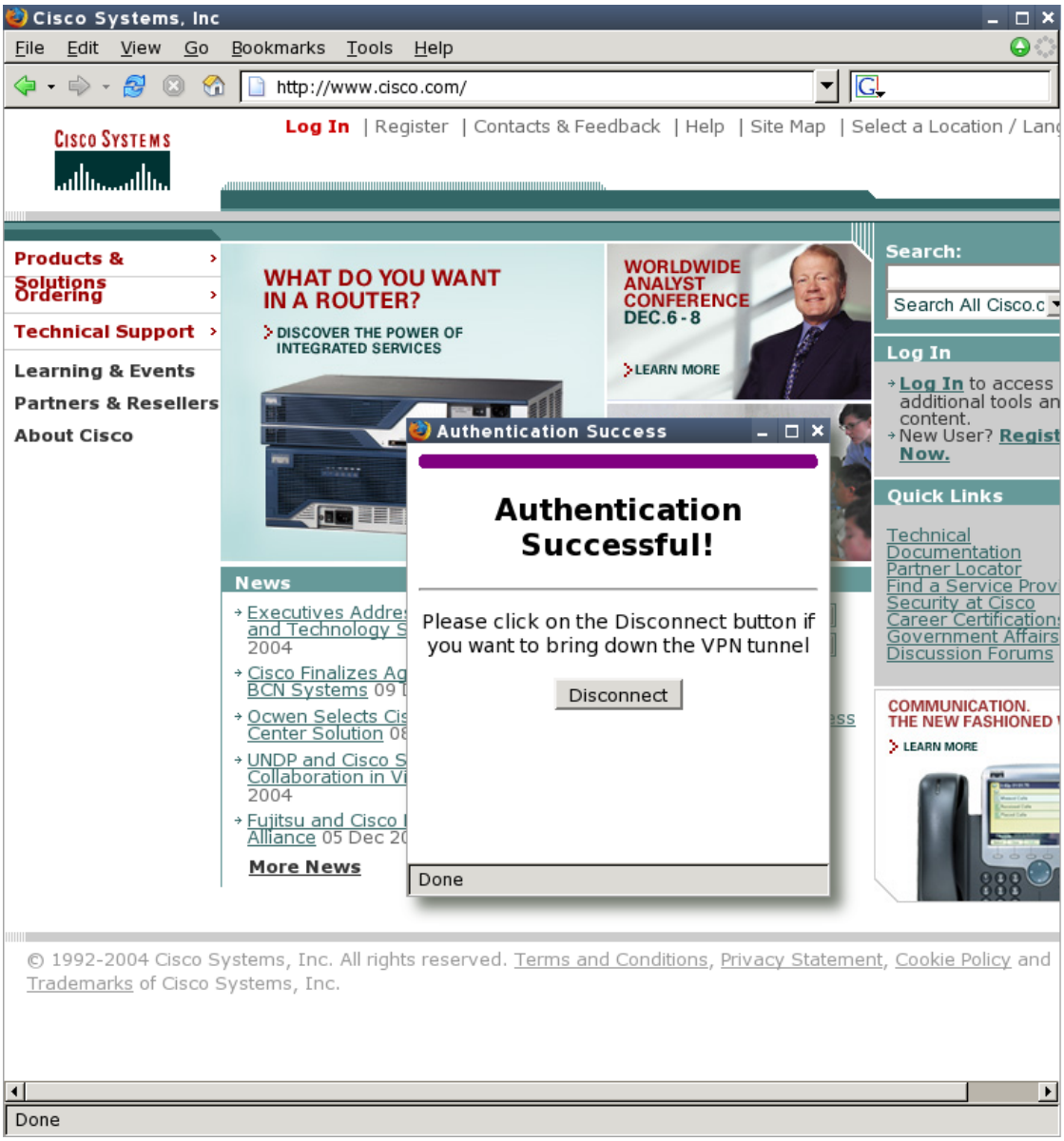
Figure 64 VPN Tunnel Authentication



Successful Authentication

Figure 65 is an example of a successful activation. If the user chooses to deactivate the VPN tunnel, he or she should click the Disconnect button. After the IKE security association (SA) times out (the default value is 24 hours), the remote teleworker has to enter the Xauth credentials to bring up the tunnel.

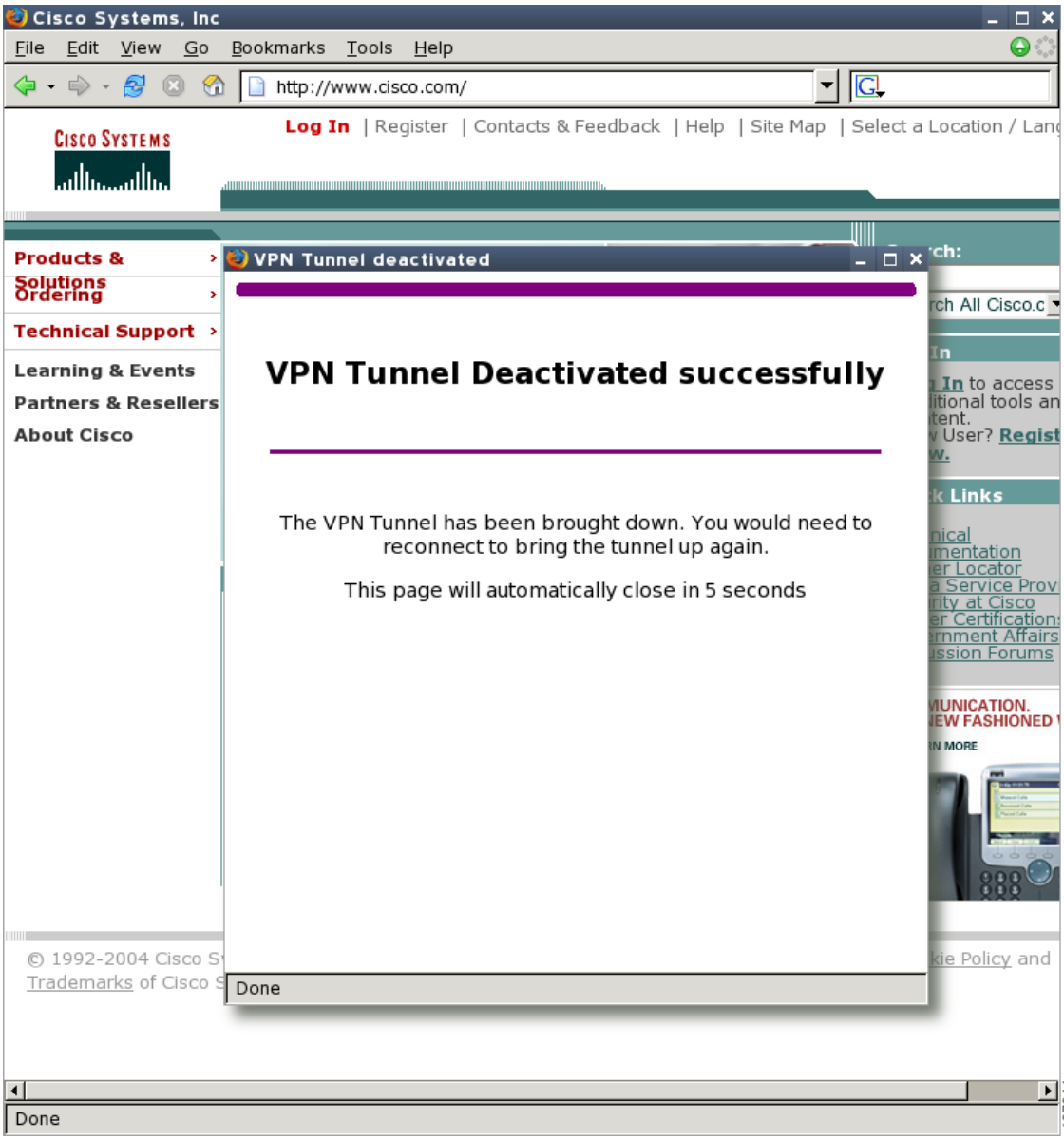
Figure 65 Successful Activation



Deactivation

Figure 66 is an example of a VPN tunnel that has been deactivated successfully. The page automatically closes in 5 seconds.

Figure 66 VPN Tunnel Deactivated Successfully



802.1x Authentication

The 802.1x Authentication feature allows you to combine Easy VPN client mode operation with 802.1x authentication on Cisco IOS routers. For more information about this feature, see "802.1 Authentication" in the section "Additional References."

Tunnel Activation Options

There are three tunnel activation options:

- Automatic activation
- Manual activation
- Traffic-triggered activation (not available in Cisco IOS Release 12.3(11)T)

Tunnel connect and disconnect options are available with SDM.

Automatic Activation

The Cisco Easy VPN tunnel is automatically connected when the Cisco Easy VPN Remote feature is configured on an interface. If the tunnel times out or fails, the tunnel automatically reconnects and retries indefinitely.

To specify automatic tunnel control on a Cisco Easy VPN remote device, you need to configure the **crypto ipsec client ezvpn** command and then the **connect auto** subcommand. However, you do not need to use these two commands when you are creating a new Easy VPN remote configuration because the default is "automatic."

To disconnect or reset a particular tunnel, you should use the **clear crypto ipsec client ezvpn** command, or you can use SDM.

Manual Activation

The Cisco Easy VPN Remote software implements manual control of the Cisco Easy VPN tunnels so that you can establish and terminate the tunnel on demand.

To specify manual tunnel control on a Cisco Easy VPN remote device, you need to input the **crypto ipsec client ezvpn** command and then the **connect manual** command.

The manual setting means that the Cisco Easy VPN remote will wait for a command before attempting to establish the Cisco Easy VPN Remote connection. When the tunnel times out or fails, subsequent connections will also have to wait for the command.

If the configuration is manual, the tunnel is connected only after you issue the command **crypto ipsec client ezvpn connect**.

To disconnect or reset a particular tunnel, you should use the **clear crypto ipsec client ezvpn** command, or you can use SDM.

See the "Configuring Manual Tunnel Control" section for specific information on how to configure manual control of a tunnel.

Traffic-Triggered Activation

Note This feature is not available in Cisco IOS Release 12.3(11)T.

The Traffic-Triggered Activation feature is recommended for transactional-based VPN applications. It is also recommended for use with the Easy VPN dial backup feature for the backup Easy VPN configuration so that backup is activated only when there is traffic to send across the tunnel.

To use Access Control List (ACL) tunnel control, you must first describe the traffic that is considered "interesting." For more information about ACLs, see the chapter "Access Control Lists: Overview and Guidelines" in the "Traffic Filtering and Firewalls" section of the *Cisco IOS Security Configuration Guide*, Release 12.3. To actually configure an ACL-triggered tunnel, use the **crypto ipsec client ezvpn** command with the **connect acl** subcommand.

Dead Peer Detection Stateless Failover Support

Two options are available for configuring Dead Peer Detection Stateless Failover Support:

- Backup Server List Local Configuration
- Backup Server List Auto Configuration

Backup Server List Local Configuration

Backup Server List Local Configuration allows users to enter multiple peer statements. With this feature configured, if the client is connecting to a peer and the negotiation fails, Easy VPN fails over to the next peer. This failover continues through the list of peers. When the last peer is reached, Easy VPN rolls over to the first peer. The IKE and IPSec SAs to the previous peer are deleted. Multiple peer statements work for both IP addresses as well as for hostnames. Setting or unsetting the peer statements will not affect the order of the peer statements.

To use this feature, use the **peer** subcommand of the **crypto ipsec client ezvpn** command.

Backup Server List Auto Configuration

Easy VPN remote that is based on Cisco IOS software can have up to 10 backup servers configured for redundancy. The Backup Server feature allows the Easy VPN server to "push" the backup server list to the Easy VPN remote.

The backup list allows the administrator to control the backup servers to which a specific Easy VPN remote will connect in case of failure, retransmissions, or dead peer detection (DPD) messages.

Note Before the backup server feature can work, the backup server list has to be configured on the server.

How Backup Server Works

If remote A goes to server A and the connection fails, remote A goes to server B. If server B has a backup list configured, that list will override the backup server list of server A. If the connection to server B fails, remote A will continue through the backup servers that have been configured.

Note If you are in auto mode and you have a failure, you will transition automatically from server A to server B. However, if you are in manual mode, you have to configure the transition manually. To configure the transition manually, use the **crypto ipsec client ezvpn** command with the **connect** keyword.

No new configuration is required at the Easy VPN remote to enable this feature. If you want to display the current server, you can use the **show crypto ipsec client ezvpn** command. If you want to find out which peers were pushed by the Easy VPN server, you can use the same command.

To troubleshoot this feature, use the **debug crypto ipsec client ezvpn** command. If more information is needed for troubleshooting purposes, use the **debug crypto isakmp** command. The **show crypto ipsec client ezvpn** command may also be used for troubleshooting.

Cisco Easy VPN Remote Features

The Cisco Easy VPN Remote feature is a collection of features that improves the capabilities of the Cisco Easy VPN Remote feature introduced in Cisco IOS Release 12.2(4)YA. The Cisco Easy VPN Remote feature includes the following:

- Default Inside Interface, page 966—Supports the autoconfiguration of the default Easy VPN inside interface for Cisco 800 series routers.
- Multiple Inside Interfaces, page 966—Configures up to eight inside interfaces on the Cisco Easy VPN remote.
- Multiple Outside Interfaces, page 967—Configures up to four outside tunnels for outside interfaces.
- VLAN Support, page 967—Allows VLANs to be configured as valid Easy VPN inside interfaces.
- Multiple Subnet Support, page 967—Allows multiple subnets from the Easy VPN inside interface to be included in the Easy VPN tunnel.
- NAT Interoperability Support, page 967—Automatically restores the NAT configuration when the IPSec VPN tunnel is disconnected.
- Local Address Support, page 968—The Cisco Easy VPN Remote feature is enhanced to support an additional **local-address** attribute that specifies which interface is used to determine the IP address used to source the Easy VPN tunnel traffic.
- Peer Hostname, page 968—When a peer is defined as a hostname, the hostname is stored and the Domain Name System (DNS) lookup is done at the time of tunnel connection.
- Proxy DNS Server Support, page 968—Configures the router in a Cisco Easy VPN remote configuration to act as a proxy DNS server for LAN-connected users.
- Cisco IOS Firewall Support, page 969—Supports Cisco IOS Firewall configurations on all platforms.
- Easy VPN Remote and Server on the Same Interface, page 969—The Easy VPN remote and Easy VPN server are supported on the same interface, which makes it possible to establish a tunnel to another Easy VPN server and terminate the Easy VPN software client on the same interface simultaneously.
- Easy VPN Remote and Site to Site on the Same Interface, page 969—The Easy VPN Remote and site to site (crypto map) are supported on the same interface, which makes it possible to establish a tunnel to another Easy VPN server and have another site to site on the same interface simultaneously.
- Cisco Easy VPN Remote Web Managers, page 969—Users can manage the Cisco Easy VPN Remote feature on the Cisco uBR905 and Cisco uBR925 cable access routers using a built-in web interface.
- Dead Peer Detection Periodic Message Option, page 970—Allows you to configure your router to query the liveliness of its IKE peer at regular intervals.
- Load Balancing, page 970—If a remote device is loaded and unable to accept more traffic, the VPN 3000 will send a notify message that contains an IP address that represents the new IKE server to which the remote should connect.
- Management Enhancements, page 970—Allows for remote management of the VPN remote.

- PFS Support, page 970—The PFS configuration mode attribute is sent by the server if requested by the VPN remote device.
- Dial Backup, page 971—Allows you to configure a dial backup tunnel connection on your remote device.

Default Inside Interface

Easy VPN Remote supports the autoconfiguration of the default Easy VPN inside interface for Cisco 800 series routers. The interface Ethernet 0 is the default inside interface.

If you want to disable the default inside interface and configure another inside interface on the Cisco 800 series router, you must configure the other inside interface first and then disable the default inside interface. You can use the following command to disable the default inside interface:

```
no crypto ipsec client ezvpn <name> inside
```

If you did not configure the other inside interface first before disabling the default inside interface, you will receive a message such as the following (see lines three and four):

```
Router (config)# interface ethernet0
Router (config-if)# no crypto ipsec client ezvpn hw-client inside
Cannot remove the single inside interface unless
one other inside interface is configured
```

Multiple Inside Interfaces

Inside interface support is enhanced in the Cisco Easy VPN Remote feature to support multiple inside interfaces for all platforms. Inside interfaces can be configured manually with the enhanced command and subcommand:

```
interface interface-name
    crypto ipsec client ezvpn name [outside | inside]
```

See the "Configuring Multiple Inside Interfaces" section for information on how to configure more than one inside interface.

Multiple inside interfaces offer the following capabilities:

- Up to eight inside interfaces are supported on the Cisco 800 and Cisco 1700 series routers.
- At least one inside interface must be configured for each outside interface; otherwise, the Cisco Easy VPN Remote feature does not establish a connection.
- Adding a new inside interface or removing an existing inside interface automatically resets the Cisco Easy VPN Remote connection (the currently established tunnel). You must reconnect a manually configured tunnel, and if Xauth is required by the Cisco Easy VPN server, the user is reprompted. If you have set the Cisco Easy VPN Remote configuration to connect automatically and no Xauth is required, no user input is required.
- Inside interfaces that are configured or the default setting can be shown by using the **show crypto ipsec client ezvpn** command.

Multiple Outside Interfaces

The Easy VPN Remote feature supports one Easy VPN tunnel per outside interface. You can configure up to four Easy VPN tunnels per Cisco router. Each Easy VPN tunnel can have multiple inside interfaces configured, but they cannot overlap with another Easy VPN tunnel unless dial backup is configured. For more information about dial backup, see the section "Dial Backup." To configure multiple outside interfaces, use the **crypto ipsec client ezvpn** command and **outside** keyword.

To disconnect or clear a specific tunnel, the **clear crypto ipsec client ezvpn** command specifies the IPSec VPN tunnel name. If there is no tunnel name specified, all existing tunnels are cleared.

See the "Configuring Multiple Outside Interfaces" section for more information on configuring more than one outside interface.

VLAN Support

Inside interface support on VLANs makes it possible to have valid Easy VPN inside interface support on a VLAN, which was not possible before Cisco IOS Release 12.3(7)XR. With this feature, SAs can be established at connection using the VLAN subnet address or mask as a source proxy.

For the inside interface support on VLANs to work, you must define each VLAN as an Easy VPN inside interface. In addition, IPSec SAs should be established for each inside interface in the same manner as for other inside interfaces. For more information about inside and outside interfaces, see the sections "Multiple Inside Interfaces" and "Multiple Outside Interfaces."

Inside interface support on VLANs is supported only on Cisco routers that support VLANs.

Multiple Subnet Support

For situations in which you have multiple subnets connected to an Easy VPN inside interface, you can optionally include these subnets in the Easy VPN tunnel. First, you must specify the subnets that should be included by defining them in an ACL. To configure an ACL, see "Access control lists, configuring" in the "Additional References" section. Next, you have to use the **acl** subcommand of the **crypto ipsec client ezvpn** (global) command to link your ACL to the Easy VPN configuration. Easy VPN Remote will automatically create the IPSec SAs for each subnet that is defined in the ACL as well as for the subnets that are defined on the Easy VPN inside interface.

Note Multiple subnet support is not supported in client mode.

NAT Interoperability Support

Cisco Easy VPN Remote supports interoperability with NAT. You can have a NAT configuration and a Cisco Easy VPN Remote configuration that coexist. When an IPSec VPN tunnel is down, the NAT configuration works.

In the Cisco Easy VPN Remote feature, the router automatically restores the previous NAT configuration when the IPSec VPN tunnel is torn down. The user-defined access lists are not disturbed. Users can continue to access nontunnel areas of the Internet when the tunnel times out or disconnects.

Note NAT interoperability is not supported in client mode with split tunneling.

Local Address Support

The Cisco Easy VPN Remote feature is enhanced to support an additional **local-address** attribute. This attribute specifies which interface is used to determine the IP address that is used to source the Easy VPN Remote tunnel traffic. After specifying the interface with the **local-address** subcommand, you can manually assign a static IP address to the interface or use the **cable-modem dhcp-proxy interface** command to automatically configure the specified interface with a public IP address. See the "Configuring Proxy DNS Server Support" section for configuration information.

Local Address Support is available for all platforms, but it is more applicable to the Cisco uBR905 and Cisco uBR925 cable access routers in conjunction with the **cable-modem dhcp-proxy interface** command. Typically, the loopback interface is the interface used to source tunnel traffic for the Cisco uBR905 and Cisco uBR925 cable access routers.

In a typical DOCSIS network, the Cisco uBR905 and Cisco uBR925 cable access routers are normally configured with a private IP address on the cable modem interface. In the initial Cisco Easy VPN Remote feature, a public IP address was required on the cable modem interface to support the Easy VPN remote.

In the Cisco Easy VPN Remote feature, cable providers can use the Cable DHCP Proxy feature to obtain a public IP address and assign it to the cable modem interface, which is usually the loopback interface.

For more information on the **cable-modem dhcp-proxy interface** command, see the "Cable CPE Commands" chapter at http://www.cisco.com/univercd/cc/td/doc/product/cable/bbccmref/bbcmcpe.htm in the *Cisco Broadband Cable Command Reference Guide* at http://www.cisco.com/univercd/cc/td/doc/product/cable/bbccmref/index.htm.

Note The **cable-modem dhcp-proxy interface** command is supported only for the Cisco uBR905 and Cisco uBR925 cable access routers.

Peer Hostname

The peer in a Cisco Easy VPN Remote configuration can be defined as an IP address or a hostname. Typically, when a peer is defined as a hostname, a DNS lookup is done immediately to get an IP address. In the Cisco Easy VPN Remote feature, the peer hostname operation is enhanced to support DNS entry changes. The text string of the hostname is stored so that the DNS lookup is done at the time of the tunnel connection, not when the peer is defined as a hostname.

See the "Configuring and Assigning the Easy VPN Remote Configuration" section for information on enabling the peer hostname functionality.

Proxy DNS Server Support

When the Easy VPN tunnel is down, the DNS addresses of the ISP or cable provider should be used to resolve DNS requests. When the WAN connection is up, the DNS addresses of the enterprise should be used.

As a way of implementing use of the DNS addresses of the cable provider when the WAN connection is down, the router in a Cisco Easy VPN Remote configuration can be configured to act as a proxy DNS server. The router, acting as a proxy DNS server for LAN-connected users, receives DNS queries from local users on behalf of the real DNS server. The DHCP server then can send out the LAN address of the router as the IP address of the DNS server. After the WAN connection comes up, the router forwards the DNS queries to the real DNS server and caches the DNS query records.

See the "Configuring Proxy DNS Server Support" section for information on enabling the proxy DNS server functionality.

Cisco IOS Firewall Support

The Cisco Easy VPN Remote feature works in conjunction with Cisco IOS Firewall configurations on all platforms.

Easy VPN Remote and Server on the Same Interface

This feature allows the Easy VPN remote and Easy VPN server to be supported on the same interface, making it possible to both establish a tunnel to another Easy VPN server and terminate the Easy VPN software client on the same interface simultaneously. A typical application would be a geographically remote location for which Easy VPN Remote is being used to connect to a corporate Easy VPN server and also to terminate local software client users.

For more information about the Easy VPN Remote and Server on the Same Interface feature, see "Easy VPN Remote and Server on the Same Interface" in the section "Additional References."

Easy VPN Remote and Site to Site on the Same Interface

This feature allows the Easy VPN remote and site to site (crypto map) to be supported on the same interface, making it possible to both establish a tunnel to another Easy VPN server and have another site to site on the same interface simultaneously. A typical application would be a third-party VPN service provider that is managing a remote router via the site-to-site tunnel and using Easy VPN Remote to connect the remote site to a corporate Easy VPN server.

For more information about the Easy VPN Remote and Site to Site on the Same Interface feature, see "Easy VPN Remote and Site to Site on the Same Interface" in the section "Additional References."

Cisco Easy VPN Remote Web Managers

Web interface managers may be used to manage the Cisco Easy VPN Remote feature. One such web interface manager is SDM, which is supported on the Cisco 830 series, Cisco 1700 series, Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers. SDM enables you to connect or disconnect the tunnel and provides a web interface for Xauth. For more information about SDM, see *Cisco Security Device Manager*.

A second web interface manager is the Cisco Router Web Setup (CRWS) tool, which is supported on the Cisco 806 router. The CRWS provides a similar web interface as SDM.

A third web interface manager, Cisco Easy VPN Remote Web Manager, is used to manage the Cisco Easy VPN Remote feature for Cisco uBR905 and Cisco uBR925 cable access routers. You do not need access to the CLI to manage the Cisco Easy VPN remote connection.

The web interface managers allow you to do the following:

- See the current status of the Cisco Easy VPN remote tunnel.
- Connect a tunnel that is configured for manual control.
- Disconnect a tunnel that is configured for manual control or reset a tunnel configured for automatic connection.
- Be prompted for Xauth information, if needed.

See the "Troubleshooting the VPN Connection" section for more information about Cisco Easy VPN Remote Web Manager.

Dead Peer Detection Periodic Message Option

The dead peer detection periodic message option allows you to configure your router to query the liveliness of its IKE peer at regular intervals. The benefit of this approach over the default approach (on-demand dead peer detection) is earlier detection of dead peers. For more information about the dead peer detection periodic message option, see "Dead peer detection" in the section "Additional References."

Load Balancing

When the Cisco VPN 3000 concentrator is configured for load balancing, the VPN 3000 will accept an incoming IKE request from the VPN remote on its virtual IP address. If the device is loaded and unable to accept more traffic, the VPN 3000 will send a notify message that contains an IP address that represents the new IKE server to which the remote should connect. The old connection will be torn down and a new connection established to the redirected VPN gateway.

There is no configuration required for load balancing to occur. If the VPN gateway is configured for load balancing, and it notifies the VPN remote that it is performing load balancing, the VPN remote has access to the load balancing feature.

To verify whether load balancing is occurring, use the **debug crypto isakmp**, **debug crypto ipsec client ezvpn**, and **show crypto ipsec** commands. To troubleshoot the load balancing process, use the **show crypto ipsec** command.

Management Enhancements

Management enhancements for Easy VPN remotes allow for the remote management of the VPN remote. The feature provides for the IPv4 address to be pushed by configuration mode to the VPN remote. The IPv4 address is assigned to the first available loopback interface on the VPN remote, and any existing statically defined loopbacks are not overridden. On disconnect, the address and loopback interface are removed from the list of active interfaces.

After the VPN remote is connected, the loopback interface should be accessible from the remote end of the tunnel. All PAT activities will be translated through this interface IP address.

If a loopback exists, and an IP address is associated with it and its state is unassigned, the interface is a good candidate for mode configuration address management.

Note After you assign an address to the loopback interface, if you save the configuration to NVRAM and reboot the VPN remote, the configuration address is permanently contained in the configuration. If you saved the configuration to NVRAM and rebooted the VPN remote, you must enter configuration mode and remove the IP address from the loopback interface manually.

You can use the **show ip interface** command with the **brief** keyword to verify that a loopback has been removed. The output of this **show** command also displays the interface.

PFS Support

The PFS configuration mode attribute is sent by the server if requested by the VPN remote device. If any subsequent connection by the remote device shows that PFS is not received by the remote, PFS will not be sent in IPSec proposal suites.

Note The PFS group that will be proposed in the IPSec proposal suites is the same as the group used for IKE.

You can use the **show crypto ipsec client ezvpn** command to display the PFS group and to verify that you are using PFS.

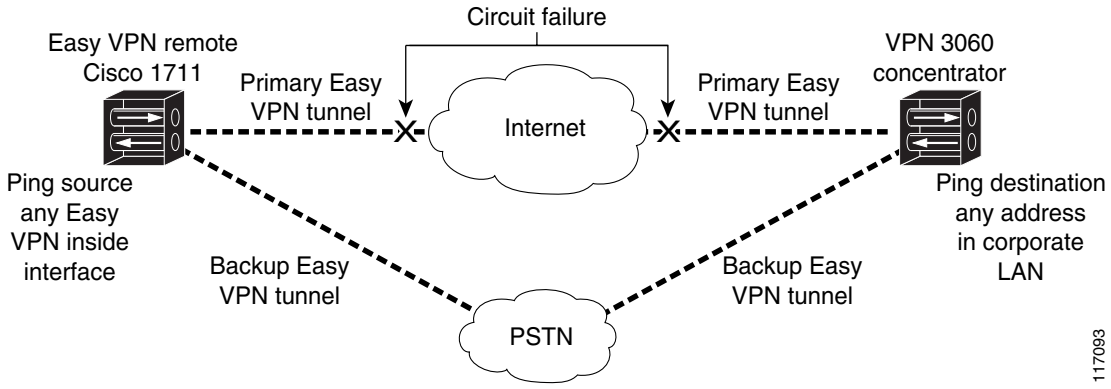
Dial Backup

Note The Dial Backup feature is not available in Cisco IOS Release 12.3(11)T.

Dial backup for Easy VPN remotes allows you to configure a dial backup tunnel connection on your remote device. The backup feature is "brought up" only when real data has to be sent, eliminating the need for expensive dialup or ISDN links that must be created and maintained even when there is no traffic.

Figure 67 illustrates a typical Easy VPN remote-with-dial-backup scenario. In this scenario, a Cisco 1751 remote device is attempting to connect to another Cisco 1751 (acting as a server). There is a failure in the primary Easy VPN tunnel, and the connection is rerouted through the Easy VPN backup tunnel to the Cisco 1751 server.

Figure 67 Dial Backup for Easy VPN Scenario



Dial Backup Using a Dial-on-Demand Solution

IP static route tracking enable Cisco IOS software to identify when a Point-to-Point Protocol over Ethernet (PPPoE) or IPSec VPN tunnel "goes down" and initiates a Dial-on-Demand (DDR) connection to a preconfigured destination from any alternative WAN or LAN port (for example, a T1, ISDN, analog, or auxiliary port). The failure may be caused by several catastropic events (for example, by Internet circuit failures or peer device failure). The remote route has only a static route to the corporate network. The IP static-route-tracking feature allows an object to be tracked (using an IP address or hostname) using Internet Control Message Protocol (ICMP), TCP, or other protocols, and it installs or removes the static route on the basis of the state of the tracked object. If the tracking feature determines that Internet connectivity is lost, the default route for the primary interface is removed, and the floating static route for the backup interface is enabled.

Dial Backup Using Object Tracking

IP static route tracking must be configured for dial backup on an Easy VPN remote device to work. The object tracking configuration is independent of the Easy VPN remote dial backup configuration. (For more information about object tracking, see the feature guide *Reliable Static Routing Backup Using Object Tracking*.)

Easy VPN Remote Dial Backup Support Configuration

You can configure dial backup for your Easy VPN remote using two Easy VPN remote options that allow a connection to the backup Easy VPN configuration and a connection to the tracking system.

- To specify the Easy VPN configuration that will be activated when backup is triggered, use the **backup** subcommand of the **crypto ipsec lient ezvpn** (global) command.
- The Easy VPN remote device registers to the tracking system to get the notifications for change in the state of the object. Use the **track** subcommand to inform the tracking process that the Easy VPN remote device is interested in tracking an object, which is identified by the object number. The tracking process, in turn, informs the Easy VPN remote device when the state of this object changes. This notification prompts the Easy VPN remote device when the state of this object changes. This notification prompts the Easy VPN remote device to bring up the backup connection when the tracked object state is DOWN. When the tracked object is UP again, the backup connection is torn down and the Easy VPN remote device will switch back to using the primary connection.

Note Only one backup configuration is supported for each primary Easy VPN configuration. Each inside interface must specify the primary and backup Easy VPN configuration.

Dynamically Addressed Environments

To allow dial backup to be deployed in dynamically addressed environments, use the IP SLA Pre-Routed ICMP Echo Probe feature. (For more information about this feature, see *Cisco 1700 Series- Cisco IOS Release 12.3(7)XR* release notes. To use the IP SLA Pre-Routed ICMP Echo Probe feature, use the **type echo** command with the **source-interface** keyword.

Dial Backup Examples

For examples of dial backup configurations, see the section "Dial Backup: Examples."

How to Configure Cisco Easy VPN Remote

This section includes the following required and optional tasks.

Remote Tasks

- Configuring and Assigning the Easy VPN Remote Configuration, page 973 (required)
- Verifying the Cisco Easy VPN Configuration, page 975 (optional)
- Configuring Save Password, page 976 (optional)
- Configuring Manual Tunnel Control, page 977 (optional)
- Configuring Automatic Tunnel Control, page 979 (optional)
- Configuring Multiple Inside Interfaces, page 980 (optional)

- Configuring Multiple Outside Interfaces, page 981 (optional)
- Configuring Multiple Subnet Support, page 982 (optional)
- Configuring Proxy DNS Server Support, page 984 (optional)
- Configuring Dial Backup, page 984 (optional)
- Configuring the DHCP Server Pool, page 985 (required)
- Resetting a VPN Connection, page 985 (optional)
- Monitoring and Maintaining VPN and IKE Events, page 986 (optional)

Easy VPN Server Tasks

- Configuring a Cisco IOS Easy VPN Server, page 987 (required)
- Configuring an Easy VPN Server on a VPN 3000 Series Concentrator, page 987 (optional)
- Configuring an Easy VPN Server on a Cisco PIX Firewall, page 989 (optional)

Web Interface Tasks

- Configuring Web-Based Activation, page 990 (optional)
- Monitoring and Maintaining Web-Based Activation, page 990 (optional)
- Using SDM As a Web Manager, page 994 (optional)

Troubleshooting the VPN Connection

- Troubleshooting a VPN Connection Using the Cisco Easy VPN Remote Feature, page 994
- Troubleshooting the Client Mode of Operation, page 994
- Troubleshooting Remote Management, page 995
- Troubleshooting Dead Peer Detection, page 995

Remote Tasks

Configuring and Assigning the Easy VPN Remote Configuration

The router acting as the Easy VPN remote must create a Cisco Easy VPN Remote configuration and assign it to the outgoing interface. To configure and assign the remote configuration, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec client ezvpn** *name*
4. **group** *group-name* **key** *group-key*
5. **peer** [*ip-address* | *hostname*]
6. **mode** {**client** | **network-extension**}
7. **exit**
8. **interface** *interface*

9. **crypto ipsec client ezvpn** *name* [**outside**]
10. **exit**
11. **exit**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec client ezvpn** *name* **Example:** `Router (config)# crypto ipsec client ezvpn easy client remote`	Creates a remote configuration and enters Cisco Easy VPN Remote configuration mode.
Step 4	**group** *group-name* **key** *group-key* **Example:** `Router (config-crypto-ezvpn)# group easy-vpn-remote-groupname key easy-vpn-remote-password`	Specifies the IPSec group and IPSec key value to be associated with this configuration. **Note** The value of the *group-name* argument must match the group defined on the Easy VPN server. On Cisco IOS routers, use the **crypto isakmp client configuration group** and **crypto map dynmap isakmp authorization list** commands. **Note** The value of the *group-key* argument must match the key defined on the Easy VPN server. On Cisco IOS routers, use the **crypto isakmp client configuration group** command.
Step 5	**peer** [*ip-address* | *hostname*] **Example:** `Router (config-crypto-ezvpn)# peer 192.185.0.5`	Specifies the IP address or hostname for the destination peer (typically the IP address on the outside interface of the destination route). • Multiple peers may be configured. **Note** You must have a DNS server configured and available to use the *hostname* option.
Step 6	**mode** {**client** | **network-extension**} **Example:** `Router (config-crypto-ezvpn)# mode client`	Specifies the type of VPN connection that should be made. • **client**—Specifies that the router is configured for VPN client operation, using NAT or PAT address translation. Client operation is the default if the type of VPN connection is not specified • **network-extension**—Specifies that the router is to become a remote extension of the enterprise network at the destination of the VPN connection.

	Command	Purpose
Step 7	**exit** **Example:** `Router (config-crypto-ezvpn)# exit`	Exits Cisco Easy VPN Remote configuration mode.
Step 8	**interface** *interface* **Example:** `Router (config)# interface Ethernet1`	Enters interface configuration mode for the interface. • This interface will become the outside interface for the NAT or PAT translation.
Step 9	**crypto ipsec client ezvpn** *name* [**outside**] **Example:** `Router (config-if)# crypto ipsec client ezvpn easy vpn remote1 outside`	Assigns the Cisco Easy VPN Remote configuration to the interface. • This configuration automatically creates the necessary NAT or PAT translation parameters and initiates the VPN connection (if you are in client mode). **Note** The inside interface must be specified on Cisco 1700 and higher platforms.
Step 10	**exit** **Example:** `Router (config-if)# exit`	Exits interface configuration mode.
Step 11	**exit** **Example:** `Router (config)# exit`	Exits global configuration mode.

Verifying the Cisco Easy VPN Configuration

To verify that the Cisco Easy VPN Remote configuration has been correctly configured, that the configuration has been assigned to an interface, and that the IPSec VPN tunnel has been established, perform the following steps:

SUMMARY STEPS

1. **show crypto ipsec client ezvpn**
2. **show ip nat statistics**

DETAILED STEPS

Step 1 Display the current state of the Cisco Easy VPN Remote connection using the **show crypto ipsec client ezvpn** command. The following is typical output for a Cisco 1700 series router using client mode:

```
Router# show crypto ipsec client ezvpn

Tunnel name : hw1
Inside interface list: FastEthernet0/0, Serial0/0,
Outside interface: Serial1/0
Current State: IPSEC_ACTIVE
Last Event: SOCKET_UP
Address: 10.0.0.5
```

```
Mask: 255.255.255.255
Default Domain: cisco.com
Tunnel name : hw2
Inside interface list: Serial0/1,
Outside interface: Serial1/1
Current State: IPSEC_ACTIVE
Last Event: SOCKET_UP
Default Domain: cisco.com
```

Step 2 Display the NAT or PAT configuration that was automatically created for the VPN connection using the **show ip nat statistics** command. The "Dynamic mappings" field of this display gives the details for the NAT or PAT translation that is occurring on the VPN tunnel.

```
Router# show ip nat statistics

Total active translations: 0 (0 static, 0 dynamic; 0 extended)
Outside interfaces:
  cable-modem0
Inside interfaces:
  Ethernet0
Hits: 1489  Misses: 1
Expired translations: 1
Dynamic mappings:
-- Inside Source
access-list 198 pool enterprise refcount 0
 pool enterprise: netmask 255.255.255.0
        start 192.1.1.90 end 192.1.1.90
        type generic, total addresses 1, allocated 0 (0%), misses 0\
```

If you are seeing IPSEC_ACTIVE in your output at this point, everything is operating as expected.

Configuring Save Password

To configure the Save Password feature, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **password encryption aes**
4. **crypto ipsec client ezvpn** *name*
5. **username** *name* **password** {**0** | **6**} {*password*}
6. **exit**
7. **show running-config**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**password encryption aes** **Example:** `Router (config)# password encryption aes`	Enables a type 6 encrypted preshared key.
Step 4	**crypto ipsec client ezvpn** *name* **Example:** `Router (config)# crypto ipsec client ezvpn ezvpn1`	Creates a Cisco Easy VPN remote configuration and enters the Cisco Easy VPN remote configuration mode.
Step 5	**username** *name* **password** {**0** | **6**} {*password*} **Example:** `Router (config-crypto-ezvpn)# username server_1 password 0 blue`	Allows you to save your Xauth password locally on the PC. • The **0** keyword specifies that an unencrypted password will follow. • The **6** keyword specifies that an encrypted password will follow. • The *password* argument is the unencrypted (cleartext) user password.
Step 6	**exit** **Example:** `Router (config-crypto-ezvpn)# exit`	Exits the Cisco Easy VPN remote configuration mode.
Step 7	**show running-config** **Example:** `Router (config)# show running-config`	Displays the contents of the configuration file that is currently running.

Configuring Manual Tunnel Control

To configure control of IPSec VPN tunnels manually so that you can establish and terminate the IPSec VPN tunnels on demand, perform the following steps.

Note CLI is one option for connecting the tunnel. The preferred method is via the web interface (using SDM).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec client ezvpn** *name*
4. **connect** [**auto** | **manual**]
5. **exit**
6. **exit**
7. **crypto ipsec client ezvpn connect** *name*

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec client ezvpn** *name* **Example:** `Router (config)# crypto ipsec client ezvpn easy vpn remote1`	Assigns a Cisco Easy VPN remote configuration to an interface and enters Cisco Easy VPN Remote configuration mode. • The *name* argument specifies the configuration name to be assigned to the interface.
Step 4	**connect** [**auto** | **manual**] **Example:** `Router (config-crypto-ezvpn)# connect manual`	Connects the VPN tunnel. Specify **manual** to configure manual tunnel control. • Automatic is the default; you do not need to use the **manual** keyword if your configuration is automatic.
Step 5	**exit** **Example:** `Router (config-crypto-ezvpn)# exit`	Exits Cisco Easy VPN Remote configuration mode.
Step 6	**exit** **Example:** `Router (config)# exit`	Exits global configuration mode and enters privileged EXEC mode.
Step 7	**crypto ipsec client ezvpn connect** *name* **Example:** `Router# crypto ipsec client ezvpn connect easy vpn remote1`	Connects a given Cisco Easy VPN remote configuration. • The *name* argument specifies the IPSec VPN tunnel name. **Note** If the tunnel name is not specified, the active tunnel is connected. If there is more than one active tunnel, the command fails with an error requesting that you specify the tunnel name.

Configuring Automatic Tunnel Control

To configure automatic tunnel control, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec client ezvpn** *name*
4. **connect** [**auto** | **manual**]
5. **exit**
6. **exit**
7. **crypto ipsec client ezvpn connect** *name*

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec client ezvpn** *name* **Example:** `Router (config)# crypto ipsec client ezvpn easy vpn remote1`	Assigns a Cisco Easy VPN remote configuration to an interface and enters Cisco Easy VPN Remote configuration mode. • Specify the configuration name to be assigned to the interface.
Step 4	**connect** [**auto** | **manual**] **Example:** `Router (config-crypto-ezvpn)# connect auto`	Connects the VPN tunnel. • Specify **auto** to configure automatic tunnel control. Automatic is the default; you do not need to use this subcommand if your configuration is automatic.
Step 5	**exit** **Example:** `Router (config-crypto-ezvpn)# exit`	Exits Cisco Easy VPN Remote configuration mode.
Step 6	**exit** **Example:** `Router (config)# exit`	Exits global configuration mode and enters privileged EXEC mode.

	Command	Purpose
Step 7	**crypto ipsec client ezvpn connect** *name* **Example:** `Router# crypto ipsec client ezvpn connect easy vpn remote1`	Connects a given Cisco Easy VPN remote configuration. • The *name* argument specifies the IPSec VPN tunnel name. **Note** If the tunnel name is not specified, the active tunnel is connected. If there is more than one active tunnel, the command fails with an error requesting that you specify the tunnel name.

Configuring Multiple Inside Interfaces

You can configure up to three inside interfaces for all platforms. You need to manually configure each inside interface using the following procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-name*
4. **exit**
5. **crypto ipsec client ezvpn** *name* [**outside** | **inside**]
6. **interface** *interface-name*
7. **exit**
8. **crypto ipsec client ezvpn** *name* [**outside** | **inside**]

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-name* **Example:** `Router (config)# interface Ethernet0`	Selects the interface you want to configure by specifying the interface name and enters interface configuration mode.
Step 4	**exit** **Example:** `Router (config-if)# exit`	Exits interface configuration mode.

	Command	Purpose
Step 5	**crypto ipsec client ezvpn** *name* [**outside** | **inside**] **Example:** `Router (config)# crypto ipsec client ezvpn easy vpn remote 1 inside`	Specifies the Cisco Easy VPN remote configuration name to be assigned to the first inside interface. • You must specify **inside** for each inside interface.
Step 6	**interface** *interface-name* **Example:** `Router (config)# interface Ethernet1`	Selects the next interface you want to configure by specifying the next interface name and enters interface configuration mode.
Step 7	**exit** **Example:** `Router (config-if)# exit`	Exits interface configuration mode.
Step 8	**crypto ipsec client ezvpn** *name* [**outside** | **inside**] **Example:** `Router (config)# crypto ipsec client ezvpn easy vpn remote2 inside`	Specifies the Cisco Easy VPN remote configuration name to be assigned to the next inside interface. • You must specify **inside** for each inside interface. Repeat Step 3 through Step 4 to configure an additional tunnel if desired.

Configuring Multiple Outside Interfaces

You can configure multiple tunnels for outside interfaces, setting up a tunnel for each outside interface. You can configure a maximum of four tunnels using the following procedure for each outside interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-name*
4. **exit**
5. **crypto ipsec client ezvpn** *name* [**outside | inside**]
6. **interface** *interface-name*
7. **exit**
8. **crypto ipsec client ezvpn** *name* [**outside | inside**]

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-name* **Example:** `Router (config)# interface Ethernet0`	Selects the first outside interface you want to configure by specifying the interface name and enters interface configuration mode.
Step 4	**exit** **Example:** `Router (config-if)# exit`	Exits interface configuration mode.
Step 5	**crypto ipsec client ezvpn** *name* [**outside** | **inside**] **Example:** `Router (config)# crypto ipsec client ezvpn easy vpn remote1 outside`	Specifies the Cisco Easy VPN remote configuration name to be assigned to the first outside interface. • Specify **outside** (optional) for each outside interface. If neither **outside** nor **inside** is specified for the interface, the default is **outside**.
Step 6	**interface** *interface-name* **Example:** `Router (config)# interface Ethernet1`	Selects the next outside interface you want to configure by specifying the next interface name.
Step 7	**exit** **Example:** `Router (config-if)# exit`	Exits interface configuration mode.
Step 8	**crypto ipsec client ezvpn** *name* [**outside** | **inside**] **Example:** `Router (config)# crypto ipsec client ezvpn easy vpn remote2 outside`	Specifies the Cisco Easy VPN remote configuration name to be assigned to the next outside interface. • Specify **outside** (optional) for each outside interface. If neither **outside** nor **inside** is specified for the interface, the default is **outside**. Repeat Step 3 through Step 4 to configure additional tunnels if desired.

Configuring Multiple Subnet Support

When configuring multiple subnet support, you must first configure an access list to define the actual subnets to be protected. Each source subnet or mask pair indicates that all traffic that is sourced from this network to any destination is protected by IPSec. For information about configuring ACLs, see "Access control lists, configuring" in the section "Additional References."

After you have defined the subnets, you must configure the crypto IPSec client EZVPN profile to use the ACLs.

Note Multiple subnets are not supported in client mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-name*
4. **exit**
5. **crypto ipsec client ezvpn** *name*
6. **acl** {*acl-name* | *acl-number*}

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-name* **Example:** `Router (config)# interface Ethernet1`	Selects the interface you want to configure by specifying the interface name and enters interface configuration mode.
Step 4	**exit** **Example:** `Router (config-if)# exit`	Exits interface configuration mode.
Step 5	**crypto ipsec client ezvpn** *name* **Example:** `Router (config)# crypto ipsec client ezvpn ez1`	Creates a Cisco Easy VPN remote configuration and enters crypto Easy VPN configuration mode.
Step 6	**acl** {*acl-name* | *acl-number*} **Example:** `Router (config-crypto-ezvpn)# acl acl-list1`	Specifies multiple subnets in a VPN tunnel.

Configuring Proxy DNS Server Support

As a way of implementing the use of the DNS addresses of the ISP when the WAN connection is down, the router in a Cisco Easy VPN remote configuration can be configured to act as a proxy DNS server. To enable the proxy DNS server functionality with the **ip dns server** command, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dns server**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip dns server** **Example:** `Router (config)# ip dns server`	Enables the router to act as a proxy DNS server. **Note** This definition is IOS specific.

What to Do Next

After configuring the router, you configure the Cisco IOS Easy VPN server as follows:

- Under the **crypto isakmp client configuration group** command, configure the **dns** subcommand as in the following example:

  dns *A.B.C.D A1.B1.C1.D1*

  These DNS server addresses should be pushed from the server to the Cisco Easy VPN remote and dynamically added to or deleted from the running configuration of the router.

For information about general DNS server functionality in Cisco IOS software applications, see *Configuring DNS* and *Configuring DNS on Cisco Routers*.

Configuring Dial Backup

Note The Dial Backup feature is not available in Cisco IOS Release 12.3(11)T.

To configure dial backup, perform the following steps.

SUMMARY STEPS

1. Create the Easy VPN backup configuration.
2. Add the backup subcommand details to the primary configuration.
3. Apply the backup Easy VPN configuration to the dial backup outside interface.
4. Apply the Easy VPN profile to the inside interfaces.

DETAILED STEPS

	Command	Purpose
Step 1	Create the Easy VPN dial backup configuration.	For details about the backup configuration, see the section "Dial Backup."
Step 2	Add the backup subcommand details to the primary configuration.	Use the **backup** subcommand and **track** keyword of the **crypto ipsec client ezvpn** command.
Step 3	Apply the backup Easy VPN configuration to the dial backup outside interface (for example, serial, async, or dialer).	For details about applying the backup configuration to the dial backup outside interface, see the section "Configuring Multiple Outside Interfaces."
Step 4	Apply the Easy VPN profile to the inside interfaces (there can be more than one).	For details about applying the Easy VPN profile to the inside interfaces, see the section "Configuring Multiple Inside Interfaces."

Configuring the DHCP Server Pool

To configure the Dynamic Host Configuration Protocol (DHCP) server pool, see the chapter "Configuring DHCP" in the *Cisco IOS IP Configuration Guide*, Release 12.3.

Resetting a VPN Connection

To reset the VPN connection, perform the following steps. The **clear** commands can be configured in any order or independent of each other.

SUMMARY STEPS

1. **enable**
2. **clear crypto ipsec client ezvpn**
3. **clear crypto sa**
4. **clear crypto isakmp**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear crypto ipsec client ezvpn** **Example:** `Router# clear crypto ipsec client ezvpn`	Resets the Cisco Easy VPN remote state machine and brings down the Cisco Easy VPN remote connection on all interfaces or on a given interface (tunnel).
Step 3	**clear crypto sa** **Example:** `Router# clear crypto sa`	Deletes IPSec SAs.
Step 4	**clear crypto isakmp** **Example:** `Router# clear crypto isakmp`	Clear active IKE connections.

Monitoring and Maintaining VPN and IKE Events

To monitor and maintain VPN and IKE events, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **debug crypto ipsec client ezvpn**
3. **debug crypto ipsec**
4. **debug crypto isakmp**

SUMMARY STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto ipsec client ezvpn** **Example:** `Router# debug crypto ipsec client ezvpn`	Displays information showing the configuration and implementation of the Cisco Easy VPN Remote feature.
Step 3	**debug crypto ipsec** **Example:** `Router# debug crypto ipsec`	Displays IPSec events.
Step 4	**debug crypto isakmp** **Example:** `Router# debug crypto isakmp`	Displays messages about IKE events.

Easy VPN Server Tasks

Configuring a Cisco IOS Easy VPN Server

For information about configuring the Easy VPN Server, see the following document:

- *Easy VPN Server*

Configuring an Easy VPN Server on a VPN 3000 Series Concentrator

This section describes the guidelines required to configure the Cisco VPN 3000 series concentrator for use with the Cisco Easy VPN Remote feature. As a general rule, you can use the default configuration except for IP addresses, server addresses, routing configurations, and for the following parameters and options:

- Peer Configuration on a Cisco Easy VPN Remote Using the Hostname, page 988
- Interactive Hardware Client Authentication Version 3.5, page 988
- IPSec Tunnel Protocol, page 988
- IPSec Group, page 988
- Group Lock, page 988
- Xauth, page 989
- Split Tunneling, page 989
- IKE Proposals, page 989
- New IPSec SA, page 989

Note You must be using Cisco VPN 3000 series concentrator software Release 3.11 or later to support Cisco Easy VPN software clients and remotes.

Peer Configuration on a Cisco Easy VPN Remote Using the Hostname

After you have configured the Cisco Easy VPN server on the VPN 3000 concentrator to use hostname as its identity, you must configure the peer on the Cisco Easy VPN remote using the hostname. You can either configure DNS on the client to resolve the peer hostname or configure the peer hostname locally on the client using the **ip host** command. As an example, you can configure the peer hostname locally on an Easy VPN remote as follows:

```
ip host crypto-gw.cisco.com 10.0.0.1
```

Or you can configure the Easy VPN remote to use the hostname with the **peer** command and *hostname* argument, as follows:

```
peer crypto-gw.cisco.com.
```

Interactive Hardware Client Authentication Version 3.5

The Cisco Easy VPN Remote feature does not support the Interactive Hardware Client Authentication Version 3.5 feature. This feature must be disabled. You can disable the feature on the VPN 3000 series concentrator by clicking the HW Client tab on the Configuration | User Management | Base Group screen.

IPSec Tunnel Protocol

IPSec Tunnel Protocol enables the IPSec tunnel protocol so that it is available for users. The IPSec Tunnel Protocol is configured on the Cisco VPN 3000 series concentrator by clicking the General tab on the Configuration | User Management | Base Group screen.

IPSec Group

IPSec group configures the Cisco VPN 3000 series concentrator with a group name and password that match the values configured for the Cisco Easy VPN remote configuration on the router. These values are configured on the router with the **group** *group-name* **key** *group-key* subcommand and arguments. The values are configured on the Cisco VPN 3000 series concentrator using the Configuration | User Management | Groups screen.

Group Lock

If you are defining multiple users in multiple groups on the VPN 3000 series concentrator, you must check the Group Lock box in the IPSec tab to prevent users in one group from logging in with the parameters of another group. For example, if you have configured one group for split tunneling access and another group without split tunneling access, clicking the Group Lock box prevents users in the second group from gaining access to the split tunneling features. The Group Lock checkbox appears in the IPSec tab in the Configuration | User Management | Base Group screen and in the IPSec tab in the Configuration | User Management | Groups | Add/Modify screens.

Xauth

To use Xauth, set the Authentication parameter to None. The Authentication parameter appears in the IPSec tab in the Configuration | User Management | Base Group screen and in the IPSec tab in the Configuration | User Management | Groups | Add/Modify screens.

Split Tunneling

The Configuration | User Management | Base Group, Mode Configuration Parameters Tab screen includes a Split Tunnel option with a checkbox that says “Allow the networks in the list to bypass the tunnel.”

IKE Proposals

The Cisco VPN 3000 series concentrator is preconfigured with a default IKE proposal, CiscoVPNClient-3DES-MD5, that can be used with Cisco Easy VPN remotes. This IKE proposal supports preshared keys with Xauth using the MD5/HMAC-128 algorithm and Diffie-Hellman Group 2.

This IKE proposal is active by default, but you should verify that it is still an active proposal using the Configuration | System | Tunneling Protocols | IPSec | IKE Proposals screen.

In addition, as part of configuring the Cisco VPN 3000 series concentrator—for the Cisco Easy VPN Remote image, you do not need to create a new IPSec SA. Use the default IKE and Easy VPN remote lifetime configured on the Cisco VPN 3000 series concentrator.

Note You can also use the default IKE proposals IKE-DES-MD5 and IKE-3DES-MD5, but they do not enable Xauth support by default.

New IPSec SA

You can create a new IPSec SA. Cisco Easy VPN clients use a SA having the following parameters:

- Authentication Algorithm=ESP/MD5/HMAC-128
- Encryption Algorithm=DES-56 or 3DES-168 (recommended)
- Encapsulation Mode=Tunnel
- IKE Proposal=CiscoVPNClient-3DES-MD5 (preferred)

The Cisco VPN 3000 series concentrator is preconfigured with several default security associations (SAs), but they do not meet the IKE proposal requirements. To use an IKE proposal of CiscoVPNClient-3DES-MD5, copy the ESP/IKE-3DES-MD5 SA and modify it to use CiscoVPNClient-3DES-MD5 as its IKE proposal. An IKE proposal is configured on the VPN 3000 series concentrator using the Configuration | Policy Management | Traffic Management | Security Associations screen.

Configuring an Easy VPN Server on a Cisco PIX Firewall

For information about configuring an Easy VPN Server on a Cisco PIX Firewall, see the following document:

- *Easy VPN Server*

Web Interface Tasks

Configuring Web-Based Activation

To configure a LAN so that any HTTP requests coming from any of the PCs on the private LAN are intercepted, providing corporate users with access to the corporate Web page, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec client ezvpn** *name*
4. **xauth userid mode** {**http-intercept** | **interactive** | **local**}

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec client ezvpn** *name* **Example:** `Router (config)# crypto ipsec client ezvpn easy vpn remote1`	Assigns a Cisco Easy VPN remote configuration to an interface and enters Cisco Easy VPN Remote configuration mode. • The *name* argument specifies the configuration name to be assigned to the interface.
Step 4	**xauth userid mode** {**http-intercept** | **interactive** | **local**} **Example:** `Router (config-crypto-ezvpn)# xauth userid mode http-intercept`	Specifies how the VPN device handles Xauth requests or prompts from the server.

Monitoring and Maintaining Web-Based Activation

To monitor and maintain web-based activation, perform the following steps. (The **debug** and **show** commands may be used independently, or they may all be configured.)

SUMMARY STEPS

1. **enable**
2. **debug crypto ipsec client ezvpn**

3. **debug ip auth-proxy ezvpn**
4. **show crypto ipsec client ezvpn**
5. **show ip auth-proxy config**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto ipsec client ezvpn** **Example:** `Router# debug crypto ipsec client ezvpn`	Displays information about the Cisco Easy VPN connection.
Step 3	**debug ip auth-proxy ezvpn** **Example:** `Router# debug ip auth-proxy ezvpn`	Displays information related to proxy authentication behavior for web-based activation.
Step 4	**show crypto ipsec client ezvpn** **Example:** `Router# show crypto ipsec client ezvpn`	Shows that the username and password used for user credentials during Xauth negotiations will be obtained by intercepting HTTP connections from the user.
Step 5	**show ip auth-proxy config** **Example:** `Router# show ip auth-proxy config`	Displays the auth-proxy rule that has been created and applied by Easy VPN.

Examples

Debug Output

The following is sample **debug** output for a typical situation in which a user has opened a browser and connected to the corporate website:

```
Router# debug ip auth-proxy ezvpn

Dec 10 12:41:13.335: AUTH-PROXY: New request received by EzVPN WebIntercept
! The following line shows the ip address of the user.
from 10.4.205.205
Dec 10 12:41:13.335: AUTH-PROXY:GET request received
Dec 10 12:41:13.335: AUTH-PROXY:Normal auth scheme in operation
Dec 10 12:41:13.335: AUTH-PROXY:Ezvpn is NOT active. Sending connect-bypass page to user
```

At this point, the user chooses "connect" on his or her browser:

```
Dec 10 12:42:43.427: AUTH-PROXY: New request received by EzVPN WebIntercept
from 9.4.205.205
Dec 10 12:42:43.427: AUTH-PROXY:POST request received
Dec 10 12:42:43.639: AUTH-PROXY:Found attribute <connect> in form
Dec 10 12:42:43.639: AUTH-PROXY:Sending POST data to EzVPN
Dec 10 12:42:43.639: EZVPN(tunnel22): Communication from Interceptor
  application.
```

```
Request/Response from 10.4.205.205, via Ethernet0
Dec 10 12:42:43.639:         connect: Connect Now
Dec 10 12:42:43.639: EZVPN(tunnel22): Received CONNECT from 10.4.205.205!
Dec 10 12:42:43.643: EZVPN(tunnel22): Current State: CONNECT_REQUIRED
Dec 10 12:42:43.643: EZVPN(tunnel22): Event: CONNECT
Dec 10 12:42:43.643: EZVPN(tunnel22): ezvpn_connect_request
```

Easy VPN contacts the server:

```
Dec 10 12:42:43.643: EZVPN(tunnel22): Found valid peer 192.0.0.1
Dec 10 12:42:43.643: EZVPN(tunnel22): Added PSK for address 192.0.0.1

Dec 10 12:42:43.643: EZVPN(tunnel22): New State: READY
Dec 10 12:42:44.815: EZVPN(tunnel22): Current State: READY
Dec 10 12:42:44.815: EZVPN(tunnel22): Event: IKE_PFS
Dec 10 12:42:44.815: EZVPN(tunnel22): No state change
Dec 10 12:42:44.819: EZVPN(tunnel22): Current State: READY
Dec 10 12:42:44.819: EZVPN(tunnel22): Event: CONN_UP
Dec 10 12:42:44.819: EZVPN(tunnel22): ezvpn_conn_up B8E86EC7 E88A8A18 D0D51422
  8AFF32B7
```

The server requests Xauth information:

```
Dec 10 12:42:44.823: EZVPN(tunnel22): No state change
Dec 10 12:42:44.827: EZVPN(tunnel22): Current State: READY
Dec 10 12:42:44.831: EZVPN(tunnel22): Event: XAUTH_REQUEST
Dec 10 12:42:44.831: EZVPN(tunnel22): ezvpn_xauth_request
Dec 10 12:42:44.831: EZVPN(tunnel22): ezvpn_parse_xauth_msg
Dec 10 12:42:44.831: EZVPN: Attributes sent in xauth request message:
Dec 10 12:42:44.831:         XAUTH_TYPE_V2(tunnel22): 0
Dec 10 12:42:44.831:         XAUTH_USER_NAME_V2(tunnel22):
Dec 10 12:42:44.831:         XAUTH_USER_PASSWORD_V2(tunnel22):
Dec 10 12:42:44.831:         XAUTH_MESSAGE_V2(tunnel22) <Enter Username and
Password.>
Dec 10 12:42:44.831: EZVPN(tunnel22): Requesting following info for xauth
Dec 10 12:42:44.831:         username:(Null)
Dec 10 12:42:44.835:         password:(Null)
Dec 10 12:42:44.835:         message:Enter Username and Password.
Dec 10 12:42:44.835: EZVPN(tunnel22): New State: XAUTH_REQ
```

The username and password prompt are displayed in the browser of the user:

```
Dec 10 12:42:44.835: AUTH-PROXY: Response to POST  is CONTINUE
Dec 10 12:42:44.839: AUTH-PROXY: Displayed POST response successfully
Dec 10 12:42:44.843: AUTH-PROXY:Served POST response to the user
```

When the user enters his or her username and password, the following is sent to the server:

```
Dec 10 12:42:55.343: AUTH-PROXY: New request received by EzVPN WebIntercept
  from 10.4.205.205
Dec 10 12:42:55.347: AUTH-PROXY:POST request received
Dec 10 12:42:55.559: AUTH-PROXY:No of POST parameters is 3
Dec 10 12:42:55.559: AUTH-PROXY:Found attribute <username> in form
Dec 10 12:42:55.559: AUTH-PROXY:Found attribute <password> in form
Dec 10 12:42:55.559: AUTH-PROXY:Found attribute <ok> in form
Dec 10 12:42:55.563: AUTH-PROXY:Sending POST data to EzVPN
Dec 10 12:42:55.563: EZVPN(tunnel22): Communication from Interceptor application.
Request/Response from 10.4.205.205, via Ethernet0
Dec 10 12:42:55.563:         username:http
Dec 10 12:42:55.563:         password:<omitted>
Dec 10 12:42:55.563:         ok:Continue
Dec 10 12:42:55.563: EZVPN(tunnel22): Received usename|password from 10.4.205.205!
Dec 10 12:42:55.567: EZVPN(tunnel22): Current State: XAUTH_PROMPT
Dec 10 12:42:55.567: EZVPN(tunnel22): Event: XAUTH_REQ_INFO_READY
Dec 10 12:42:55.567: EZVPN(tunnel22): ezvpn_xauth_reply
```

```
Dec 10 12:42:55.567:         XAUTH_TYPE_V2(tunnel22): 0
Dec 10 12:42:55.567:         XAUTH_USER_NAME_V2(tunnel22): http
Dec 10 12:42:55.567:         XAUTH_USER_PASSWORD_V2(tunnel22): <omitted>
Dec 10 12:42:55.567: EZVPN(tunnel22): New State: XAUTH_REPLIED
Dec 10 12:42:55.891: EZVPN(tunnel22): Current State: XAUTH_REPLIED
Dec 10 12:42:55.891: EZVPN(tunnel22): Event: XAUTH_STATUS
Dec 10 12:42:55.891: EZVPN(tunnel22): xauth status received: Success
```

After using the tunnel, the user chooses "Disconnect":

```
Dec 10 12:48:17.267: EZVPN(tunnel22): Received authentic disconnect credential
Dec 10 12:48:17.275: EZVPN(): Received an HTTP request: disconnect
Dec 10 12:48:17.275: %CRYPTO-6-EZVPN_CONNECTION_DOWN: (Client)  User=
  Group=tunnel22  Client_public_addr=192.168.0.13  Server_public_addr=192.168.0.1
  Assigned_client_addr=10.3.4.5
```

Show Output Before the User Is Connected to the Tunnel

The following output from the two **show** commands (**show crypto ipsec client ezvpn** and **show ip auth-proxy config**) displays what you might see before a user is connected to a VPN tunnel:

```
Router# show crypto ipsec client ezvpn tunnel22

Tunnel name : tunnel22
Inside interface list: Ethernet0
Outside interface: Ethernet1
Current State: CONNECT_REQUIRED
Last Event: RESET
Save Password: Disallowed
! Note the next line.
        XAuth credentials: HTTP intercepted
        HTTP return code : 200
        IP addr being prompted: 0.0.0.0
Current EzVPN Peer: 192.0.0.1

Router# show ip auth-proxy config

Authentication global cache time is 60 minutes
Authentication global absolute time is 0 minutes
Authentication Proxy Watch-list is disabled

Authentication Proxy Rule Configuration
! Note that the next line is the Easy VPN-defined internal rule.
    Auth-proxy name ezvpn401***
      Applied on Ethernet0
      http list not specified inactivity-timer 60 minutes
```

Show Output After the User Is Connected to the Tunnel

The following output from the two **show** commands (**show crypto ipsec client evpn** and **show ip auth-proxy config**) displays what you might see after the user has been connected to the tunnel:

```
Router# show crypto ipsec client ezvpn tunnel22

Tunnel name : tunnel22
Inside interface list: Ethernet0
Outside interface: Ethernet1
Current State: IPSEC_ACTIVE
Last Event: SOCKET_UP
Address: 10.3.4.5
Mask: 255.255.255.255
Save Password: Disallowed
        XAuth credentials: HTTP intercepted
        HTTP return code : 200
        IP addr being prompted: 192.0.0.0
```

```
Current EzVPN Peer: 192.0.0.1

Router# show ip auth-proxy config

Authentication global cache time is 60 minutes
Authentication global absolute time is 0 minutes
Authentication Proxy Watch-list is disabled

Auth-proxy name ezvpnWeb*** (EzVPN-defined internal rule)
http list not specified inactivity-timer 60 minutes
```

Using SDM As a Web Manager

For information about the SDM web manager, see the following document:

- *Cisco Security Device Manager*

Troubleshooting the VPN Connection

Troubleshooting a VPN Connection Using the Cisco Easy VPN Remote Feature

To troubleshoot a VPN connection created using the Cisco Easy VPN Remote feature, use the following suggested techniques.

- Be aware that any changes to an active Cisco Easy VPN remote configuration or IP address changes to the involved interfaces, such as adding or removing an inside interface, result in a reset of the Cisco Easy VPN Remote connection.
- Enable debugging of the Cisco Easy VPN Remote feature using the **debug crypto ipsec client ezvpn** command.
- Enable debugging of IKE events using the **debug crypto ipsec** and **debug crypto isakmp** commands.
- Display the active IPSec VPN connections using the **show crypto engine connections active** command.
- To reset the VPN connection, use the **clear crypto ipsec client ezvpn** command. If you have debugging enabled, you might prefer to use the **clear crypto sa** and **clear crypto isakmp** commands.

Troubleshooting the Client Mode of Operation

The following information may be used to troubleshoot the Easy VPN Remote configuration for the client mode of operation.

In client mode, the Cisco Easy VPN Remote feature automatically configures the NAT or PAT translation and access lists that are needed to implement the VPN tunnel. These configurations are automatically created when the IPSec VPN connection is initiated. When the tunnel is torn down, the NAT or PAT and access list configurations are automatically deleted.

The NAT or PAT configuration is created with the following assumptions:

- The **ip nat inside** command is applied to all inside interfaces, including default inside interfaces. The default inside interface is the Ethernet 0 interface (for the Cisco 806, Cisco 826, Cisco 827, Cisco 828, Cisco 831, Cisco 836, and Cisco 837 routers).

- The **ip nat outside** command is applied to the interface that is configured with the Cisco Easy VPN Remote configuration. On the Cisco 800 series and Cisco 1700 series routers, the outside interface is configured with the Cisco Easy VPN Remote configuration. On the Cisco 1700 series routers, Cisco 2600 series routers, Cisco 3600 series routers, and Cisco 3700 series routers, multiple outside interfaces can be configured.

Tip The NAT or PAT translation and access list configurations that are created by the Cisco Easy VPN Remote feature are not written to either the startup configuration or running configuration files. These configurations, however, can be displayed using the **show ip nat statistics** and **show access-list** commands.

Troubleshooting Remote Management

To troubleshoot remote management of the VPN remote, use the **show ip interface** command. Using the **brief** keyword, you can verify that the loopback has been removed and that the interface is shown correctly.

Examples

Following is a typical example of output from the **show ip interface** command.

```
Router# show ip interface brief

Interface                  IP-Address      OK? Method Status                Protocol
Ethernet0                  unassigned      YES NVRAM  administratively down down

Ethernet1                  10.0.0.11       YES NVRAM  up                    up

Loopback0                  192.168.6.1     YES manual up                    up

Loopback1                  12.12.12.12     YES NVRAM  up                    up

Router# show ip interface brief

Interface                  IP-Address      OK? Method Status                Protocol
Ethernet0                  unassigned      YES NVRAM  administratively down down

Ethernet1                  10.0.0.11       YES NVRAM  up                    up

Loopback1                  12.12.12.12     YES NVRAM  up                    up
```

Troubleshooting Dead Peer Detection

To troubleshoot dead peer detection, use the **show crypto ipsec client ezvpn** command.

Examples

The following typical output displays the current server and the peers that have been pushed by the Easy VPN server:

```
Router# show crypto ipsec client ezvpn

Easy VPN Remote Phase: 4
Tunnel name : ez1
Inside interface list: Loopback1,
Outside interface: Ethernet1
```

```
Current State: IPSEC_ACTIVE
Last Event: CONNECT
Address: 192.168.6.5
Mask: 255.255.255.255
DNS Primary: 10.2.2.2
DNS Secondary: 10.2.2.3
NBMS/WINS Primary: 10.6.6.6
Default Domain: cisco.com
Save Password: Allowed
Current EzVPN Peer:10.0.0.110
Backup Gateways
 (0): green.cisco.com
 (1): blue
```

Configuration Examples for Cisco Easy VPN Remote

This section provides the following configuration examples.

Easy VPN Remote Configuration Examples

- Client Mode Configuration: Examples, page 996
- Local Address Support for Easy VPN Remote: Example, page 1002
- Network Extension Mode Configuration: Examples, page 1003
- Save Password Configuration: Example, page 1007
- PFS Support: Examples, page 1008
- Dial Backup: Examples, page 1008
- Web-Based Activation: Example, page 1014

Easy VPN Server Configuration Examples

- Cisco Easy VPN Server Without Split Tunneling: Example, page 1015
- Cisco Easy VPN Server Configuration with Split Tunneling: Example, page 1016
- Cisco Easy VPN Server Configuration with Xauth: Example, page 1018
- Easy VPN Server Interoperability Support: Example, page 1020

Easy VPN Remote Configuration Examples

Client Mode Configuration: Examples

The examples in this section show configurations for the Cisco Easy VPN Remote feature in client mode. Also shown are the Cisco IOS Easy VPN server configurations that correspond to these client configurations.

- Cisco Easy VPN Client in Client Mode (Cisco 831): Example, page 997
- Cisco Easy VPN Client in Client Mode (Cisco 837): Example, page 998
- Cisco Easy VPN Client in Client Mode (Cisco 1700 Series): Example, page 1000

For more client-mode configuration examples, see *IPSec VPN* (under the “Technical Documents” and “Cisco IOS IPSec Configuration Documents” sections) and to *Cisco Easy VPN Solutions*.

Note Typically, users configure the Cisco 800 series routers with the SDM or CRWS web interface, not by entering CLI commands. However, the configurations shown here for the Cisco 800 series routers display typical configurations that can be used if manual configuration is desired.

Cisco Easy VPN Client in Client Mode (Cisco 831): Example

In the following example, a Cisco 831 router is configured as an Easy VPN remote using the Cisco Easy VPN Remote feature in client mode. This example shows the following components of the Cisco Easy VPN Remote configuration:

- DHCP server pool—The **ip dhcp pool** command creates a pool of IP addresses to be assigned to the PCs connected to the Ethernet 0 interface of the router. The pool assigns addresses in the class C private address space (192.168.100.0) and configures each PC so that its default route is 192.168.100.1, which is the IP address assigned to the Ethernet interface of the router. The DHCP lease period is one day.
- Cisco Easy VPN remote configuration—The first **crypto ipsec client ezvpn easy vpn remote** command (global configuration mode) creates a Cisco Easy VPN remote configuration named "easy vpn remote." This configuration specifies the group name "easy vpn remote-groupname" and the shared key value "easy vpn remote-password," and it sets the peer destination to the IP address 192.185.0.5 (which is the address assigned to the interface connected to the Internet on the destination peer router). The Cisco Easy VPN Remote configuration is configured for the default **client** mode.

  Note If DNS is also configured on the router, the **peer** option also supports a hostname instead of an IP address.

- The second **crypto ipsec client ezvpn easy vpn remote** command (interface configuration mode) assigns the Cisco Easy VPN Remote configuration to the Ethernet 1 interface so that all traffic that is received and transmitted on that interface is sent through the VPN tunnel.

```
! Cisco Router Web Setup Template
!
no service pad
no service tcp-small-servers
no service udp-small-servers
service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname 806Router
!
!
ip subnet-zero
ip domain-lookup
ip dhcp excluded-address 10.10.10.1
!
ip dhcp pool CLIENT
   import all
   network 10.10.10.0 255.255.255.0
   default-router 10.10.10.1
   lease 1 0 0
!
!
crypto ipsec client ezvpn easy_vpn_remote
 peer 192.185.0.5
```

```
 group easy_vpn_remote_groupname key easy_vpn_remote_password
 mode client
!
!
interface Ethernet0
 ip address 10.10.10.1 255.255.255.0
 no cdp enable
 hold-queue 32 in
!
interface Ethernet1
 ip address dhcp
 no cdp enable
 crypto ipsec client ezvpn easy_vpn_remote
!
ip classless
ip http server
!
!
ip route 0.0.0.0 0.0.0.0 Ethernet1
!
line con 0
 exec-timeout 120 0
 stopbits 1
line vty 0 4
 exec-timeout 0 0
 login local
```

Cisco Easy VPN Client in Client Mode (Cisco 837): Example

In the following example, a Cisco 837 router is configured as an Easy VPN remote using the Cisco Easy VPN Remote feature in the client mode of operation. This example shows the following components of the Cisco Easy VPN remote configuration:

- PPPoE configuration—The ATM 0 interface is configured to support PPPoE connections over the Dialer 1 virtual interface. Because the interfaces use PPPoE, a DHCP IP address pool is not required to provide IP addresses to the connected PCs.
- Cisco Easy VPN Remote configuration—The first **crypto ipsec client ezvpn** command (global configuration mode) creates a Cisco Easy VPN remote configuration named "easy vpn remote." This configuration specifies the group name "easy vpn remote-groupname" and the shared key value of "easy vpn remote-password," and it sets the peer destination to the IP address 10.0.0.5 (which is the address assigned to the interface connected to the Internet on the destination peer router). The Cisco Easy VPN remote configuration is configured for the default client mode.

  Note If DNS is also configured on the router, the **peer** option also supports a hostname instead of an IP address.

- The second **crypto ipsec client ezvpn** command (interface configuration mode) assigns the Cisco Easy VPN remote configuration to the Dialer 1 interface so that all traffic received and transmitted on that interface is sent through the VPN tunnel.

```
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname c827
!
```

```
!
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
ip subnet-zero
!
ip ssh time-out 120
ip ssh authentication-retries 3
vpdn enable
!
vpdn-group pppoe
 request-dialin
  protocol pppoe
 ip mtu adjust
!!
!
crypto ipsec client ezvpn easy_vpn_remote
 group easy_vpn_remote_groupname key easy_vpn_remote_password
 mode client
 peer 10.0.0.5
!!
!
interface Ethernet0
 ip address 10.0.0.117 255.0.0.0
 hold-queue 100 out
!
interface ATM0
 no ip address
 no atm ilmi-keepalive
 pvc 1/40
  pppoe-client dial-pool-number 1
 !
 dsl operating-mode auto
!
interface Dialer1
 ip address 10.0.0.3 255.0.0.0
 ip mtu 1492
 encapsulation ppp
 dialer pool 1
 crypto ipsec client ezvpn easy_vpn_remote
!
ip classless
ip route 0.0.0.0 0.0.0.0 ATM0
ip route 0.0.0.0 0.0.0.0 Dialer1 permanent
ip route 10.0.0.0 255.0.0.0 12.0.0.13
ip http server
ip pim bidir-enable
!
line con 0
 stopbits 1
line vty 0 4
 login
!
scheduler max-task-time 5000
end
```

Cisco Easy VPN Client in Client Mode (Cisco 1700 Series): Example

In the following example, a Cisco 1753 router is configured as an Easy VPN remote using the Cisco Easy VPN Remote feature in the client mode of operation. This example shows a running configuration of a Cisco 1753 that has two inside interfaces and one outside interface on one tunnel. The **connect auto** subcommand manually establishes the IPSec VPN tunnel.

```
Router# show running-config

Building configuration...
Current configuration : 881 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname mma-1753
!
!
memory-size iomem 15
ip subnet-zero
!!
!
ip ssh time-out 120
ip ssh authentication-retries 3
! !
!
crypto ipsec client ezvpn easy_vpn_remote
connect auto
group ezvpn key ezvpn
mode client
peer 10.6.6.1
! !
!
interface FastEthernet0/0
ip address 10.4.4.2 255.255.255.0
speed auto
crypto ipsec client ezvpn easy_vpn_remote inside
!
interface Serial0/0
ip address 10.6.6.2 255.255.255.0
no fair-queue
crypto ipsec client ezvpn easy_vpn_remote
!
interface Serial1/0
ip address 10.5.5.2 255.255.255.0
clock rate 4000000
crypto ipsec client ezvpn easy_vpn_remote inside
!
ip classless
no ip http server
ip pim bidir-enable
! !
!
line con 0
line aux 0
line vty 0 4
login
!
end
```

The following example shows a running configuration of a Cisco 1760 router that has two active, automatically connected tunnels, easy vpn remote1 and easy vpn remote2. Tunnel easy vpn remote1 has two configured inside interfaces and one configured outside interface. Tunnel easy vpn remote2 has one configured inside interface and one configured outside interface. The example also shows the output for the **show crypto ipsect client ezvpn** command that lists the tunnel names and the outside and inside interfaces.

```
Router# show running-config

Building configuration...
Current configuration : 1246 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname 1760
!
aaa new-model
!
!
aaa session-id common
!
ip subnet-zero
!!
!
crypto ipsec client ezvpn easy_vpn_remote2
connect auto
group ez key ez
mode network-extension
peer 10.7.7.1
crypto ipsec client ezvpn easy_vpn_remote1
connect auto
group ezvpn key ezvpn
mode client
peer 10.6.6.1
! !
!
interface FastEthernet0/0
ip address 10.5.5.2 255.255.255.0
speed auto
no cdp enable
crypto ipsec client ezvpn easy_vpn_remote1 inside
!
interface Serial0/0
ip address 10.4.4.2 255.255.255.0
no ip route-cache
no ip mroute-cache
no fair-queue
no cdp enable
crypto ipsec client ezvpn easy_vpn_remote1 inside
!
interface Serial0/1
ip address 10.3.3.2 255.255.255.0
no cdp enable
crypto ipsec client ezvpn easy_vpn_remote2 inside
!
interface Serial1/0
ip address 10.6.6.2 255.255.255.0
clockrate 4000000
no cdp enable
crypto ipsec client ezvpn easy_vpn_remote1
```

```
!
interface Serial1/1
ip address 10.7.7.2 255.255.255.0
no keepalive
no cdp enable
crypto ipsec client ezvpn easy_vpn_remote2
!
ip classless
no ip http server
ip pim bidir-enable
!
!
radius-server retransmit 3
radius-server authorization permit missing Service-Type
!
line con 0
line aux 0
line vty 0 4
!
no scheduler allocate
end


Router# show crypto ipsec client ezvpn

Tunnel name : easy_vpn_remote1
Inside interface list: FastEthernet0/0, Serial0/0,
Outside interface: Serial1/0
Current State: IPSEC_ACTIVE
Last Event: SOCKET_UP
Address: 10.0.0.5
Mask: 255.255.255.255
Default Domain: cisco.com
Tunnel name : easy_vpn_remote2
Inside interface list: Serial0/1,
Outside interface: Serial1/1
Current State: IPSEC_ACTIVE
Last Event: SOCKET_UP
Default Domain: cisco.com
```

Local Address Support for Easy VPN Remote: Example

The following example shows that the **local-address** command is used to specify the loopback 0 interface for sourcing tunnel traffic:

```
Router# configure terminal
Router(config)# crypto ipsec client ezvpn telecommuter-client
Router(config-crypto-ezvpn)# local-address loopback0
```

Network Extension Mode Configuration: Examples

In this section, the following examples demonstrate how to configure the Cisco Easy VPN Remote feature in the network extension mode of operation. Also shown are the Cisco IOS Easy VPN server configurations that correspond to these client configurations.

- Cisco Easy VPN Client in Network Extension Mode (Cisco 831): Example, page 1003
- Cisco Easy VPN Client in Network Extension Mode (Cisco 837): Example, page 1004
- Cisco Easy VPN Client in Network Extension Mode (Cisco 1700 Series): Example, page 1006

For more network extension mode configuration examples, see *IPSec VPN* (under the "Technical Documents" and "Cisco IOS IPSec Configuration Documents" sections) and to *Cisco Easy VPN Solutions*.

Cisco Easy VPN Client in Network Extension Mode (Cisco 831): Example

In the following example, a Cisco 831 router is configured as an Easy VPN remote using the Cisco Easy VPN Remote feature. This example shows the following components of the Cisco Easy VPN remote configuration:

- The Ethernet 0 interface is assigned an address in the network address space of the Cisco IOS Easy VPN server. The **ip route** command directs all traffic for this network space from the Ethernet 1 interface to the destination server.
- Cisco Easy VPN Remote configuration—The first **crypto ipsec client ezvpn** command (global configuration mode) creates a Cisco Easy VPN remote configuration named "easy vpn remote." This configuration specifies the group name "easy vpn remote-groupname" and the shared key value "easy vpn remote-password," and it sets the peer destination to the IP address 192.185.0.5 (which is the address assigned to the interface connected to the Internet on the destination peer router). The Cisco Easy VPN remote configuration is configured for network extension mode.

  Note If DNS is also configured on the router, the **peer** option also supports a hostname instead of an IP address.

- The second **crypto ipsec client ezvpn** command (interface configuration mode) assigns the Cisco Easy VPN Remote configuration to the Ethernet 1 interface so that all traffic that is received and transmitted on that interface is sent through the VPN tunnel.

```
! Cisco Router Web Setup Template
!
no service pad
no service tcp-small-servers
no service udp-small-servers
service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname Router
!
!
ip subnet-zero
ip domain-lookup
!
!
ip dhcp excluded-address 172.168.1.1
!
ip dhcp pool localpool
```

```
   import all
   network 172.168.1.0 255.255.255.248
   default-router 172.168.1.1
   lease 1 0 0
!
!
crypto ipsec client ezvpn easy_vpn_remote
 peer 192.185.0.5
 group easy_vpn_remote_groupname key easy_vpn_remote_password
 mode network-extension
!
!
interface Ethernet0
 ip address 172.168.1.1 255.255.255.192
 no cdp enable
 hold-queue 32 in
!
interface Ethernet1
 ip address dhcp
 no cdp enable
 crypto ipsec client ezvpn easy_vpn_remote
!
ip classless
ip route 172.168.0.0 255.255.255.128 Ethernet1
ip http server
!
!
line con 0
 exec-timeout 120 0
 stopbits 1
line vty 0 4
 exec-timeout 0 0
 login local
```

Cisco Easy VPN Client in Network Extension Mode (Cisco 837): Example

In the following example, a Cisco 837 router is configured as an Easy VPN remote using the Cisco Easy VPN Remote feature in client mode. This example shows the following components of the Cisco Easy VPN remote configuration:

- PPPoE configuration—The ATM 0 interface is configured to support PPPoE connections over the Dialer 1 virtual interface. Because the interfaces use PPPoE, a DHCP IP address pool is not required to provide IP addresses to the connected PCs.
- The Ethernet 0 interface is assigned an address in the network address space of the Cisco IOS Easy VPN server. The **ip route** command directs all traffic for this network space from the Dialer 1 interface to the destination server.
- Cisco Easy VPN Remote configuration—The first **crypto ipsec client ezvpn** command (global configuration mode) creates a Cisco Easy VPN remote configuration named “easy vpn remote.” This configuration specifies the group name “easy vpn remote-groupname” and the shared key value “easy vpn remote-password,” and it sets the peer destination to the IP address 10.0.0.5 (which is the address assigned to the interface connected to the Internet on the destination peer router). The Cisco Easy VPN remote configuration is configured for the default network extension mode.

Note If DNS is also configured on the router, the **peer** option also supports a hostname instead of an IP address.

- The second **crypto ipsec client ezvpn** command (interface configuration mode) assigns the Cisco Easy VPN remote configuration to the Dialer1 interface so that all traffic that is received and transmitted on that interface is sent through the VPN tunnel.

```
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname c827
!
!
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
ip subnet-zero
!
ip ssh time-out 120
ip ssh authentication-retries 3
vpdn enable
!
vpdn-group pppoe
 request-dialin
  protocol pppoe
 ip mtu adjust
!
!
crypto ipsec client ezvpn easy_vpn_remote
 group easy_vpn_remote_groupname key easy_vpn_remote_password
 mode network-extension
 peer 20.0.0.5
!
!
interface Ethernet0
 ip address 172.168.0.30 255.255.255.192
 hold-queue 100 out
!
interface ATM0
 no ip address
 no atm ilmi-keepalive
 pvc 1/40
  pppoe-client dial-pool-number 1
 !
 dsl operating-mode auto
!
interface Dialer1
 ip address 12.0.0.3 255.0.0.0
 ip mtu 1492
 encapsulation ppp
 dialer pool 1
 crypto ipsec client ezvpn easy_vpn_remote
!
ip classless
ip route 172.168.0.0 255.255.255.128 Dialer1
ip route 0.0.0.0 0.0.0.0 ATM0
ip route 0.0.0.0 0.0.0.0 Dialer1 permanent
ip route 20.0.0.0 255.0.0.0 12.0.0.13
ip http server
ip pim bidir-enable
!
line con 0
 stopbits 1
```

```
line vty 0 4
 login
!
scheduler max-task-time 5000
```

Cisco Easy VPN Client in Network Extension Mode (Cisco 1700 Series): Example

In the following example, a Cisco 1700 series router is configured as an Easy VPN remote using the Cisco Easy VPN Remote feature in the network extension mode of operation. This example shows the following components of the Cisco Easy VPN remote configuration:

- Cisco Easy VPN Remote configuration—The first **crypto ipsec client ezvpn** command (global configuration mode) creates a Cisco Easy VPN remote configuration that is named "easy vpn remote." This configuration specifies the group name "easy vpn remote-groupname" and the shared key value "easy vpn remote-password," and it sets the peer destination to the IP address 10.0.0.2 (which is the address assigned to the interface connected to the Internet on the destination peer router). The Cisco Easy VPN remote configuration is configured for network extension mode.

  Note If DNS is also configured on the router, the **peer** option also supports a hostname instead of an IP address.

- The second **crypto ipsec client ezvpn easy vpn remote** command (interface configuration mode) assigns the Cisco Easy VPN remote configuration to the Ethernet 0 interface so that all traffic that is received and transmitted on that interface is sent through the VPN tunnel.

```
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname 1710
!
!
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
ip subnet-zero
!
!
ip ssh time-out 120
ip ssh authentication-retries 3
!
!
ip dhcp excluded-address 70.0.0.10
!
ip dhcp pool localpool
   import all
   network 10.70.0.0 255.255.255.248
   default-router 10.70.0.10
   lease 1 0 0
!
!
crypto ipsec client ezvpn easy_vpn_remote
 group easy_vpn_remote_groupname key easy_vpn_remote_password
 mode network-extension
 peer 10.0.0.2
!
!
```

```
interface Ethernet0
 ip address 10.50.0.10 255.0.0.0
 half-duplex
 crypto ipsec client ezvpn easy_vpn_remote
!
interface FastEthernet0
 ip address 10.10.0.10 255.0.0.0
 speed auto
!
ip classless
ip route 10.20.0.0 255.0.0.0 Ethernet0
ip route 10.20.0.0 255.0.0.0 Ethernet0
no ip http server
ip pim bidir-enable
!!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 login
```

Save Password Configuration: Example

The following sample **show running-config** output shows that the Save Password feature has been configured (note the **password encryption aes** command and **username** keywords in the output):

```
Router# show running-config

133.CABLEMODEM.CISCO: Oct 28 18:42:07.115: %SYS-5-CONFIG_I: Configured from console by
consolen
Building configuration...

Current configuration : 1269 bytes
!
! Last configuration change at 14:42:07 UTC Tue Oct 28 2003
!
version 12.3
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
!
clock timezone UTC -4
no aaa new-model
ip subnet-zero
no ip routing
!
!
ip audit notify log
ip audit po max-events 100
ip ssh break-string
no ftp-server write-enable
password encryption aes
!
!
```

```
no crypto isakmp enable
!
!
crypto ipsec client ezvpn remote_vpn_client
 connect auto
 mode client
 username greentree password  6 ARiFgh`SOJfMHLK[MHMQJZagR\M
!
!
interface Ethernet0
 ip address 10.3.66.4 255.255.255.0
 no ip route-cache
 bridge-group 59
```

PFS Support: Examples

The following **show crypto ipsec client ezvpn** command output shows the group name ("2") and that PFS is being used:

```
Router# show crypto ipsec client ezvpn

Easy VPN Remote Phase: 4

Tunnel name : ez1
Inside interface list: Loopback1,
Outside interface: Ethernet1
Current State: IPSEC_ACTIVE
Last Event: SOCKET_UP
Address: 192.168.6.6
Mask: 255.255.255.255
Using PFS Group: 2
Save Password: Allowed
Current EzVPN Peer:10.0.0.110
```

Note that on a Cisco IOS EasyVPN server, PFS must be included in IPSec proposals by adding to the crypto map, as in the following example:

```
crypto dynamic-map mode 1
 set security-association lifetime seconds 180
 set transform-set client
 set pfs group2
 set isakmp-profile fred
 reverse-route
```

Dial Backup: Examples

Static IP Addressing

The following example shows that static IP addressing has been configured for a Cisco 1711 router:

```
Router# show running-config

Building configuration...

Current configuration : 3427 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname ph4_R5
```

```
!
boot-start-marker
boot-end-marker
!
no logging buffered
!
username ph4_R8 password 0 cisco
username ph4_R7 password 0 lab
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
aaa new-model
!
!
aaa session-id common
ip subnet-zero
!
!
no ip domain lookup
ip cef
ip ids po max-events 100
ip dhcp-client default-router distance 1
no ftp-server write-enable
!
!
track 123 rtr 3 reachability
!
crypto isakmp keepalive 10 periodic
!
!
crypto ipsec client ezvpn backup_profile_vpn3k
 connect auto
 group hw_client_groupname key password123
 mode client
 peer 10.0.0.5
 username rchu password  password123
crypto ipsec client ezvpn hw_client_vpn3k
 connect auto
 group hw_client_groupname key password123
 backup backup_profile_vpn3k track 123
 mode client
 peer 10.0.0.5
 username rchu password password123
!
!
interface Loopback0
 ip address 10.40.40.50 255.255.255.255
!
interface Loopback1
 ip address 10.40.40.51 255.255.255.255
!
interface Loopback2
 no ip address
!
interface FastEthernet0
 description Primary Link to 10.0.0.2
 ip address 10.0.0.10 255.255.255.0
 duplex auto
 speed auto
 no cdp enable
 crypto ipsec client ezvpn hw_client_vpn3k
!
interface FastEthernet1
```

```
 no ip address
 duplex full
 speed 100
 no cdp enable
!
interface FastEthernet2
 no ip address
 no cdp enable
!
interface FastEthernet3
 no ip address
 no cdp enable
!
interface FastEthernet4
 no ip address
 no cdp enable
!
interface Vlan1
 ip address 10.0.0.1 255.255.255.0
 crypto ipsec client ezvpn backup_profile_vpn3k inside
 crypto ipsec client ezvpn hw_client_vpn3k inside
!
interface Async1
 description Backup Link
 no ip address
 ip nat outside
 ip virtual-reassembly
 encapsulation ppp
 no ip route-cache cef
 dialer in-band
 dialer pool-member 1
 dialer-group 1
 async default routing
 async mode dedicated
!
interface Dialer1
 ip address 10.30.0.1 255.255.255.0
 encapsulation ppp
 no ip route-cache cef
 dialer pool 1
 dialer idle-timeout 60
 dialer string 102
 dialer hold-queue 100
 dialer-group 1
 crypto ipsec client ezvpn backup_profile_vpn3k
!
ip local policy route-map policy_for_rtr
ip classless

ip route 0.0.0.0 0.0.0.0 faste0 track 123


ip route 0.0.0.0 0.0.0.0 Dialer1 240
no ip http server
no ip http secure-server
!
!
ip access-list extended dummy1
 permit ip host 10.0.0.2 host 30.0.0.1
ip access-list extended important_traffic
 permit ip 10.0.0.0 0.0.0.255 20.0.0.0 0.0.0.255
 permit ip 10.0.0.0 0.0.0.255 30.0.0.0 0.0.0.255
ip access-list extended important_traffic_2
 permit ip 10.0.0.0 0.0.0.255 30.0.0.0 0.0.0.255
```

```
access-list 112 permit icmp any host 10.0.10.2 echo
dialer-list 1 protocol ip permit
no cdp run
!
route-map policy_for_rtr permit 10
 match ip address 112
 set interface Null0
 set ip next-hop 1.0.10.2
!
!
control-plane
!
rtr 2
 type echo protocol ipIcmpEcho 10.0.0.2 source-ipaddr 10.0.0.3
 timeout 10000
 threshold 1000
 frequency 11
rtr schedule 2 life forever start-time now
rtr 3
 type echo protocol ipIcmpEcho 10.0.0.2 source-interface FastEthernet0
 timeout 10000
 threshold 1000
 frequency 11
rtr schedule 3 life forever start-time now
!
line con 0
 exec-timeout 0 0
line 1
 modem InOut
 modem autoconfigure discovery
 transport input all
 autoselect ppp
 stopbits 1
 speed 115200
 flowcontrol hardware
line aux 0
line vty 0 4
 password lab
!
```

DHCP Configured on Primary Interface and PPP Async As Backup

The following example shows that a Cisco 1711 router has been configured so that DHCP is configured on the primary interface and PPP asynchronous mode is configured as the backup:

```
Router# show running-config

Building configuration...

Current configuration : 3427 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname ph4_R5
!
boot-start-marker
boot-end-marker
!
no logging buffered
!
username ph4_R8 password 0 cisco
```

```
username ph4_R7 password 0 lab
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
aaa new-model
!
!
aaa session-id common
ip subnet-zero
!
!
no ip domain lookup
ip cef
ip ids po max-events 100
ip dhcp-client default-router distance 1
no ftp-server write-enable
!
!
track 123 rtr 3 reachability
!
crypto isakmp keepalive 10 periodic
!
!
crypto ipsec client ezvpn backup_profile_vpn3k
 connect auto
 group hw_client_groupname key password123
 mode client
 peer 10.0.0.5
 username rchu password  password123
crypto ipsec client ezvpn hw_client_vpn3k
 connect auto
 group hw_client_groupname key password123
 backup backup_profile_vpn3k track 123
 mode client
 peer 10.0.0.5
 username rchu password  password123
!
!
interface Loopback0
 ip address 10.40.40.50 255.255.255.255
!
interface Loopback1
 ip address 10.40.40.51 255.255.255.255
!
interface Loopback2
 no ip address
!
interface FastEthernet0
 description Primary Link to 10.0.0.2
 ip dhcp client route track 123
 ip address dhcp
 duplex auto
 speed auto
 no cdp enable
 crypto ipsec client ezvpn hw_client_vpn3k
!
interface FastEthernet1
 no ip address
 duplex full
 speed 100
 no cdp enable
!
interface FastEthernet2
```

```
 no ip address
 no cdp enable
!
interface FastEthernet3
 no ip address
 no cdp enable
!
interface FastEthernet4
 no ip address
 no cdp enable
!
interface Vlan1
 ip address 10.0.0.1 255.255.255.0
 crypto ipsec client ezvpn backup_profile_vpn3k inside
 crypto ipsec client ezvpn hw_client_vpn3k inside
!
interface Async1
 description Backup Link
 no ip address
 ip nat outside
 ip virtual-reassembly
 encapsulation ppp
 no ip route-cache cef
 dialer in-band
 dialer pool-member 1
 dialer-group 1
 async default routing
 async mode dedicated
!
interface Dialer1
 ip address 10.0.0.3 255.255.255.0
 encapsulation ppp
 no ip route-cache cef
 dialer pool 1
 dialer idle-timeout 60
 dialer string 102
 dialer hold-queue 100
 dialer-group 1
 crypto ipsec client ezvpn backup_profile_vpn3k
!
ip local policy route-map policy_for_rtr
ip classless
ip route 0.0.0.0 0.0.0.0 Dialer1 240
no ip http server
no ip http secure-server
!
!
ip access-list extended dummy1
 permit ip host 10.10.0.2 host 30.0.0.1
ip access-list extended important_traffic
 permit ip 10.0.0.0 0.0.0.255 20.0.0.0 0.0.0.255
 permit ip 10.0.0.0 0.0.0.255 30.0.0.0 0.0.0.255
ip access-list extended important_traffic_2
 permit ip 10.0.0.0 0.0.0.255 30.0.0.0 0.0.0.255
access-list 112 permit icmp any host 1.0.0.2 echo
dialer-list 1 protocol ip permit
no cdp run
!
route-map policy_for_rtr permit 10
 match ip address 112
 set interface Null0
 set ip next-hop 10.0.0.2
!
!
```

```
control-plane
!
rtr 2
 type echo protocol ipIcmpEcho 10.0.0.2 source-ipaddr 10.0.0.3
 timeout 10000
 threshold 1000
 frequency 11
rtr schedule 2 life forever start-time now
rtr 3
 type echo protocol ipIcmpEcho 10.0.0.2 source-interface FastEthernet0
 timeout 10000
 threshold 1000
 frequency 11
rtr schedule 3 life forever start-time now
!
line con 0
 exec-timeout 0 0
line 1
 modem InOut
 modem autoconfigure discovery
 transport input all
 autoselect ppp
 stopbits 1
 speed 115200
 flowcontrol hardware
line aux 0
line vty 0 4
 password lab
!
```

Web-Based Activation: Example

The following example shows that HTTP connections from the user are to be intercepted and that the user can do web-based authentication (192.0.0.13 is the VPN client device and 192.0.0.1 is the server device):

```
crypto ipsec client ezvpn tunnel22
  connect manual
  group tunnel22 key 22tunnel
  mode client
  peer 192.0.0.1
  xauth userid mode http-intercept
!
!
interface Ethernet0
  ip address 10.4.23.15 255.0.0.0
  crypto ipsec client ezvpn tunnel22 inside!
interface Ethernet1
  ip address 192.0.0.13 255.255.255.128
  duplex auto
  crypto ipsec client ezvpn tunnel22
!
```

Easy VPN Server Configuration Examples

This section describes basic Cisco Easy VPN server configurations that support the Cisco Easy VPN remote configurations given in the previous sections. For complete information on configuring these servers, see *Easy VPN Server* for Cisco IOS Release 12.3(7)T, available on Cisco.com.

- Cisco Easy VPN Server Without Split Tunneling: Example, page 1015
- Cisco Easy VPN Server Configuration with Split Tunneling: Example, page 1016
- Cisco Easy VPN Server Configuration with Xauth: Example, page 1018
- Easy VPN Server Interoperability Support: Example, page 1020

Cisco Easy VPN Server Without Split Tunneling: Example

The following example shows the Cisco Easy VPN server that is the destination peer router for the Cisco Easy VPN remote network extension mode configurations shown earlier in this section. In addition to the other IPSec configuration commands, the **crypto isakmp client configuration group** command defines the attributes for the VPN group that was assigned to the Easy VPN remote router. This includes a matching key value (easy vpn remote password), and the appropriate routing parameters, such as DNS server, for the Easy VPN remotes.

To support the network extension mode of operation, the **ip route** command instructs that incoming packets for the 172.168.0.0 network be directed from the cable modem interface to the Cisco Easy VPN remote. Other **ip route** commands might be needed, depending on the topology of your network.

Note This example shows a Cisco uBR925 cable access router, but typically the destination Easy VPN remote is a router, such as a Cisco VPN 3000 concentrator or a Cisco IOS router, that supports the Easy VPN Server feature.

```
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
!
hostname uBR925Server
!
aaa new-model
!
!
aaa authorization network easy vpn remote-groupname local
aaa session-id common
!
!
clock timezone - 0 6
ip subnet-zero
!
ip ssh time-out 120
ip ssh authentication-retries 3
!
crypto isakmp policy 1
 authentication pre-share
 group 2
crypto isakmp client configuration address-pool local dynpool
!
```

```
crypto isakmp client configuration group easy vpn remote-groupname
 key easy vpn remote-password
 dns 172.168.0.250 172.168.0.251
 wins 172.168.0.252 172.168.0.253
 domain cisco.com
 pool dynpool
!
!
crypto ipsec transform-set transform-1 esp-des esp-sha-hmac
!
crypto dynamic-map dynmap 1
 set transform-set transform-1
 reverse-route
!
!
crypto map dynmap isakmp authorization list easy vpn remote-groupname
crypto map dynmap client configuration address respond
crypto map dynmap 1 ipsec-isakmp dynamic dynmap
!
!
interface Ethernet0
 ip address 172.168.0.129 255.255.255.128
!
interface cable-modem0
 no cable-modem compliant bridge
 crypto map dynmap
!
interface usb0
 no ip address
 arp timeout 0
!
ip local pool dynpool 172.168.0.65 172.168.0.127
ip classless
! Add the appropriate ip route commands for network-extension mode
ip route 172.168.1.0 255.255.255.248 cable-modem0
no ip http server
no ip http cable-monitor
!
snmp-server manager
!
line con 0
 exec-timeout 0 0
line vty 0 4
!
scheduler max-task-time 5000
```

Cisco Easy VPN Server Configuration with Split Tunneling: Example

The following example shows a Cisco Easy VPN server that is configured for a split tunneling configuration with a Cisco Easy VPN remote. This example is identical to that shown in the "Cisco Easy VPN Server Without Split Tunneling: Example" except for access list 150, which is assigned as part of the **crypto isakmp client configuration group** command. This access list allows the Cisco Easy VPN remote to use the server to access one additional subnet that is not part of the VPN tunnel without compromising the security of the IPSec connection.

To support network extension mode, the **ip route** command instructs that incoming packets for the 172.168.0.0 network be directed from the cable modem interface to the Cisco Easy VPN remote. Other **ip route** commands might be necessary, depending on the topology of your network.

Note This example shows a Cisco uBR925 cable access router, but typically the destination Easy VPN remote will be a router, such as a VPN 3000 concentrator or a Cisco IOS router, that supports the Easy VPN Server feature.

```
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
!
hostname uBR925Server
!
aaa new-model
!
!
aaa authorization network easy vpn remote-groupname local
aaa session-id common
!
!
clock timezone - 0 6
ip subnet-zero
!
ip ssh time-out 120
ip ssh authentication-retries 3
!
crypto isakmp policy 1
 authentication pre-share
 group 2
crypto isakmp client configuration address-pool local dynpool
!
crypto isakmp client configuration group easy vpn remote-groupname
 key easy vpn remote-password
 dns 172.168.0.250 172.168.0.251
 wins 172.168.0.252 172.168.0.253
 domain cisco.com
 pool dynpool
 acl 150
!
!
crypto ipsec transform-set transform-1 esp-des esp-sha-hmac
!
crypto dynamic-map dynmap 1
 set transform-set transform-1
 reverse-route
!
!
crypto map dynmap isakmp authorization list easy vpn remote-groupname
crypto map dynmap client configuration address respond
crypto map dynmap 1 ipsec-isakmp dynamic dynmap
!
!
interface Ethernet0
 ip address 172.168.0.129 255.255.255.128
!
interface cable-modem0
 no cable-modem compliant bridge
 crypto map dynmap
!
interface usb0
 no ip address
```

```
 arp timeout 0
!
ip local pool dynpool 172.168.0.65 172.168.0.127
ip classless
! Add the appropriate ip route commands for network-extension mode
ip route 172.168.1.0 255.255.255.248 cable-modem0
no ip http server
no ip http cable-monitor
!
access-list 150 permit ip 172.168.0.128 0.0.0.127 any
snmp-server manager
!
line con 0
 exec-timeout 0 0
line vty 0 4
!
scheduler max-task-time 5000
end
```

Cisco Easy VPN Server Configuration with Xauth: Example

The following example shows a Cisco Easy VPN server configured to support Xauth with the Cisco Easy VPN Remote feature. This example is identical to that shown in the "Cisco Easy VPN Server Configuration with Split Tunneling: Example" except for the following commands that enable and configure Xauth:

- **aaa authentication login userlist local**—Specifies the local username database for authentication at login time. You could also specify the use of RADIUS servers by first using the **aaa authentication login userlist group radius** command and then by specifying the RADIUS servers using the **aaa group server radius** command.
- **crypto isakmp xauth timeout**—Specifies the amount of time, in seconds, that the user has to enter the appropriate username and password to authenticate the session.
- **crypto map dynmap client authentication list userlist**—Creates a crypto map named "dynmap" that enables Xauth.
- **username cisco password 7 cisco**—Creates an entry in the local username database for a user with the username of "cisco" and an encrypted password of "cisco." This command should be repeated for each separate user that accesses the server.

The following commands, which are also present in the non-Xauth configurations, are also required for Xauth use:

- **aaa authorization network easy vpn remote-groupname local**—Requires authorization for all network-related service requests for users in the group named "easy vpn remote-groupname" using the local username database.
- **aaa new-model**—Specifies that the router should use the new AAA authentication commands.
- **aaa session-id common**—Specifies that a unique and common session ID should be used for AAA sessions.
- **crypto map dynmap 1 ipsec-isakmp dynamic dynmap**—Specifies that IKE should be used to establish the IPSec SAs, using the crypt map named "dynmap" as the policy template.
- **crypto map dynmap client configuration address respond**—Enables IKE negotiation, accepting requests from any requesting peers.
- **crypto map dynmap isakmp authorization list easy vpn remote-groupname**—Configures the crypto map named "dynmap" to use IKE Shared Secret using the group named "easy vpn remote-groupname."

Tip This configuration shows the server configured for split tunneling, but Xauth can also be used with nonsplit tunnel configurations as well.

Note This example shows a Cisco uBR925 cable access router, but typically the destination Easy VPN server is a router such as a VPN 3000 concentrator or a Cisco IOS router that supports the Easy VPN Server feature.

```
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
!
hostname uBR925Server
!
aaa new-model
!
!
aaa authentication login userlist local
aaa authorization network easy vpn remote-groupname local
aaa session-id common
!
username cisco password 7 cisco
!
!
clock timezone - 0 6
ip subnet-zero
!
ip ssh time-out 120
ip ssh authentication-retries 3
!
crypto isakmp policy 1
 authentication pre-share
 group 2
crypto isakmp client configuration address-pool local dynpool
crypto isakmp xauth timeout 60
!
crypto isakmp client configuration group easy vpn remote-groupname
 key easy vpn remote-password
 dns 172.168.0.250 172.168.0.251
 wins 172.168.0.252 172.168.0.253
 domain cisco.com
 pool dynpool
 acl 150
!
!
crypto ipsec transform-set transform-1 esp-des esp-sha-hmac
!
crypto dynamic-map dynmap 1
 set transform-set transform-1
 reverse-route
!
!
crypto map dynmap client authentication list userlist
crypto map dynmap isakmp authorization list easy vpn remote-groupname
crypto map dynmap client configuration address respond
crypto map dynmap 1 ipsec-isakmp dynamic dynmap
```

```
!!
!
interface Ethernet0
 ip address 172.168.0.129 255.255.255.128
!
interface cable-modem0
 no cable-modem compliant bridge
 crypto map dynmap
!
interface usb0
 no ip address
 arp timeout 0
!
ip local pool dynpool 172.168.0.65 172.168.0.127
ip classless
ip route 172.168.1.0 255.255.255.248 cable-modem0
no ip http server
no ip http cable-monitor
!
access-list 150 permit ip 172.168.0.128 0.0.0.127 any
snmp-server manager
!
line con 0
 exec-timeout 0 0
line vty 0 4
!
scheduler max-task-time 5000
end
```

Easy VPN Server Interoperability Support: Example

For information about this feature, see “General information on IPSec and VPN” in the section “Additional References” (*Managing VPN Remote Access*).

Additional References

The following sections provide references related to Cisco Easy VPN Remote.

Related Documents

Related Topic	Document Title
Platform-specific documentation	
Cisco 800 series routers	• *Cisco 800 Series Routers* • *Cisco 806 Router Hardware Installation Guide* • *Cisco 806 Router and SOHO 71 Router Hardware Installation Guide* • *Cisco 806 Software Configuration Guide* • *Cisco 826 Router Hardware Installation Guide* • *Cisco 827 Router Hardware Installation Guide* • *Cisco 827 and SOHO 77 Routers Hardware Installation Guide* • *Cisco 828 and SOHO 78 Routers Hardware Installation Guide* • *Cisco 827 Router Software Configuration Guide* • *Cisco 828 Router and SOHO 78 Router Software Configuration Guide* • *Cisco 837 ADSL Broadband Router*
Cisco uBR905 and Cisco uBR925 cable access routers	• *Cisco uBR925 Cable Access Router Hardware Installation Guide* • *Cisco uBR905 Hardware Installation Guide* • *Cisco uBR905/uBR925 Cable Access Router Software Configuration Guide* • *Cisco uBR905 Cable Access Router Subscriber Setup Quick Start* Card • *Cisco uBR925 Cable Access Router Subscriber Setup Quick Start* Card • *Cisco uBR925 Cable Access Router Quick Start User Guide*

Related Topic	Document Title
Cisco 1700 series routers	• *Cisco 1700 Series Router Software Configuration Guide* • *Cisco 1710 Security Router Hardware Installation Guide* • *Cisco 1710 Security Router Software Configuration Guide* • *Cisco 1711 Security Access Router* • *Cisco 1720 Series Router Hardware Installation Guide* • *Cisco 1721 Access Router Hardware Installation Guide* • *Cisco 1750 Series Router Hardware Installation Guide* • *Cisco 1751 Router Hardware Installation Guide* • *Cisco 1751 Router Software Configuration Guide* • *Cisco 1760 Modular Access Router Hardware Installation Guide* Also see the Cisco IOS release notes for Cisco IOS Release 12.2(4)YA: • *SOHO 70 and Cisco 800 Series—Release Notes for Release 12.2(4)YA* • *Release Notes for Cisco uBR905 and Cisco uBR925 Cable Access Routers for Cisco IOS Release 12.2 YA* • *Cisco 1700 Series—Release Notes for Release 12.2(4)YA*
Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers	• *Cisco 2600 Series Multiservice Platforms* • *Cisco 2600 Series Routers Hardware Installation Guide* • *Cisco 3600 Series Multiservice Platforms* • *Cisco 3600 Series Hardware Installation Guide* • *Cisco 3700 Series Multiservice Access Routers* • *Cisco 3700 Series Routers Hardware Installation Guide* • *Software Configuration Guide for Cisco 2600 Series, Cisco 3600 Series, and Cisco 3700 Series Routers* • Cisco 2600 Series, 3600 Series, and 3700 Series Regulatory Compliance and Safety Information on Cisco.com
Access control lists, configuring	• "Access Control Lists: Overview and Guidelines" chapter of the "Traffic Filtering and Firewalls" section of the *Cisco IOS Security Configuration Guide*, Release 12.3
IPSec and VPN documentation	
Easy VPN Server feature, which provides Cisco Unity client support for the Cisco Easy VPN Remote feature	• *Easy VPN Server*

Related Topic	Document Title
General information on IPSec and VPN	• *Deploying IPsec*—Provides an overview of IPSec encryption and its key concepts, along with sample configurations. Also provides a link to many other documents on related topics. • *Certificate Authority Support for IPsec Overview*—Describes the concept of digital certificates and how they are used to authenticate IPSec users. • "Configuring Authentication Proxy" chapter of the *Cisco IOS Security Configuration Guide*, Release 12.3 • *An Introduction to IP Security (IPsec) Encryption*—Provides a step-by-step description of how to configure IPSec encryption. • *Configuring Cisco VPN Client and Easy VPN Server with Xauth*—Describes how to configure a host-to-router Easy VPN solution based on the Cisco VPN client and Cisco IOS remote access server. • *Configure Cisco VPN Client-Easy VPN Server, Xauth, Split Tunnel*—Describes how to configure a host-to-router Easy VPN solution using a Cisco VPN client and Easy VPN server. • *Managing VPN Remote Access*—Describes how to configure the Cisco PIX firewall as an Easy VPN server and how to configure Easy VPN remote software clients. • *Configuring VPN Settings*—Provides information about configuring a PIX firewall to operate as a Cisco Secure VPN client. • IP technical tips sections on Cisco.com.
Encrypted Preshared Key feature	• *Encrypted Preshared Key* feature guide, Release 12.3(2)T
Digital certificates (RSA signature support)	• *Easy VPN Remote RSA Signature Support* feature guide, Release 12.3(7)T1
Dead peer detection	• *IPSec Dead Peer Detection Periodic Message Option* feature guide, Release 12.3(7)T
Object tracking	• *Reliable Static Routing Backup Using Object Tracking* feature guide, Release 12.3(8)T
DNS, configuring	• *Configuring DNS* and *Configuring DNS on Cisco Routers*
DHCP, configuring	• "Configuring DHCP" in the *Cisco IOS IP Configuration Guide*, Release 12.3
802.1x authentication	• *Cisco IOS Easy VPN Remote with 802.1x Authentication* (white paper) • *VPN Access Control Using 802.1X Local Authentication*
Easy VPN remote and server on the same interface	• *Easy VPN Remote and Server on the Same Interface* (white paper)
Easy VPN remote and site to site on the same interface	• *Easy VPN Remote and Site to Site on the Same Interface* (white paper)

Related Topic	Document Title
Additional in-depth configuration information (available on Cisco.com)	• *IPSec VPN* (under the "Technical Documents" and "Cisco IOS IPSec Configuration Documents" sections)—Provides examples for configuring Cisco IOS Easy VPN in client mode and in network extension mode. • *Cisco Easy VPN Solutions*—Provides white papers and examples for configuring Cisco IOS Easy VPN in network extension mode. • *Cisco IOS Security Configuration Guide*, Cisco IOS Release 12.3—Provides an overview of Cisco IOS security features. • *Cisco IOS Security Command Reference*, Cisco IOS Release 12.3 T—Provides a reference for each of the Cisco IOS commands used to configure IPsec encryption and related security features. • *Cisco IOS Master Commands List*, Release 12.3—Lists the Cisco IOS commands used to configure all Release 12.3 security features.

Note Additional documentation on IPSec becomes available on Cisco.com as new features and platforms are added. Cisco Press also publishes several books on IPSec—go to http://www.ciscopress.com for more information on Cisco Press books.

Standards

Standards	Title
No new or modified standards are supported by this feature.	—

MIBs

MIBs	MIBs Link
• CISCO-IPSEC-FLOW-MONITOR-MIB—Contains attributes describing IPSec-based VPNs (Internet Engineering Task Force (IETF) IPSec Working Group Draft). • CISCO-IPSEC-MIB—Describes Cisco implementation-specific attributes for Cisco routers implementing IPSec VPNs. • CISCO-IPSEC-POLICY-MAP-MIB—Extends the CISCO-IPSEC-FLOW-MONITOR-MIB to map dynamically instantiated structures to the policies, transforms, cryptomaps, and other structures that created or are using them.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

New Commands

- **clear crypto ipsec client ezvpn**
- **crypto ipsec client ezvpn (global)**
- **crypto ipsec client ezvpn (interface)**
- **crypto ipsec client ezvpn connect**
- **crypto ipsec client ezvpn xauth**
- **debug crypto ipsec client ezvpn**
- **debug ip auth-proxy ezvpn**
- **ip http ezvpn**
- **show crypto ipsec client ezvpn**
- **xauth userid mode**

Modified Commands

- **show tech-support**
- **type echo**

Appendix A: Supported Mode Configuration Attributes

Mode configuration attributes that are supported by the Easy VPN Remote feature are as follows:

- INTERNAL_IPV4_ADDRESS
- INTERNAL_IPV4_NETMASK
- INTERNAL_IPV4_DNS
- INTERNAL_IPV4_WINS
- MODECFG_BACKUPSERVER
- MODECFG_DEFDOMAIN
- MODECFG_PFS
- MODECFG_SPLIT_INCLUDE

Glossary

AAA—authentication, authorization, and accounting. Framework of security services that provide the method for identifying users (authentication); for remote access control (authorization); and for collecting and sending security server information used for billing, auditing, and reporting (accounting).

aggressive mode—Mode that eliminates several steps during Internet Key Exchange (IKE) authentication negotiation between two or more IPSec peers. Aggressive mode is faster than main mode but is not as secure.

authorization—Method for remote access control, including one-time authorization or authorization for each service; per-user account list and profile; user group support; and support of IP, IPX, ARA, and Telnet. AAA authorization works by assembling a set of attributes that describe what the user is authorized to perform. These attributes are compared to the information contained in a database for a given user and the result is returned to AAA to determine the actual capabilities and restrictions of the user. The database can be located locally on the access server or router, or it can be hosted remotely on a RADIUS or TACACS+ security server. Remote security servers, such as RADIUS and TACACS+, authorize users for specific rights by associating attribute-value (AV) pairs, which define those rights, with the appropriate user. All authorization methods must be defined through AAA.

CA—certificate authority. An entity in a network that issues and manages security credentials and public keys (in the form of X509v3 certificates) for message encryption. As part of a public key infrastructure (PKI), a CA checks with a registration authority (RA) to verify information provided by the requestor of a digital certificate. If the RA verifies the information of the requestor, the CA can then issue a certificate. Certificates generally include the public key of the owner, the expiration date of the certificate, the name of the owner, and other information about the public key owner.

CRWS—Cisco Router Web Setup Tool. Tool that provides web interface capabilities.

DPD—dead peer detection. Queries the liveliness of the Internet Key Exchange (IKE) peer of a router at regular intervals.

DSLAM—digital subscriber line access multiplexer. A device that connects many digital subscriber lines to a network by multiplexing the DSL traffic onto one or more network trunk lines.

IKE—Internet Key Exchange. Key management protocol standard that is used in conjunction with the IP Security (IPSec) standard. IPSec is an IP security feature that provides robust authentication and encryption of IP packets. IPSec can be configured without IKE, but IKE enhances IPSec by providing additional features, flexibility, and ease of configuration for the IPSec standard. IKE is a hybrid protocol that implements the Oakley key exchange and Skeme key exchange inside the Internet Security Association and Key Management Protocol (ISAKMP) framework. ISAKMP, Oakley, and Skeme are security protocols implemented by IKE.

IPSec—IP Security Protocol. Framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer. IPSec uses IKE to handle negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPSec. IPSec can be used to protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

main mode—Mode that ensures the highest level of security when two or more IPSec peers are negotiating IKE authentication. It requires more processing time than aggressive mode.

MIB—Management Information Base. Database of network management information that is used and maintained by a network management protocol, such as Simple Network Management Protocol (SNMP) or Common Management Information Protocol (CMIP). The value of a MIB object can be changed or

retrieved using SNMP or CMIP commands, usually through a graphical user interface (GUI) network management system (NMS). MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches.

peer—Router or device that participates as an endpoint in IPSec and IKE.

preshared key—Shared, secret key that uses IKE for authentication.

QoS—quality of service. Capability of a network to provide better service to selected network traffic over various technologies, including Frame Relay; Asynchronous Transfer Mode (ATM); Ethernet; and 802.1 networks, SONET, and IP-routed networks that may use any or all of these underlying technologies.

RADIUS—Remote Authentication Dial-In User Service. Distributed client or server system that secures networks against unauthorized access. RADIUS clients run on Cisco routers and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.

SA—security association. Instance of security policy and keying material applied to a data flow. Both IKE and IPSec use SAs, although SAs are independent of one another. IPSec SAs are unidirectional, and they are unique in each security protocol. An IKE SA is used by IKE only, and unlike the IPSec SA, it is bi-directional. IKE negotiates and establishes SAs on behalf of IPSec. A user can also establish IPSec SAs manually.

A set of SAs are needed for a protected data pipe, one per direction per protocol. For example, if you have a pipe that supports encapsulating security payload (ESP) between peers, one ESP SA is required for each direction. SAs are uniquely identified by destination (IPSec endpoint) address, security protocol (AH or ESP), and security parameter index (SPI).

SDM—Security Device Manager. Web interface manager that enables you to connect or disconnect a VPN tunnel and that provides a web interface for extended authentication (Xauth).

SNMP—Simple Network Management Protocol. Application-layer protocol that provides a message format for communication between SNMP managers and agents.

trap—Message sent by an SNMP agent to a network management system, console, or terminal to indicate the occurrence of a significant event, such as a specifically defined condition or a threshold that was reached.

VPN—virtual private network. Enables IP traffic to travel securely over a public TCP/IP network by encrypting all traffic from one network to another. A VPN uses tunnels to encrypt all information at the IP level.

Note See *Internetworking Terms and Acronyms* for terms not included in this glossary.

Crypto Access Check on Clear-Text Packets

The Crypto Access Check on Clear-Text Packets feature removes the checking of clear-text packets that go through the IP Security (IPSec) tunnel just prior to encryption or just after decryption. The clear-text packets were checked against the outside physical interface access control lists (ACLs). This checking was often referred to as a double ACL check. This feature enables easier configuration of ACLs and eliminates the security risks that are associated with a double check when using dynamic crypto maps.

Feature History for Crypto Access Check on Clear-Text Packets

Release	Modification
12.3(8)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for Crypto Access Check on Clear-Text Packets, page 1029
- Restrictions for Crypto Access Check on Clear-Text Packets, page 1030
- Information About Crypto Access Check on Clear-Text Packets, page 1030
- How to Configure Crypto Map Access ACLs, page 1034
- Configuration Examples for Crypto Access Check on Clear-Text Packets, page 1036
- Additional References, page 1038
- Command Reference, page 1039

Prerequisites for Crypto Access Check on Clear-Text Packets

- You should be familiar with configuring IPSec.
- You should be familiar with ACLs.

Restrictions for Crypto Access Check on Clear-Text Packets

- This feature does not apply to IPSec configurations on the Virtual Private Network (VPN) service module (card) on Cisco Catalyst 6500 series switches and Cisco 7600 series router platforms.
- This feature supports only extended ACLs.

Information About Crypto Access Check on Clear-Text Packets

To use the crypto access check on clear-text packets, you should understand the following concepts:

- Crypto Access Check on Clear-Text Packets Overview, page 1030
- Configuration Changes That Are Required for This Feature, page 1030
- How ACL Access Checking Worked Prior to This Feature, page 1031
- ACL Checking Behavior After Upgrading to This Feature, page 1032
- Backward Compatibility, page 1034

Crypto Access Check on Clear-Text Packets Overview

The Crypto Access Check on Clear-Text Packets feature provides four changes for the interaction between IPSec and interface access lists. The changes are as follows:

- Removes the checking of inbound, just-decrypted clear-text packets against the outside interface inbound ACL.
- Removes the checking of outbound clear-text packets just prior to encryption against the outside interface outbound ACL.
- Adds the checking of outbound encrypted packets against the outside interface outbound ACL.
- Adds the capability to configure ACLs under the crypto map to check inbound clear-text packets after decryption or outbound clear-text packets prior to encryption.

This feature enables the easier and more consistent configuration of ACLs that control packet movement in and out of the outside interface as well as in and out of the IPSec encryption tunnel. This feature also eliminates security risks that are associated with the current double check when using dynamic crypto maps.

Configuration Changes That Are Required for This Feature

This feature requires the following configuration changes to be performed. Some are required and some are optional.

Prior to Upgrading

Prior to upgrading to this feature, you should do the following. This change is required.

Check all outside interfaces for outbound ACLs. If any outbound ACLs exist, check to ensure that they include access-list entries (ACEs) that permit outbound Encapsulating Security Payload (ESP) IP protocol 50 packets or Authentication Header (AH) IP protocol 51 packets. The ACL entries will be

needed after the upgrade because the outbound encrypted packets will be checked against the outside interface outbound ACL. If the ESP or AH packets are not allowed by the outside interface outbound ACL, the IPSec VPN tunnels will not forward traffic.

After Upgrading

After upgrading to this feature, you should do the following. The first two procedures are required if you are using dynamic crypto maps. However, these procedures are recommended even if you are not using dynamic crypto maps. The third and fourth procedures are optional.

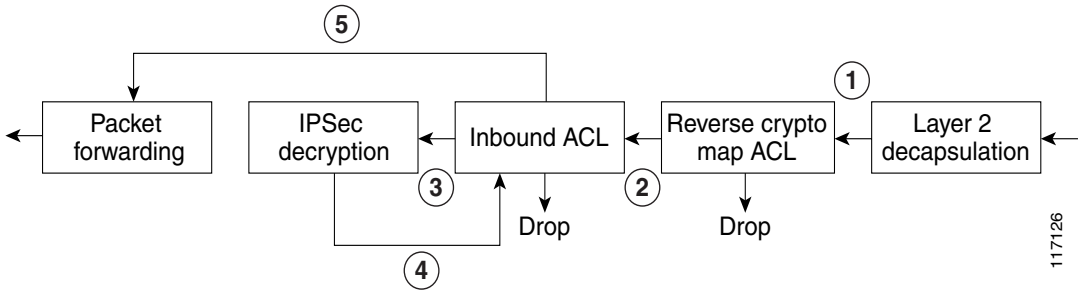
- Check all outside interfaces for inbound ACLs that contain ACEs that permit inbound, just-decrypted clear-text packets. These ACEs need to be removed if dynamic crypto maps are being used because when the IPSec tunnel is not "up," the ACEs will allow the clear-text packets into the network. If dynamic crypto maps are not being used, the ACEs can still be removed to simplify the outside interface ACLs.
- Check all outside interfaces for outbound ACLs that contain ACEs that permit outbound clear-text packets that would be encrypted. These ACEs need to be removed if dynamic crypto maps are being used because when the IPSec tunnel is not up, these ACEs will allow the clear-text packets out of the network. If dynamic crypto maps are not being used, these ACEs can still be removed to simplify the outside interface ACLs.
- Add an outbound crypto map access ACL under the crypto map to deny to-be-encrypted, outbound clear-text packets that should be dropped. Be sure that you also permit all other packets in this ACL.
- Add an inbound crypto map access ACL under the crypto map to deny just-decrypted, inbound clear-text packets that should be dropped. Be sure to also permit all other packets in this ACL.

The last two configuration changes are needed only in the rare cases in which the crypto map ACL (that selects packets to be encrypted) is more general than the packet flows that you want to encrypt. Adding outbound or inbound crypto map ACLs is usually done to keep the crypto map ACL small and simple, which saves CPU utilization and memory. The **set ip access-group** command, which is used to cause the checking of clear-text packets after decryption and before encryption, can be used under the crypto map to accomplish this task independent of the outside interface ACLs.

How ACL Access Checking Worked Prior to This Feature

Prior to Cisco Release 12.3(8)T, there was a double ACL check on the inbound packets, once on the encrypted packet and then again on the just-decrypted clear-text packet. The process on the inbound path is shown in Figure 68.

Figure 68 Inbound Encrypted Packet Flow Prior to This Feature

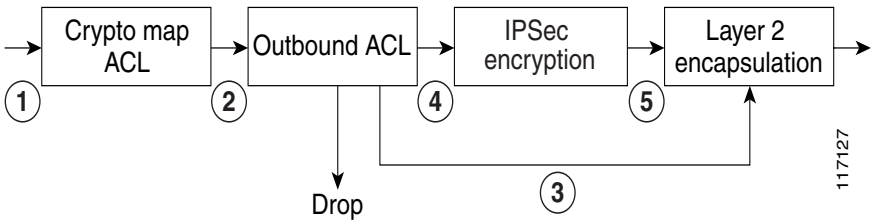


1. Arriving IP packet is checked against the reverse crypto map ACL. If the packet is denied, it is dropped because the incoming packet was not encrypted. The IPSec configuration specifies that it should have been encrypted.
2. IP packet is checked against the interface inbound ACL. If denied, it is dropped.
3. If the IP packet is encrypted, it is then decrypted.
4. Just-decrypted IP packet is again checked against the interface inbound ACL. If denied, it is dropped.
5. Just-decrypted and not-encrypted IP packets that are permitted by the interface inbound ACL are forwarded.

On the outbound path, crypto feature checking for encryption took place before the output feature check for the ACL. The output ACL was run on the clear-text packet before the packet was sent for encryption. After the packet was encrypted, it was not checked against the outside interface outbound ACL again.

The process on the outbound path is shown in Figure 69.

Figure 69 Outbound Encrypted Packet Flow Prior to This Feature

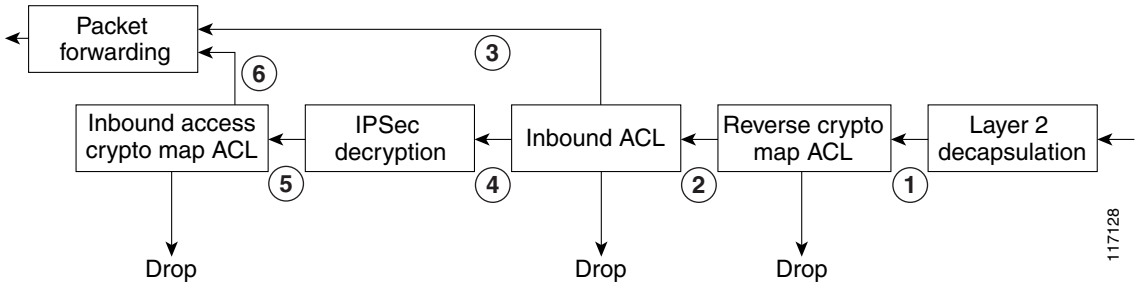


1. Departing IP packet is checked against the crypto map ACL. If permitted, the packet is marked for encryption.
2. All IP packets are checked against the outbound interface ACL. If denied, they are dropped.
3. IP packets not marked for encryption are Layer 2 encapsulated.
4. IP packets marked for encryption are encrypted.
5. Encrypted IP packets are Layer 2 encapsulated.

ACL Checking Behavior After Upgrading to This Feature

Figure 70 illustrates the ACL checking behavior on the inbound path using the Crypto Access Check on Clear-Text Packets feature.

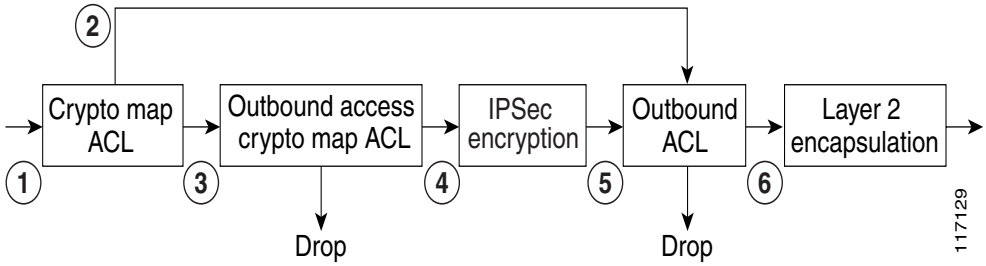
Figure 70 New Inbound Encrypted Packet Flow



1. Arriving IP packet is checked against the reverse crypto map ACL. If it is denied, it is dropped because the incoming packet was not encrypted. The IPSec configuration specifies that it should have been encrypted.
2. IP packet is checked against the interface inbound ACL. If denied, it is dropped.
3. If the IP packet is not encrypted, it is forwarded.
4. If the IP packet is encrypted, it is then decrypted.
5. Just-decrypted IP packet is checked against the inbound access crypto map ACL (optional). If the packet is denied, it is dropped.
6. Just-decrypted IP packet is forwarded.

Figure 71 illustrates the ACL checking behavior on the outbound path using the Crypto Access Check on Clear-Text Packets feature.

Figure 71 New Outbound Encrypted Packet Flow



1. All departing IP packets are checked against the crypto map ACL. If the packets are permitted, they are marked for encryption.
2. IP packets not marked for encryption are checked against the outbound interface ACL. If the packets are denied, they are dropped.
3. IP packets marked for encryption are checked against the outbound access crypto map ACL (optional). If the packets are denied, they are dropped.
4. Permitted IP packets are encrypted.
5. Encrypted IP packets are checked against the outbound interface ACL. If the packets are denied, they are dropped.
6. Permitted IP packets are Layer 2 encapsulated.

Backward Compatibility

If the Cisco IOS software is subsequently downgraded to a release that does not have the Crypto Access Check on Clear-Text Packets feature, the just-decrypted and to-be-encrypted clear-text packets will again be blocked by the outside interface ACLs. Therefore, if you have removed lines from the interface ACLs, you should undo the changes that were made to the ACLs if you are downgrading to an earlier version.

How to Configure Crypto Map Access ACLs

This section contains the following procedures:

- Adding or Removing ACLs, page 1034 (optional)
- Verifying the Configured ACLs, page 1035 (optional)

Adding or Removing ACLs

To add or remove crypto map access ACLs, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-number*
4. **set ip access-group** {*access-list-number* | *access-list-name*} {**in** | **out**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto map** *map-name* seq-*number* **Example:** `Router(config)# crypto map vpn1 10`	Selects the crypto map and the sequence map entry under the crypto map to which you want to add the crypto map access ACL; also enters crypto map configuration mode.
Step 4	**set ip access-group** {*access-list-number* |*access-list-name*}{**in** | **out**} **Example:** `Router(config-crypto-map)# set ip access-group 151 in`	Allows you to check the postdecrypted or preencrypted packet against an ACL without having to use the outside physical interface ACL.

Verifying the Configured ACLs

The **show ip access-list** command can be used to verify the crypto input or output access-check ACLs that have been configured. Also, the packets that have been dropped in the context of the crypto input access-check ACL in the inbound path will be logged as receive (recv) errors, and packets dropped on the outbound path will be logged as send errors.

The **show crypto map** command can be used to verify crypto map configuration information.

SUMMARY STEPS

1. **enable**
2. **show ip access-list** [*access-list-number* | *access-list-name* | **dynamic**]
3. **show crypto map** [**interface** *interface* | **tag** *map-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip access-list** [*access-list-number* | *access-list-name* | **dynamic**] **Example:** `Router# show ip access-list Internetfilter`	Displays a configured ACL.
Step 3	**show crypto map** [**interface** *interface* | **tag** *map-name*] **Example:** `Router# show crypto map`	Displays the crypto maps that have been configured.

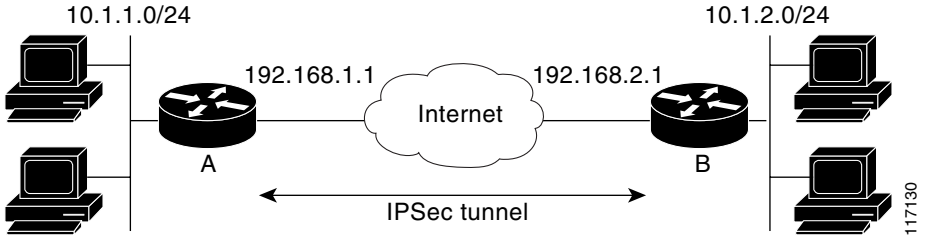
Configuration Examples for Crypto Access Check on Clear-Text Packets

This section contains the output for the following stages of crypto access configuration:

- Previous IPSec ACL Configuration: Example, page 1036
- New IPSec ACL Configuration Without Crypto Access ACLs: Example, page 1037
- New IPSec ACL Configuration with Crypto Access ACLs: Example, page 1037

The network diagram used for the following examples is shown in Figure 72.

Figure 72 Network Diagram for Crypto Access Check Configuration Examples



The configuration examples assume these policy rules:

- Allow only encrypted host traffic between hosts on 10.1.1.0/24 and 10.1.2.0/24.
- No clear-text traffic from the Internet to any host.

Previous IPSec ACL Configuration: Example

The following is a sample configuration using an earlier version of Cisco IOS software (before Release 12.3(8)T). The configuration shows outside interface ACLs with a double check on the inbound packets.

```
crypto map vpnmap 10 ipsec-isakmp
 set peer 192.168.2.1
 set transform-set trans1
 match address 101

interface Ethernet0/0
 ip address 10.1.1.1 255.255.255.0
interface Serial1/0
 ip address 192.168.1.1 255.255.255.0
 ip access-group 150 in
 ip access-group 160 out
 crypto map vpnmap

access-list 101 permit ip 10.1.1.0 0.0.0.255 10.1.2.0 0.0.0.255
access-list 150 permit udp host 192.168.2.1 eq 500 host 192.168.1.1 eq 500
access-list 150 permit esp host 192.168.2.1 host 192.168.1.1
access-list 150 permit ip 10.1.2.0 0.0.0.255 10.1.1.0 0.0.0.255

access-list 160 permit udp host 192.168.1.1 eq 500 host 192.168.2.1 eq 500
access-list 160 permit ip 10.1.1.0 0.0.0.255 10.1.2.0 0.0.0.255
```

New IPSec ACL Configuration Without Crypto Access ACLs: Example

The following is a sample configuration using the current version of Cisco IOS software (Release 12.3(8)T). Before the crypto map access ACL is added, clear-text packets through the IPSec tunnel are not checked against an ACL (other packets are checked against the outside interface ACLs). Note the permitting of ESP packets in the outside interface outbound ACL.

```
crypto map vpnmap 10 ipsec-isakmp
 set peer 192.168.2.1
 set transform-set trans1
 match address 101

interface Ethernet0/0
 ip address 10.1.1.1 255.255.255.0
interface Serial1/0
 ip address 192.168.1.1 255.255.255.0
 ip access-group 150 in
 ip access-group 160 out
 crypto map vpnmap
access-list 101 permit ip 10.1.1.0 0.0.0.255 10.1.2.0 0.0.0.255

access-list 150 permit udp host 192.168.2.1 eq 500 host 192.168.1.1 eq 500
access-list 150 permit esp host 192.168.2.1 host 192.168.1.1

access-list 160 permit udp host 192.168.1.1 eq 500 host 192.168.2.1 eq 500
access-list 160 permit esp host 192.168.1.1 host 192.168.2.1
```

New IPSec ACL Configuration with Crypto Access ACLs: Example

The following is a sample configuration using the current version of Cisco IOS software (Release 12.3(8)T). Before a crypto map access ACL is added, clear-text packets through the IPSec tunnel are checked against the crypto map access ACLs (other packets are checked against the outside interface ACLs).

Note In the following example, all IP packets between the subnets 10.1.1.0/24 and 10.1.2.0/24 are to be encrypted, but the crypto map access ACLs allow only Telnet traffic through the IPSec tunnel.

```
crypto map vpnmap 10 ipsec-isakmp
 set peer 192.168.2.1
 set transform-set trans1
 set ip access-group 151 in
 set ip access-group 161 out
 match address 101

interface Ethernet0/0
 ip address 10.1.1.1 255.255.255.0
interface Serial1/0
 ip address 192.168.1.1 255.255.255.0
 ip access-group 150 in
 ip access-group 160 out
 crypto map vpnmap

access-list 101 permit ip 10.1.1.0 0.0.0.255 10.1.2.0 0.0.0.255

access-list 150 permit udp host 192.168.2.1 eq 500 host 192.168.1.1 eq 500
access-list 150 permit esp host 192.168.2.1 host 192.168.1.1
```

```
access-list 151 permit ip 10.1.2.0 0.0.0.255 eq telnet 10.1.1.0 0.0.0.255
access-list 151 permit ip 10.1.2.0 0.0.0.255 10.1.1.0 0.0.0.255 eq telnet

access-list 160 permit udp host 192.168.1.1 eq 500 host 192.168.2.1 eq 500
access-list 160 permit esp host 192.168.1.1 host 192.168.2.1

access-list 161 permit ip 10.1.1.0 0.0.0.255 10.1.2.0 0.0.0.255 eq telnet
access-list 161 permit ip 10.1.1.0 0.0.0.255 eq telnet 10.1.2.0 0.0.0.255
```

Additional References

The following sections provide references related to the Crypto Access Check on Clear-Text Packets feature.

Related Documents

Related Topic	Document Title
Configuring IPSec	"IP Security and Encryption" section of the *Cisco IOS Security Configuration Guide*, Release 12.3
Configuring ACLs	"Traffic Filtering and Firewall" section of the *Cisco IOS Security Configuration Guide*, Release 12.3
IPSec Commands	*Cisco IOS Security Command Reference*, Release 12.3T

Standards

Standards	Title
There are no new or modified standards associated with this feature.	—

MIBs

MIBs	MIBs Link
There are no new or modified MIBs associated with this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
There are no new or modified RFCs associated with this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

New Command

- **set ip access-group**

Modified Command

- **show crypto map (IPSec)**

DF Bit Override Functionality with IPSec Tunnels

This feature module describes the DF Bit Override Functionality with IPSec Tunnels feature and contains the following sections:

- Feature Overview, page 1041
- Supported Platforms, page 1042
- Supported Standards, MIBs, and RFCs, page 1043
- Prerequisites, page 1043
- Configuration Tasks, page 1043
- Configuration Examples, page 1044
- Command Reference, page 1045

Feature Overview

The DF Bit Override Functionality with IPSec Tunnels feature allows customers to specify whether their router can clear, set, or copy the Don't Fragment (DF) bit from the encapsulated header. A DF bit is a bit within the IP header that determines whether a router is allowed to fragment a packet.

Some customer configurations have hosts that perform the following functions:

- Set the DF bit in packets they send
- Use firewalls that block Internet Control Message Protocol (ICMP) errors from outside the firewall, preventing hosts from learning about the maximum transmission unit (MTU) size outside the firewall
- Use IP Security (IPSec) to encapsulate packets, reducing the available MTU size

Customers whose configurations have hosts that prevent them from learning about their available MTU size can configure their router to clear the DF bit and fragment the packet.

Note In compliance with RFC 2401, this feature can be configured globally or per interface. If both levels are configured, the interface configuration will override the global configuration.

Benefits

The DF Bit Override Functionality with IPSec Tunnels feature allows customers to configure the setting of the DF bit when encapsulating tunnel mode IPSec traffic on a global or per-interface level. Thus, if the DF bit is set to clear, routers can fragment packets regardless of the original DF bit setting.

Restrictions

Performance Impact

Because each packet is reassembled at the process level, a significant performance impact occurs at a high data rate. Two major caveats are as follows:

- The reassemble queue can fill up and force fragments to be dropped.
- The traffic is slower because of the process switching.

DF Bit Setting Requirement

If several interfaces share the same crypto map using the local address feature, these interfaces must share the same DF bit setting.

Feature Availability

This feature is available only for IPSec tunnel mode. (IPSec transport mode is not affected because it does not provide an encapsulating IP header.)

Related Documents

The following documents provide information related to the DF Bit Override Functionality with IPSec Tunnels feature:

- "Configuring IPSec Network Security" chapter, *Cisco IOS Security Configuration Guide*, Release 12.2
- "IPSec Network Security Commands" chapter, *Cisco IOS Security Command Reference*, Release 12.2

Supported Platforms

This feature is supported on the following platforms:

- Cisco 800
- Cisco 827
- Cisco 1600
- Cisco 1600R
- Cisco 1700
- Cisco 2600
- Cisco 3620
- Cisco 3640

- Cisco 3660
- Cisco 4000
- Cisco 4500
- Cisco 5200
- Cisco 5300
- Cisco 5400
- Cisco 6400
- Cisco 7100
- Cisco 7200
- Cisco 7500
- Cisco uBR7200
- Cisco uBR900
- Cisco uBR905
- Cisco uBR910

This feature runs on all platforms that support IPSec.

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

No new or modified MIBS are supported by this feature.

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml.

RFCs

- RFC 2401, *Security Architecture for the Internet Protocol*

Prerequisites

IPSec must be enabled on your router.

Configuration Tasks

See the following section for configuration tasks for the DF-Bit Override Functionality with IPSec Tunnels feature:

- Configuring the DF Bit for the Encapsulating Header in Tunnel Mode

Configuring the DF Bit for the Encapsulating Header in Tunnel Mode

To set the DF bit for the encapsulating header in tunnel mode, use the following command in global configuration mode:

Command	Purpose
`Router(config)# crypto ipsec df-bit [clear | set | copy]`	Sets the DF bit for the encapsulating header in tunnel mode for all interfaces. To set the DF bit for a specified interface, use the **crypto ipsec df-bit** command in interface configuration mode. **Note** DF bit interface configuration settings override all DF bit global configuration settings.

Verifying DF Bit Setting

To verify the current DF Bit settings on your router, use the **show running-config** command in EXEC mode.

Configuration Examples

This section provides the following configuration example:

- DF Bit Setting Configuration Example

DF Bit Setting Configuration Example

In following example, the router is configured to globally clear the setting for the DF bit and copy the DF bit on the interface named Ethernet0. Thus, all interfaces *except* Ethernet0 will allow the router to send packets larger than the available MTU size; Ethernet0 will allow the router to fragment the packet.

```
crypto isakmp policy 1
   hash md5
   authentication pre-share
crypto isakmp key Delaware address 192.168.10.66
crypto isakmp key Key-What-Key address 192.168.11.19
!
!
crypto ipsec transform-set BearMama ah-md5-hmac esp-des

crypto ipsec df-bit clear
!
!
crypto map armadillo 1 ipsec-isakmp
set peer 192.168.10.66
set transform-set BearMama
match address 101
!
crypto map basilisk 1 ipsec-isakmp
set peer 192.168.11.19
```

```
set transform-set BearMama
match address 102

!
!
interface Ethernet0
   ip address 192.168.10.38 255.255.255.0
   ip broadcast-address 0.0.0.0
   media-type 10BaseT
   crypto map armadillo
   crypto ipsec df-bit copy
!
interface Ethernet1
   ip address 192.168.11.75 255.255.255.0
   ip broadcast-address 0.0.0.0
   media-type 10BaseT
   crypto map basilisk
!
interface Serial0
   no ip address
   ip broadcast-address 0.0.0.0
   no ip route-cache
   no ip mroute-cache
```

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **crypto ipsec df-bit (global configuration)**
- **crypto ipsec df-bit (interface configuration)**

Distinguished Name Based Crypto Maps

Feature History

Release	Modification
12.2(4)T	This feature was introduced.

This feature module describes the Distinguished Name Based Crypto Map feature in Cisco IOS Release 12.2(4)T. It includes the following sections:

- Feature Overview, page 1047
- Supported Platforms, page 1048
- Supported Standards, MIBs, and RFCs, page 1049
- Prerequisites, page 1049
- Configuration Tasks, page 1049
- Configuration Examples, page 1051
- Command Reference, page 1052

Feature Overview

The Distinguished Name Based Crypto Maps feature allows you to configure the router to restrict access to selected encrypted interfaces for those peers with specific certificates, especially certificates with particular Distinguished Names (DNs).

Previously, if the router accepted a certificate or a shared secret from the encrypting peer, Cisco IOS did not have a method of preventing the peer from communicating with any encrypted interface other than the restrictions on the IP address of the encrypting peer. This feature allows you to configure which crypto maps are usable to a peer based on the DN that a peer used to authenticate itself, thereby, enabling you to control which encrypted interfaces a peer with a specified DN can access.

Benefits

The Distinguished Name Based Crypto Maps feature allows you to set restrictions in the router configuration that prevent peers with specific certificates—especially certificates with particular DNs—from having access to selected encrypted interfaces.

Restrictions

System Requirements

To configure this feature, your router must support IP Security.

Performance Impact

If you restrict access to a large number of DNs, it is recommended that you specify a few number of crypto maps referring to large identity sections instead of specifying a large number of crypto maps referring to small identity sections.

Related Documents

The following documents provide information related to the Distinguished Name Based Crypto Maps feature:

- *Cisco IOS Security Command Reference*, Release 12.2
- *Cisco IOS Security Configuration Guide*, Release 12.2

Supported Platforms

This feature is supported on the following platforms:

- Cisco 1700 series
- Cisco 2600 series
- Cisco 3620
- Cisco 3640
- Cisco 3660
- Cisco 7100 series
- Cisco 7200 series
- Cisco uBR905 Cable Access Router
- Cisco uBR925 Cable Access Router

Determining Platform Support Through Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Feature Navigator. Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image.

To access Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

None

Prerequisites

Before configuring a DN based crypto map, you must perform the following tasks:

- Create an Internet Key Exchange (IKE) policy at each peer.

  For more information on creating IKE policies, refer to the chapter "Configuring Internet Key Exchange Security Protocol" of the *Cisco IOS Security Configuration Guide*.
- Create crypto map entries for IPSec.

  For more information on creating crypto map entries, refer to the chapter "Configuring IPSec Network Security" of the *Cisco IOS Security Configuration Guide*.

Configuration Tasks

See the following sections for configuration tasks for the Distinguished Name Based Crypto Maps feature. Each task in the list is identified as either required or optional.

- Configuring DN Based Crypto Maps (authenticated by DN) (required)
- Configuring DN Based Crypto Maps (authenticated by hostname) (required)
- Applying Identity to DN Based Crypto Maps (required)
- Verifying DN Based Crypto Maps (optional)

Configuring DN Based Crypto Maps (authenticated by DN)

To configure a DN based crypto map that can be used only by peers that have been authenticated by a DN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto identity name`	Configures the identity of a router with the given list of DNs in the certificate of the router and enters crypto identity configuration mode.
Step 2	`Router(crypto-identity)# dn name=string [,name=string]`	Associates the identity of the router with the DN in the certificate of the router. **Note** The identity of the peer must match the identity in the exchanged certificate.

Configuring DN Based Crypto Maps (authenticated by hostname)

To configure a DN based crypto map that can be used only by peers that have been authenticated by a hostname, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto identity name`	Configures the identity of a router with the given list of DNs in the certificate of the router and enters crypto identity configuration mode.
Step 2	`Router(crypto-identity)# fqdn name`	Associates the identity of the router with the hostname that the peer used to authenticate itself. **Note** The identity of the peer must match the identity in the exchanged certificate.

Applying Identity to DN Based Crypto Maps

To apply the identity (within the crypto map context), use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto map map-name seq-num ipsec-isakmp`	Creates or modifies a crypto map entry and enters the crypto map configuration mode.
Step 2	`Router(config-crypto-map)# identity name`	Applies the identity to the crypto map. When this command is applied, only the hosts that match a configuration listed within the **identity** *name* can use the specified crypto map. **Note** If the **identity** command does not appear within the crypto map, the encrypted connection does not have any restrictions other than the IP address of the encrypting peer.

Verifying DN Based Crypto Maps

To verify that this functionality is properly configured, use the following command in EXEC mode:

Command	Purpose
`Router# show crypto identity`	Displays the configured identities.

Troubleshooting Tips

If an encrypting peer attempts to establish a connection that is blocked by the DN based crypto map configuration, the following error message will be logged:

```
<time>: %CRYPTO-4-IKE_QUICKMODE_BAD_CERT: encrypted connection attempted with a peer
without the configured certificate attributes.
```

Configuration Examples

This section provides the following configuration example:

- DN Based Crypto Map Configuration Example

DN Based Crypto Map Configuration Example

The following example shows how to configure DN based crypto maps that have been authenticated by DN and hostname. Comments are included inline to explain various commands.

```
! DN based crypto maps require you to configure an IKE policy at each peer.
crypto isakmp policy 15
```

```
 encryption 3des
 hash md5
 authentication rsa-sig
 group 2
 lifetime 5000
crypto isakmp policy 20
 authentication pre-share
 lifetime 10000
crypto isakmp key 1234567890 address 171.69.224.33
!
! The following is an IPSec crypto map (part of IPSec configuration). It can be used only
! by peers that have been authenticated by DN and if the certificate belongs to BigBiz.
crypto map map-to-bigbiz 10 ipsec-isakmp
 set peer 172.21.114.196
 set transform-set my-transformset
 match address 124
 identity to-bigbiz
!
crypto identity to-bigbiz
 dn ou=BigBiz
!
!
! This crypto map can be used only by peers that have been authenticated by hostname
! and if the certificate belongs to little.com.
crypto map map-to-little-com 10 ipsec-isakmp
 set peer 172.21.115.119
 set transform-set my-transformset
 match address 125
 identity to-little-com
!
crypto identity to-little-com
 fqdn little.com
!
```

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **crypto identity**
- **dn**
- **fqdn**
- **identity**

Dynamic Multipoint VPN (DMVPN)

The Dynamic Multipoint VPN (DMVPN) feature allows users to better scale large and small IPSec Virtual Private Networks (VPNs) by combining generic routing encapsulation (GRE) tunnels, IPSecurity (IPSec) encryption, and Next Hop Resolution Protocol (NHRP).

Feature Specifications for the Dynamic Multipoint VPN (DMVPN) feature

Feature History	
Release	**Modification**
12.2(13)T	This feature was introduced.
Supported Platforms	
For platforms supported in Cisco IOS Release 12.2(13)T, consult Cisco Feature Navigator.	

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that are supported on specific platforms. To obtain updated information about platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. In the release section, you can compare releases side by side to display both the features unique to each software release and the features that releases have in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Contents

- Prerequisites for Dynamic Multipoint VPN (DMVPN), page 1054
- Restrictions for Dynamic Multipoint VPN (DMVPN), page 1054
- Information About Dynamic Multipoint VPN (DMVPN), page 1054
- How to Configure DMVPN, page 1057
- Configuration Examples for Dynamic Multipoint VPN (DMVPN) Feature, page 1064
- Additional References, page 1068
- Command Reference, page 1070

Prerequisites for Dynamic Multipoint VPN (DMVPN)

- Before an mGRE and IPSec tunnel can be established, you must define an IKE policy by using the **crypto isakmp policy** command.

Restrictions for Dynamic Multipoint VPN (DMVPN)

- If you use the Dynamic Tunnel Creation for Spoke-to-Spoke Tunnels benefit of this feature, you must use Internet Key Exchange (IKE) Certificates or wildcard preshared keys for Internet Security Association Key Management Protocol (ISAKMP) authentication.

  Note It is highly recommended that you *do not use* wildcard preshared keys because the attacker will have access to the VPN if one spoke router is compromised.

- GRE tunnel keepalives (that is, the **keepalive** command under GRE interface) are not supported on multipoint GRE tunnels.

Information About Dynamic Multipoint VPN (DMVPN)

To configure the Dynamic Multipoint VPN (DMVPN) feature, you must understand the following concepts:

- Benefits of Dynamic Multipoint VPN (DMVPN), page 1055
- Feature Design of Dynamic Multipoint VPN (DMVPN), page 1055
- IPSec Profiles, page 1056

Benefits of Dynamic Multipoint VPN (DMVPN)

Hub Router Configuration Reduction

Currently, for each spoke router, there is a separate block of configuration lines on the hub router that define the crypto map characteristics, the crypto access list, and the GRE tunnel interface. This feature allows users to configure a single multipoint GRE tunnel interface, a single IPSec profile, and no crypto access lists on the hub router to handle all spoke routers. Thus, the size of the configuration on the hub router remains constant even if spoke routers are added to the network.

Automatic IPSec Encryption Initiation

GRE has the peer source and destination address configured or resolved with NHRP. Thus, this feature allows IPSec to be immediately triggered for the point-to-point GRE tunneling or when the GRE peer address is resolved via NHRP for the multipoint GRE tunnel.

Support for Dynamically Addressed Spoke Routers

When using point-to-point GRE and IPsec hub-and-spoke VPN networks, the physical interface IP address of the spoke routers must be known when configuring the hub router because IP address must be configured as the GRE tunnel destination address. This feature allows spoke routers to have dynamic physical interface IP addresses (common for cable and DSL connections). When the spoke router comes online, it will send registration packets to the hub router: Within these registration packets is the current physical interface IP address of this spoke.

Dynamic Tunnel Creation for Spoke-to-Spoke Tunnels

This feature eliminates the need for spoke-to-spoke configuration for direct tunnels. When a spoke router wants to transmit a packet to another spoke router, it can now use NHRP to dynamically determine the required destination address of the target spoke router. (The hub router acts as the NHRP server, handling the request for the source spoke router.) The two spoke routers dynamically create an IPSec tunnel between them so data can be directly transferred.

Feature Design of Dynamic Multipoint VPN (DMVPN)

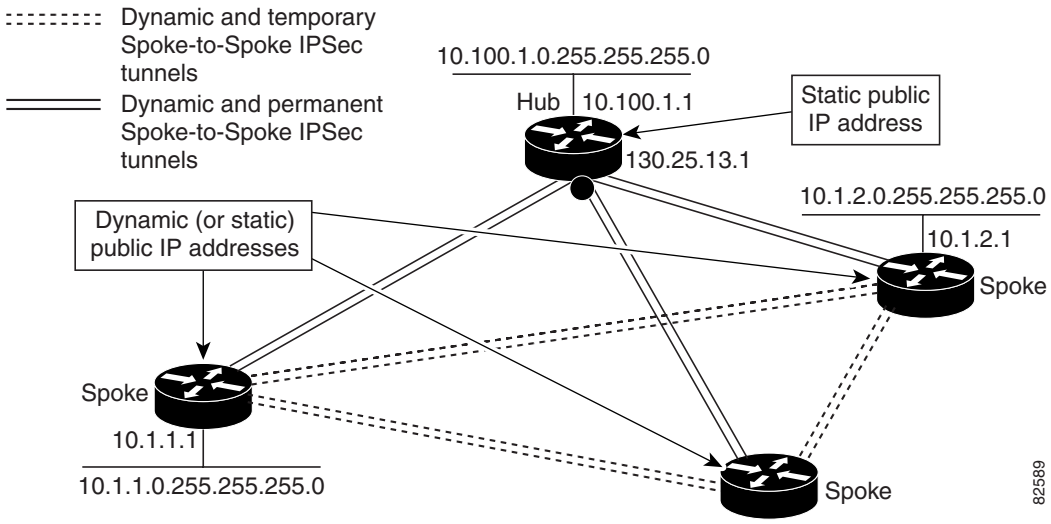
The Dynamic Multipoint VPN (DMVPN) feature combines GRE tunnels, IPSec encryption, and NHRP routing to provide users an ease of configuration via crypto profiles, which override the requirement for defining static crypto maps, and dynamic discovery of tunnel endpoints.

This feature relies on the following two Cisco technologies:

- NHRP—A client and server protocol where the hub is the server and the spokes are the clients. The hub maintains an NHRP database of the public interface addresses of the each spoke. Each spoke registers its real address when it boots and queries the NHRP database for real addresses of the destination spokes in order to build direct tunnels.
- mGRE Tunnel Interface —Allows a single GRE interface to support multiple IPSec tunnels and simplifies the size and complexity of the configuration.

The topology shown in Figure 73 and the corresponding bullets explain how this feature works.

Figure 73 Sample mGRE and IPSec Integration Topology



- Each spoke has a permanent IPSec tunnel to the hub, not to the other spokes within the network. Each spoke registers as clients of the NHRP server.
- When a spoke needs to send a packet to a destination (private) subnet on another spoke, it queries the NHRP server for the real (outside) address of the destination (target) spoke.
- After the originating spoke learns the peer address of the target spoke, it can initiate a dynamic IPSec tunnel to the target spoke.
- The spoke-to-spoke tunnel is built over the multipoint GRE interface.
- The spoke-to-spoke links are established on demand whenever there is traffic between the spokes. Thereafter, packets are able to bypass the hub and use the spoke-to-spoke tunnel.

Note After a preconfigured amount of inactivity on the spoke-to-spoke tunnels, the router will tear down those tunnels to save resources (IPSec security associations).

Note The traffic profile should follow the 80-20% rule: 80% of the traffic consists of spoke-to-hub traffic, and 20% of the traffic consists of spoke-to-spoke traffic.

IPSec Profiles

IPSec profiles abstract IPSec policy information into a single configuration entity, which can be referenced by name from other parts of the configuration. Therefore, users can configure things such as GRE tunnel protection with a single line of configuration. By referencing an IPSec profile, the user does not have to configure an entire crypto map configuration.

An IPSec profile contains only IPSec information; that is, it does not contain any access list information or peering information.

How to Configure DMVPN

To enable multipoint GRE (mGRE) and IPSec tunneling for hub and spoke routers, you must configure a crypto map that uses a global IPSec policy template and configure your mGRE tunnel for IPSec encryption. This section contains the following procedures:

- Configure an IPSec Profile, page 1057 (required)
- Configure the Hub for DMVPN, page 1059 (required)
- Configure the Spoke for DMVPN, page 1060 (required)
- Verify Dynamic Multipoint VPN (DMVPN), page 1063 (optional)

Configure an IPSec Profile

The IPSec profile shares most of the same commands with the crypto map configuration, but only a subset of the commands are valid in an IPSec profile. Only commands that pertain to an IPSec policy can be issued under an IPSec profile; you cannot specify the IPSec peer address or the access control list (ACL) to match the packets that are to be encrypted.

Prerequisites

Before configuring an IPSec profile, you must define a transform set by using the **crypto ipsec transform-set** command.

SUMMARY STEPS

1. **enable**
2. **configure** {**terminal** | **memory** | **network**}
3. **crypto ipsec profile** *name*
4. **set transform-set** *transform-set-name*
5. **set identity**
6. **set security association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes*}
7. **set pfs** [**group1** | **group2**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**configure** {**terminal**	**memory**	**network**} **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto ipsec profile** *name* **Example:** `Router(config)# crypto ipsec profile vpnprof`	Defines the IPSec parameters that are to be used for IPSec encryption between "spoke and hub" and "spoke and spoke" routers. This command enters crypto map configuration mode. • The *name* argument specifies the name of the IPSec profile.
Step 4	**set transform-set** *transform-set-name* **Example:** `Router(config-crypto-map)# set transform-set trans2`	Specifies which transform sets can be used with the IPSec profile. • The *transform-set-name* argument specifies the name of the transform set.
Step 5	**set identity** **Example:** `Router(config-crypto-map)# set identity`	(Optional) Specifies identity restrictions to be used with the IPSec profile.
Step 6	**set security association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes*} **Example:** `Router(config-crypto-map)# set security-association lifetime seconds 60`	(Optional) Overrides the global lifetime value for the IPSec profile. • The **seconds** *seconds* option specifies the number of seconds a security association will live before expiring; the **kilobytes** *kilobytes* option specifies the volume of traffic (in kilobytes) that can pass between IPSec peers using a given security association before that security association expires.
Step 7	**set pfs** [**group1** | **group2**] **Example:** `Router(config-crypto-map)# set pfs group2`	(Optional) Specifies that IP Security should ask for perfect forward secrecy (PFS) when requesting new security associations for this IPSec profile. If this command is not specified, the default (**group1**) will be enabled. • The **group1** keyword specifies that IPSec should use the 768-bit Diffie-Hellman (DH) prime modulus group when performing the new DH exchange; the **group2** keyword specifies the 1024-bit DH prime modulus group.

What to Do Next

Proceed to the following sections "Configure the Hub for DMVPN" and "Configure the Spoke for DMVPN."

Configure the Hub for DMVPN

To configure the hub router for mGRE and IPSec integration (that is, associate the tunnel with the IPSec profile configured in the previous procedure), use the following commands:

SUMMARY STEPS

1. **enable**
2. **configure** {**terminal** | **memory** | **network**}
3. **interface tunnel** *number*
4. **ip address** *ip-address mask* [*secondary*]
5. **ip mtu** *bytes*
6. **ip nhrp authentication** *string*
7. **ip nhrp map multicast dynamic**
8. **ip nhrp network-id** *number*
9. **tunnel source** {*ip-address* | *type number*}
10. **tunnel key** *key-number*
11. **tunnel mode gre multipoint**
12. **tunnel protection ipsec profile** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**configure** {**terminal** | **memory** | **network**} **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 5`	Configures a tunnel interface and enters interface configuration mode • The *number* argument specifies the number of the tunnel interface that you want to create or configure. There is no limit on the number of tunnel interfaces you can create.
Step 4	**ip address** *ip-address mask* [*secondary*] **Example:** `Router(config-if)# ip address 10.0.0.1 255.255.255.0`	Sets a primary or secondary IP address for the tunnel interface.

	Command or Action	Purpose
Step 5	**ip mtu** *bytes* **Example:** `Router(config-if)# ip mtu 1416`	Sets the maximum transmission unit (MTU) size, in bytes, of IP packets sent on an interface.
Step 6	**ip nhrp authentication** *string* **Example:** `Router(config-if)# ip nhrp authentication donttell`	Configures the authentication string for an interface using NHRP.
Step 7	**ip nhrp map multicast dynamic** **Example:** `Router(config-if)# ip nhrp map multicast dynamic`	Allows NHRP to automatically add spoke routers to the multicast NHRP mappings.
Step 8	**ip nhrp network-id** *number* **Example:** `Router(config-if)# ip nhrp network-id 99`	Enables NHRP on an interface. • The *number* argument specifies a globally unique 32-bit network identifier from a nonbroadcast multiaccess (NBMA) network. The range is from 1 to 4294967295.
Step 9	**tunnel source** {*ip-address* | *type number*} **Example:** `tunnel source Ethernet0`	Sets source address for a tunnel interface.
Step 10	**tunnel key** *key-number* **Example:** `tunnel key 100000`	Enables an ID key for a tunnel interface. • The *key-number* argument specifies a number from 0 to 4,294,967,295 that identifies the tunnel key.
Step 11	**tunnel mode gre multipoint** **Example:** `Router(config-if)# tunnel mode gre multipoint`	Sets the encapsulation mode to mGRE for the tunnel interface.
Step 12	**tunnel protection ipsec profile** *name* **Example:** `Router(config-if)# tunnel protection ipsec profile vpnprof`	Associates a tunnel interface with an IPSec profile. • The *name* argument specifies the name of the IPSec profile; this value must match the *name* specified in the **crypto ipsec profile** *name* command.

Configure the Spoke for DMVPN

To configure spoke routers for mGRE and IPSec integration, use the following commands:

SUMMARY STEPS

1. **enable**
2. **configure** {**terminal** | **memory** | **network**}

3. **interface tunnel** *number*
4. **ip address** *ip-address mask* [*secondary*]
5. **ip mtu** *bytes*
6. **ip nhrp authentication** *string*
7. **ip nhrp map** *hub-tunnel-ip-address hub-physical-ip-address*
8. **ip nhrp map multicast** *hub-physical-ip-address*
9. **ip nhrp nhs** *hub-tunnel-ip-address*
10. **ip nhrp network-id** *number*
11. **tunnel source** {*ip-address* | *type number*}
12. **tunnel key** *key-number*
13. **tunnel mode gre multipoint**

    or

    tunnel destination *hub-physical-ip-address*
14. **tunnel protection ipsec profile** *name*

DETAILED STEPS:

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**configure** {**terminal** | **memory** | **network**} **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 5`	Configures a tunnel interface and enters interface configuration mode • The *number* argument specifies the number of the tunnel interface that you want to create or configure. There is no limit on the number of tunnel interfaces you can create.
Step 4	**ip address** *ip-address mask* [*secondary*] **Example:** `Router(config-if)# ip address 10.0.0.1 255.255.255.0`	Sets a primary or secondary IP address for the tunnel interface.
Step 5	**ip mtu** *bytes* **Example:** `Router(config-if)# ip mtu 1416`	Sets the maximum transmission unit (MTU) size, in bytes, of IP packets sent on an interface.

	Command or Action	Purpose
Step 6	**ip nhrp authentication** *string* **Example:** `Router(config-if)# ip nhrp authentication donttell`	Configures the authentication string for an interface using NHRP.
Step 7	**ip nhrp map** *hub-tunnel-ip-address hub-physical-ip-address* **Example:** `Router(config-if)# ip nhrp map 10.0.0.1 172.17.0.1`	Statically configures the IP-to-NonBroadcast MutiAccess (NBMA) address mapping of IP destinations connected to an MBMA network. • *hub-tunnel-ip-address*—Defines the NHRP server at the hub, which is permanently mapped to the static public IP address of the hub. • *hub-physical-ip-address*—Defines the static public IP address of the hub.
Step 8	**ip nhrp map multicast** *hub-physical-ip-address* Example: `Router(config-if)# ip nhrp map multicast 172.17.0.1`	Enables the use of a dynamic routing protocol between the spoke and hub, and sends multicast packets to the hub router.
Step 9	**ip nhrp nhs** *hub-tunnel-ip-address* **Example:** `Router(config-if)# ip nhrp nhs 10.0.0.1`	Configures the hub router as the NHRP next-hop server.
Step 10	**ip nhrp network-id** *number* **Example:** `Router(config-if)# ip nhrp network-id 99`	Enables NHRP on an interface. • The *number* argument specifies a globally unique 32-bit network identifier from a nonbroadcast multiaccess (NBMA) network. The range is from 1 to 4294967295.
Step 11	**tunnel source** {*ip-address* | *type number*} **Example:** `tunnel source Ethernet0`	Sets source address for a tunnel interface.
Step 12	**tunnel key** *key-number* **Example:** `tunnel key 100000`	Enables an ID key for a tunnel interface. • The *key-number* argument specifies a number from 0 to 4,294,967,295 that identifies the tunnel key.

	Command or Action	Purpose
Step 13	`tunnel mode gre multipoint` OR `tunnel destination` *`hub-physical-ip-address`* **Example:** `Router(config-if)# tunnel mode gre multipoint` OR `Router(config-if)# tunnel destination`	Sets the encapsulation mode to mGRE for the tunnel interface. Use this command if data traffic can use dynamic spoke-to-spoke traffic. Specifies the destination for a tunnel interface. Use this command if data traffic can use hub-and-spoke tunnels.
Step 14	**`tunnel protection ipsec profile`** *`name`* **Example:** `Router(config-if)# tunnel protection ipsec profile vpnprof`	Associates a tunnel interface with an IPSec profile. • The *name* argument specifies the name of the IPSec profile; this value must match the *name* specified in the **crypto ipsec profile** *name* command.

Verify Dynamic Multipoint VPN (DMVPN)

To verify that the Dynamic Multipoint VPN (DMVPN) feature is working, perform the following optional steps:

SUMMARY STEPS:

1. **enable**
2. **show crypto isakmp sa**
3. **show crypto map**
4. **show ip nhrp**

DETAILED STEPS:

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**`show crypto isakmp sa`** **Example:** `Router# show crypto isakmp sa`	(Optional) Displays all current IKE security associations (SAs) at a peer.

	Command or Action	Purpose
Step 3	**show crypto map** **Example:** `Router# show crypto map`	(Optional) Displays the crypto map configuration.
Step 4	**show ip nhrp** **Example:** `show ip nhrp`	(Optional) Displays the NHRP cache.

Configuration Examples for Dynamic Multipoint VPN (DMVPN) Feature

This section provides the following comprehensive configuration examples:

- Hub Configuration for DMVPN Example, page 1064
- Spoke Configuration for DMVPN Example, page 1065
- Verify Dynamic Multipoint VPN (DMVPN) Example, page 1066

Hub Configuration for DMVPN Example

In the following example, which configures the hub router for multipoint GRE and IPSec integration, no explicit configuration lines are needed for each spoke; that is, the hub is configured with a global IPSec policy template that all spoke routers can talk to. In this example, EIGRP is configured to run over the private physical interface and the tunnel interface.

```
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco47 address 0.0.0.0
!
crypto ipsec transform-set trans2 esp-des esp-md5-hmac
!
crypto ipsec profile vpnprof
 set transform-set trans2
!
interface Tunnel0
 bandwidth 1000
 ip address 10.0.0.1 255.255.255.0
! Ensures longer packets are fragmented before they are encrypted; otherwise, the
! receiving router would have to do the reassembly.
 ip mtu 1416
! The following line must match on all nodes that want to use this mGRE tunnel:
 ip nhrp authentication donttell
 ip nhrp map multicast dynamic
! The following line must match on all nodes that want to use this mGRE tunnel:
 ip nhrp network-id 99
 ip nhrp holdtime 300
! Turns off split horizon on the mGRE tunnel interface; otherwise, EIGRP will not
! advertise routes that are learned via the mGRE interface back out that interface.
 no ip split-horizon eigrp 1
! Enables dynamic, direct spoke-to-spoke tunnels when using EIGRP.
 delay 1000
```

```
! Sets IPSec peer address to Ethernet interface's public address.
 tunnel source Ethernet0
 tunnel mode gre multipoint
! The following line must match on all nodes that want to use this mGRE tunnel.
 tunnel key 100000
 tunnel protection ipsec profile vpnprof
!
interface Ethernet0
 ip address 172.17.0.1 255.255.255.0
!
interface Ethernet1
 ip address 192.168.0.1 255.255.255.0
!
router eigrp 1
 network 10.0.0.0 0.0.0.255 area 0
 network 192.168.0.0 0.0.0.255 area 0
!
```

Spoke Configuration for DMVPN Example

In the following example, all spokes are configured the same except for tunnel and local interface address, thereby, reducing necessary configurations for the user:

```
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco47 address 0.0.0.0
!
crypto ipsec transform-set trans2 esp-des esp-md5-hmac
!
crypto ipsec profile vpnprof
 set transform-set trans2
!
interface Tunnel0
 bandwidth 1000
 ip address 10.0.0.2 255.255.255.0
 ip mtu 1416
! The following line must match on all nodes that want to use this mGRE tunnel:
 ip nhrp authentication donttell
! Definition of NHRP server at the hub (10.0.0.1), which is permanently mapped to the
! static public address of the hub (172.17.0.1).
 ip nhrp map 10.0.0.1 172.17.0.1
! Sends multicast packets to the hub router, and enables the use of a dynamic routing
! protocol between the spoke and the hub.
 ip nhrp map multicast 172.17.0.1
! The following line must match on all nodes that want to use this mGRE tunnel:
 ip nhrp network-of 99
 ip nhrp holdtime 300
! Configures the hub router as the NHRP next-hop server.
 ip nhrp nhs 10.0.0.1
 delay 1000
 tunnel source Ethernet0
 tunnel mode gre multipoint
! The following line must match on all nodes that want to use this mGRE tunnel:
 tunnel key 100000
 tunnel protection ipsec profile vpnprof
!
! This is a spoke, so the public address might be dynamically assigned via DHCP.
interface Ethernet0
 ip address dhcp hostname Spoke1
!
interface Ethernet1
 ip address 192.168.1.1 255.255.255.0
```

```
!
! EIGRP is configured to run over the inside physical interface and the tunnel.
router eigrp 1
 network 10.0.0.0 0.0.0.255
 network 192.168.1.0 0.0.0.255
```

Verify Dynamic Multipoint VPN (DMVPN) Example

In the following examples, the output is displayed for each command in the task.

Sample Output for the show crypto isakmp sa Command

The following sample output is displayed after IKE negotiations have successfully completed between two peers.

```
Router# show crypto isakmp sa

dst             src             state          conn-id    slot
172.17.63.19    172.16.175.76   QM_IDLE              2       0
172.17.63.19    172.17.63.20    QM_IDLE              1       0
172.16.175.75   172.17.63.19    QM_IDLE              3       0
```

Sample Output for the show crypto map Command

The following sample output is displayed after a crypto map has been configured. Table 48 describes the significant fields shown in the display.

```
Router# show crypto map

Crypto Map "Tunnel5-head-0" 10 ipsec-isakmp
        Profile name: vpnprof
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={trans2, }

Crypto Map "Tunnel5-head-0" 20 ipsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 172.16.175.75
        Extended IP access list
            access-list permit gre host 172.17.63.19 host 172.16.175.75
        Current peer: 172.16.175.75
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={trans2, }

Crypto Map "Tunnel5-head-0" 30 ipsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 172.17.63.20
        Extended IP access list
            access-list permit gre host 172.17.63.19 host 172.17.63.20
        Current peer: 172.17.63.20
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={trans2, }
```

```
Crypto Map "Tunnel5-head-0" 40 ipsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 172.16.175.76
        Extended IP access list
            access-list permit gre host 172.17.63.19 host 172.16.175.76
        Current peer: 172.16.175.76
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={trans2, }
        Interfaces using crypto map Tunnel5-head-0:
                Tunnel5
```

Table 48 Field Descriptions for the show crypto map Command

Field	Description
Crypto Map “Tunnel5-head-0” 10 ipsec-isakmp	Crypto map name and number. Also indicated whether the crypto map is ipsec-manual or ipsec-isakmp.
Profile name	The name of the IPSec profile that was specified via the **crypto ipsec profile** *name* command.
Peer	IP address of the IPSec peer.
Security association lifetime	The lifetime value for the crypto map. Kilobytes specifies the volume of traffic that can pass between IPSec peers using a given security association before that security association expires; seconds specifies the number of seconds a security association will live before expiring. This value is configured via the **set security association lifetime** command.
PFS	Specifies whether IPSec should use the 768-bit DH prime modulus group (group1) or the 1024-bit DH prime modulus group (group2) when performing the new Diffie-Hellman exchange. This command is configured via the **set pfs** command.
Transform sets	Specifies the name of the transform set to be used with the crypto map entry. This value is configured via the **set transform-set** command.
Interfaces using crypto map Tunnel5-head-0	The name of the interface that is using the named crypto map.

Sample Output for the show ip nhrp Command

The following sample output shows that NHRP registration occurred, thereby, allowing the user to apply tunnel protection. Table 49 describes the significant fields shown in the display.

```
Router# show ip nhrp

10.10.1.75/32 via 10.10.1.75, Tunnel5 created 00:32:11, expire 00:01:46
  Type: dynamic, Flags: authoritative unique registered
  NBMA address: 172.16.175.75
10.10.1.76/32 via 10.10.1.76, Tunnel5 created 00:26:41, expire 00:01:37
  Type: dynamic, Flags: authoritative unique registered
  NBMA address: 172.16.175.76
10.10.1.77/32 via 10.10.1.77, Tunnel5 created 00:31:26, expire 00:01:33
  Type: dynamic, Flags: authoritative unique registered
  NBMA address: 172.17.63.20
```

Table 49 Field Descriptions for the show ip nhrp Command

Field	Description
Tunnel5 created 00:32:11	Interface type and number (in this case, tunnel and tunnel interface) and how long ago it was created (hours:minutes:seconds).
expire 00:01:46	Time in which the positive and negative authoritative NBMA address will expire (hours:minutes:seconds). This value is based on the **ip nhrp holdtime** command.
Type	• dynamic—NBMA address was obtained from NHRP Request packet. • static—NBMA address was statically configured.
Flags	• authoritative—Indicates that the NHRP information was obtained from the Next Hop Server or router that maintains the NBMA-to-IP address mapping for a particular destination. • implicit—Indicates that the information was learned not from an NHRP request generated from the local router, but from an NHRP packet being forwarded or from an NHRP request being received by the local router. • negative—For negative caching; indicates that the requested NBMA mapping could not be obtained.
NBMA address	Nonbroadcast multiaccess address. The address format is appropriate for the type of network being used (for example, ATM, Ethernet, Switched Multimegabit Data Service (SMDS), or multipoint tunnel).

Additional References

For additional information related to Multipoint GRE and IPSec Integration, refer to the following references:

- Related Documents, page 1068
- Standards, page 1069
- MIBs, page 1069
- RFCs, page 1070
- Technical Assistance, page 1070

Related Documents

Related Topic	Document Title
IPSec configuration tasks	The chapter "Configuring IPSec Network Security" in the *Cisco IOS Security Configuration Guide*, Release 12.2
Additional IPSec Commands	The chapter "IPSec Network Security Commands" in the *Cisco IOS Security Command Reference*, Release 12.2
IKE configuration tasks such as defining an IKE policy	The chapter "Configuring Internet Key Exchange Security Protocol" in the *Cisco IOS Security Configuration Guide*, Release 12.2

Related Topic	Document Title
Tunnel interface configuration tasks	The section "Configuring a Tunnel Interface" in the chapter "Configuring Logical Interfaces" in the *Cisco IOS Interface Configuration Guide*, Release 12.2
GRE tunnel keepalive information	*Generic Routing Encapsulation (GRE) Tunnel Keepalive*, Cisco IOS Release 12.2(8)T feature module
Additional NHRP Commands	The chapter "IP Addressing Commands" in the *Cisco IOS IP Command Reference, Volume 1 of 3: Addressing and Services*, Release 12.2

Standards

Standards[1]	Title
None	—

1. Not all supported standards are listed.

MIBs

MIBs[1]	MIBs Link
None	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

1. Not all supported MIBs are listed.

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

RFCs[1]	Title
None	—

1. Not all supported RFCs are listed.

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, tools, and lots more. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

New Commands

- **crypto ipsec profile**
- **ip nhrp map multicast dynamic**
- **tunnel protection**

Modified Command

- **tunnel mode**

Glossary

AM—aggressive mode. A mode during IKE negotiation. Compared to MM, AM eliminates several steps, making it faster but less secure than MM. Cisco IOS software will respond in aggressive mode to an IKE peer that initiates aggressive mode.

GRE—generic routing encapsulation. Tunnels that provide a specific pathway across the shared WAN and encapsulate traffic with new packet headers to ensure delivery to specific destinations. The network is private because traffic can enter a tunnel only at an endpoint. Tunnels do not provide true confidentiality (encryption does) but can carry encrypted traffic.

GRE tunneling can also be used to encapsulate non-IP traffic into IP and send it over the Internet or IP network. The Internet Package Exchange (IPX) and AppleTalk protocols are examples of non-IP traffic.

IKE—Internet Key Exchange. A hybrid protocol that implements Oakley key exchange and Skeme key exchange inside the ISAKMP framework. Although IKE can be used with other protocols, its initial implementation is with IPSec. IKE provides authentication of the IPSec peers, negotiates IPSec keys, and negotiates IPSec security associations.

IPSec—IP Security. A framework of open standards developed by the Internet Engineering Task Force (IETF). IPSec provides security for transmission of sensitive information over unprotected networks such as the Internet. IPSec acts at the network layer, protecting and authenticating IP packets between participating IPSec devices ("peers"), such as Cisco routers.

ISAKMP—Internet Security Association Key Management Protocol. A protocol framework that defines payload formats, the mechanics of implementing a key exchange protocol, and the negotiation of a security association.

MM—main mode. Mode that is slower than aggressive mode but more secure and more flexible than aggressive mode because it can offer an IKE peer more security proposals. The default action for IKE authentication (rsa-sig, rsa-encr, or preshared) is to initiate main mode.

NHRP—Next Hop Resolution Protocol. Routers, access servers, and hosts can use NHRP to discover the addresses of other routers and hosts connected to a nonbroadcast multiaccess (NBMA) network.

The Cisco implementation of NHRP supports the IETF draft version 11 of *NBMA Next Hop Resolution Protocol (NHRP)*.

The Cisco implementation of NHRP supports IP Version 4, Internet Packet Exchange (IPX) network layers, and, at the link layer, ATM, Ethernet, SMDS, and multipoint tunnel networks. Although NHRP is available on Ethernet, NHRP need not be implemented over Ethernet media because Ethernet is capable of broadcasting. Ethernet support is unnecessary (and not provided) for IPX.

PFS—Perfect Forward Secrecy. A cryptographic characteristic associated with a derived shared secret value. With PFS, if one key is compromised, previous and subsequent keys are not compromised, because subsequent keys are not derived from previous keys.

SA—security association. Describes how two or more entities will utilize security services to communicate securely. For example, an IPSec SA defines the encryption algorithm (if used), the authentication algorithm, and the shared session key to be used during the IPSec connection.

Both IPSec and IKE require and use SAs to identify the parameters of their connections. IKE can negotiate and establish its own SA. The IPSec SA is established either by IKE or by manual user configuration.

transform—The list of operations done on a dataflow to provide data authentication, data confidentiality, and data compression. For example, one transform is the ESP protocol with the HMAC-MD5 authentication algorithm; another transform is the AH protocol with the 56-bit DES encryption algorithm and the ESP protocol with the HMAC-SHA authentication algorithm.

VPN—Virtual Private Network. A framework that consists of multiple peers transmitting private data securely to one another over an otherwise public infrastructure. In this framework, inbound and outbound network traffic is protected using protocols that tunnel and encrypt all data. This framework permits networks to extend beyond their local topology, while remote users are provided with the appearance and functionality of a direct network connection.

Note Refer to the *Internetworking Terms and Acronyms* for terms not included in this glossary.

Easy VPN Remote RSA Signature Support

The Easy VPN Remote RSA Signature Support feature provides for the support of Rivest, Shamir, and Adelman (RSA) signatures on Easy VPN remote devices. The support is provided through RSA certificates that can be stored on or off the remote device.

Feature History for Easy VPN Remote RSA Signature Support

Release	Modification
12.3(7)T1	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for Easy VPN Remote RSA Signature Support, page 1073
- Restrictions for Easy VPN Remote RSA Signature Support, page 1074
- Information About Easy VPN Remote RSA Signature Support, page 1074
- How to Configure RSA Signatures, page 1074
- Additional References, page 1075
- Command Reference, page 1076

Prerequisites for Easy VPN Remote RSA Signature Support

- You must have a Cisco Virtual Private Network (VPN) remote device and be familiar with configuring the device.
- You must have a certification authority (CA) available to your network before you configure this interoperability feature. The CA must support the public key infrastructure (PKI) protocol of Cisco Systems, which is the Simple Certificate Enrollment Protocol (SCEP) (formerly called certificate enrollment protocol [CEP]).

- You should be familiar with IP Security (IPSec) and PKI.
- You should be familiar with configuring RSA key pairs.
- You should be familiar with configuring CAs.

Restrictions for Easy VPN Remote RSA Signature Support

- This feature should be configured only when you also configure both IPSec and Internet Key Exchange (IKE) in your network.
- The Cisco IOS software does not support CA server public keys greater than 2048 bits.

Information About Easy VPN Remote RSA Signature Support

To configure the Easy VPN Remote RSA Signature Support feature, you should understand the following concept:

- Easy VPN Remote RSA Signature Support Overview, page 1074

Easy VPN Remote RSA Signature Support Overview

The Easy VPN Remote RSA Signature Support feature allows you to configure RSA signatures on your Easy VPN remote device. The signatures can be stored on or off your remote device.

How to Configure RSA Signatures

This section contains the following procedure:

- Configuring Easy VPN Remote RSA Signature Support, page 1074

Configuring Easy VPN Remote RSA Signature Support

The RSA signatures for an Easy VPN remote device are configured the same way that you would configure RSA signatures for any other Cisco device. (For information about configuring RSA signatures, refer to the "Configuring Certification Authority Interoperability" chapter of the "IP Security and Encryption" section of the *Cisco IOS Security Configuration Guide*, Release 12.3.)

To enable the RSA signatures, when you are configuring the Easy VPN remote and assigning the configuration to the outgoing interface, you must omit the **group** command. The content of the first Organizational Unit (OU) field will be used as the group. (For information about configuring Cisco Easy VPN remote devices, refer to the feature document "*Cisco Easy VPN Remote*," Release 12.3(7)T.)

To troubleshoot your Easy VPN remote RSA signature configuration, you can use the following **debug** commands. The **debug** commands can be used in any order or individually.

SUMMARY STEPS

1. **enable**

2. **debug crypto ipsec client ezvpn**
3. **debug crypto isakmp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto ipsec client ezvpn** **Example:** `Router# debug crypto ipsec client ezvpn`	Displays information about the VPN tunnel as it relates to the Easy VPN remote configuration.
Step 3	**debug crypto isakmp** **Example:** `Router# debug crypto isakmp`	Displays messages about IKE events.

Additional References

The following sections provide references related to Easy VPN Remote RSA Signature Support.

Related Documents

Related Topic	Document Title
Configuring IPSec	"IP Security and Encryption Overview" chapter of the *Cisco IOS Security Configuration Guide*.
Configuring IKE	"Configuring Internet Key Exchange Security Protocol" chapter of the "IP Security and Encryption" section of the *Cisco IOS Security Configuration Guide*.
Configuring RSA key pairs	Feature document "*Exporting and Importing RSA Keys*," Release 12.2(15)T.
Declaring a CA	"Configuring Certification Authority Interoperability" chapter of the "IP Security and Encryption" section of the *Cisco IOS Security Configuration Guide*, Release 12.3
Configuring a Cisco Easy VPN remote device	Feature document "*Cisco Easy VPN Remote*," Release 12.3(7)T
Security commands	*Cisco IOS Security Command Reference*, Release 12.3 T

Standards

Standards	Title
There are no new or modified standards associated with this feature.	—

MIBs

MIBs	MIBs Link
There are no new or modified MIBs associated with this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
There are no new or modified RFCs associated with this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

This feature uses no new or modified commands. To see the command pages for the commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

Easy VPN Server

The Easy VPN Server feature introduces server support for the Cisco VPN Client Release 3.x and later software clients and Cisco VPN hardware clients (such as the Cisco 800, Cisco 900, Cisco 1700, VPN 3002, and PIX 501 devices). The feature allows a remote end user to communicate using IP Security (IPSec) with any Cisco IOS Virtual Private Network (VPN) gateway. Centrally managed IPSec policies are "pushed" to the client by the server, minimizing configuration by the end user.

Feature History for Easy VPN Server

Release	Modification
12.2(8)T	This feature was introduced.
12.3(2)T	New attributes were added to the server group, and the following commands, which correspond to the added attributes, were added: **access-restrict, firewall are-u-there**, **group-lock**, **include-local-lan**, and **save-password**.
12.3(4)T	RADIUS support for user profiles, user-based policy control, session monitoring for VPN group access, backup-gateway list, and PFS were added.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Restrictions for Easy VPN Server, page 1078
- Information About Easy VPN Server, page 1078
- How to Configure Easy VPN Server, page 1087
- Configuration Examples for Easy VPN Server, page 1098
- Additional References, page 1101
- Command Reference, page 1102
- Glossary, page 1104

Restrictions for Easy VPN Server

Nonsupported Protocols

Table 50 outlines IPSec protocol options and attributes that currently are *not* supported by Cisco VPN clients, so these options and attributes should not be configured on the router for these clients.

Table 50 Nonsupported IPSec Protocol Options and Attributes

Options	Attributes
Authentication Types	Authentication with public key encryption Digital Signature Standard (DSS)
Diffie-Hellman (D-H) groups	1
IPSec Protocol Identifier	IPSEC_AH
IPSec Protocol Mode	Transport mode
Miscellaneous	Manual keys Perfect Forward Secrecy (PFS)

Cisco Secure VPN Client 1.x Restrictions

When used with this feature, the Cisco Secure VPN Client 1.x has the following restrictions:

- It does not support dead peer detection (DPD) or any other keepalive scheme.
- It does not support initial contact.

This feature cannot use per-group attribute policy profiles such as IP addresses, Domain Name Service (DNS), and split tunnel access. Thus, customers must continue to use existing, globally defined parameters for IP address assignment, Windows Internet Naming Service (WINS) and DNS, and preshared keys.

Information About Easy VPN Server

Before using the Easy VPN Server Enhancements feature, you should understand the following concepts:

- How It Works, page 1079
- RADIUS Support for Group Profiles, page 1080
- RADIUS Support for User Profiles, page 1083
- Supported Protocols, page 1084
- Functions Supported by Easy VPN Server, page 1084

How It Works

When the client initiates a connection with a Cisco IOS VPN device, the "conversation" that occurs between the peers consists of device authentication via Internet Key Exchange (IKE), followed by user authentication using IKE Extended Authentication (Xauth), VPN policy push (using Mode Configuration), and IPSec security association (SA) creation. An overview of this process is as follows:

- The client initiates IKE Phase 1 via aggressive mode (AM) if a preshared key is to be used for authentication; the client initiates main mode (MM) if digital certificates are used. If the client identifies itself with a preshared key, the accompanying group name entered in the configuration GUI (ID_KEY_ID) is used to identify the group profile associated with this client. If digital certificates are used, the organizational unit (OU) field of a distinguished name (DN) is used to identify the group profile.

Note Because the client may be configured for preshared key authentication, which initiates IKE AM, it is recommended that the administrator change the identity of the Cisco IOS VPN device via the **crypto isakmp identity hostname** command. This will not affect certificate authentication via IKE MM.

- The client attempts to establish an IKE SA between its public IP address and the public IP address of the Cisco IOS VPN device. To reduce the amount of manual configuration on the client, every combination of encryption and hash algorithms, in addition to authentication methods and D-H group sizes, is proposed.
- Depending on its IKE policy configuration, the Cisco IOS VPN device will determine which proposal is acceptable to continue negotiating Phase 1.

Tip IKE policy is global for the Cisco IOS VPN device and can consist of several proposals. In the case of multiple proposals, the Cisco IOS VPN device will use the first match, so you should always list your most secure policies first.

Note Device authentication ends and user authentication begins at this point.

- After the IKE SA is successfully established, and if the Cisco IOS VPN device is configured for Xauth, the client waits for a "username/password" challenge and then responds to the challenge of the peer. The information that is entered is checked against authentication entities using authentication, authorization, and accounting (AAA) protocols such as RADIUS and TACACS+. Token cards may also be used via AAA proxy. During Xauth, it is also possible for a user-specific attribute to be retrieved if the credentials of that user are validated via RADIUS.

Note VPN devices that are configured to handle remote clients should always be configured to enforce user authentication.

- If the Cisco IOS VPN device indicates that authentication was successful, the client requests further configuration parameters from the peer. The remaining system parameters (for example, IP address, DNS, and split tunnel attributes) are pushed to the client at this time using Mode Configuration.

Note The IP address pool and group preshared key (if Rivest, Shamir, and Adelman [RSA] signatures are not being used) are the only required parameter in a group profile, all other parameters are optional.

- After each client is assigned an internal IP address via Mode Configuration, it is important that the Cisco IOS VPN device knows how to route packets through the appropriate VPN tunnel. Reverse route injection (RRI) will ensure that a static route is created on the Cisco IOS VPN device for each client internal IP address.

Note It is recommended that you enable RRI on the crypto map (static or dynamic) for the support of VPN clients unless the crypto map is being applied to a Generic Routing Encapsulation (GRE) tunnel that is already being used to distribute routing information.

- After the configuration parameters have been successfully received by the client, IKE quick mode is initiated to negotiate IPSec SA establishment.
- After IPSec SAs are created, the connection is complete.

RADIUS Support for Group Profiles

Group policy information is stored in a profile that can be defined locally in the router configuration or on a RADIUS server that is accessible by the Cisco IOS VPN device. If RADIUS is used, you must configure access to the server and allow the Cisco IOS VPN device to send requests to the server.

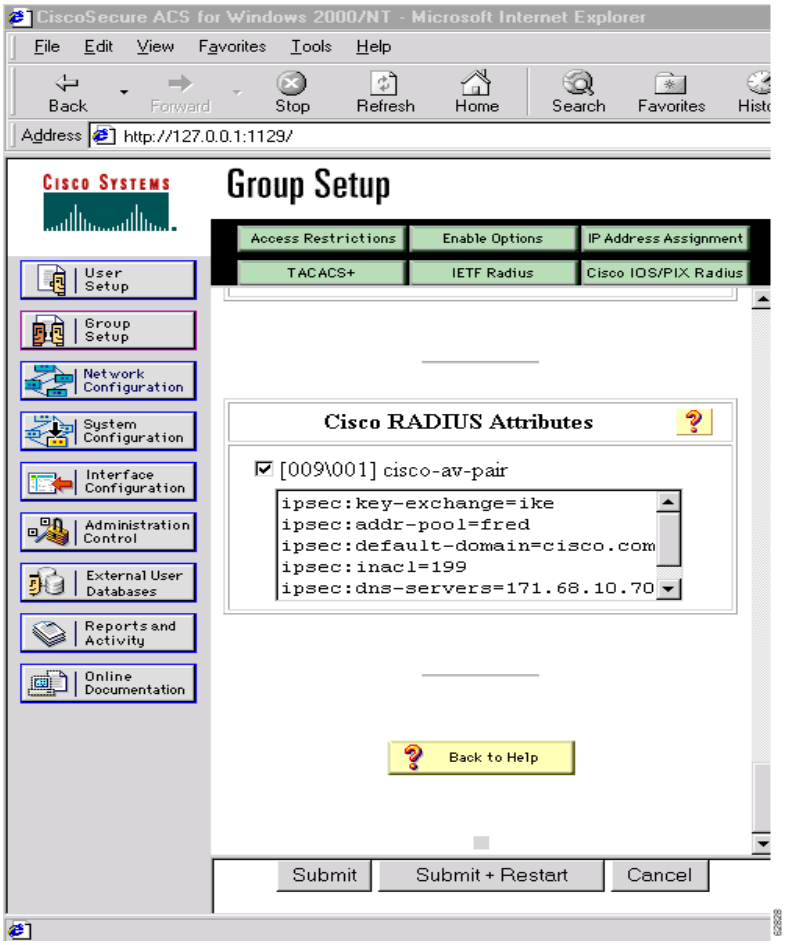
To define group policy attributes for RADIUS, you must do the following task on your RADIUS server:

- Define a user that has a name equal to the group name as defined in the client graphical user interface (GUI). For example, if users will be connecting to the Cisco IOS VPN device using the group name “sales,” you will need a user whose name is “sales.” The password for this user is “cisco,” which is a special identifier that is used by the router for RADIUS purposes. The username must then be made a member of a group in which the correct policy is defined. For simplicity, it is recommended that the group name be the same as the username.

For a Cisco Secure Access Control Server

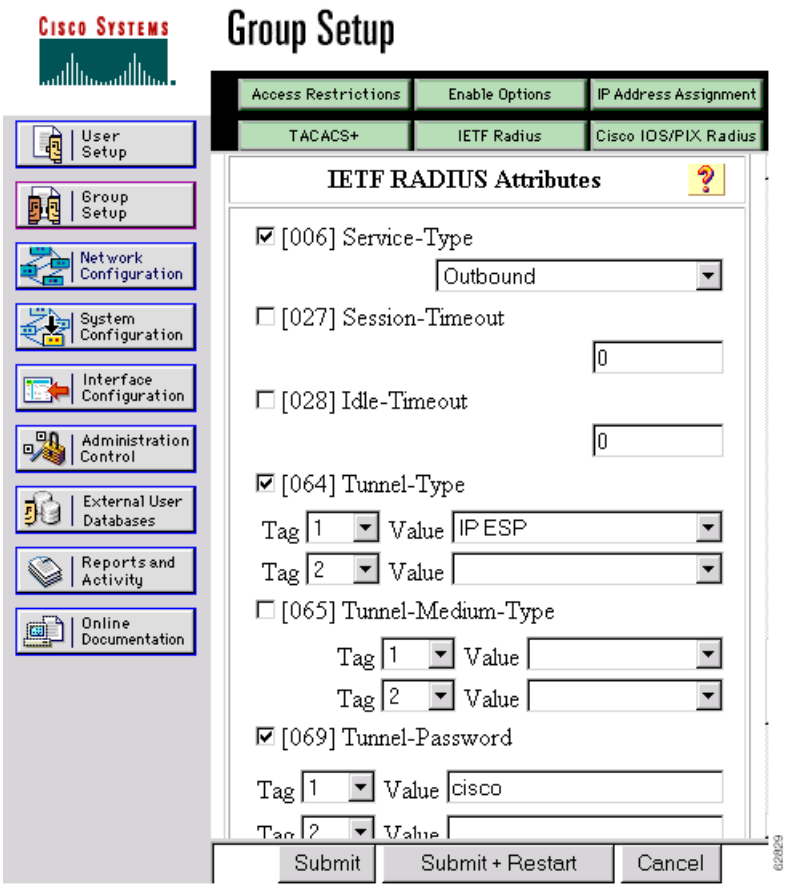
If you are using a Cisco Secure access control server (ACS), you may configure your remote access VPN group profiles on this server. To perform this task, you must ensure that Internet Engineering Task Force (IETF) RADIUS attributes are selected for group configuration as shown in Figure 74. (This figure also shows the compulsory attributes required for a remote access VPN group.) All values must be entered except the Tunnel-Password attribute, which is actually the preshared key for IKE purposes; if digital certificates are preferred, this attribute may be omitted.

Figure 74 IETF RADIUS Attributes Selection for Group Configuration



In addition to the compulsory attributes shown in Figure 74, other values can be entered that represent the group policy that is pushed to the remote client via Mode Configuration. Figure 75 shows an example of a group policy. All attributes are optional except the Addr-Pool attribute. The values of the attributes are the same as the setting that is used if the policy is defined locally on the router rather than in a RADIUS server. (These values are explained in the section "Defining Group Policy Information for Mode Configuration Push" later in this document.)

Figure 75 CiscoSecure ACS Group Policy Setup



After the group profile is created, a user who is a member of the group should be added. (Remember that the username that is defined maps to the group name as defined on the remote client, and the password defined for the username in the RADIUS database must be "cisco.") If digital certificates are the preferred method of IKE authentication, the username should reflect the OU field in the certificate presented by the remote client.

For All Other RADIUS Servers

Ensure that your RADIUS server allows you to define attribute-value (AV) pairs. (For an example, see the section "Configuring Cisco IOS for Easy VPN Server: Example" later in this document).

Note If digital certificates are used, the username defined in RADIUS must be equal to the OU field of the DN of the certificate of the client.

RADIUS Support for User Profiles

Attributes may also be applied on a per-user basis. If you apply attributes on a per-user basis, you can override a group attribute value with an individual user attribute. The attributes are retrieved at the time that user authentication via Xauth occurs. The attributes are then combined with group attributes and applied during Mode Configuration.

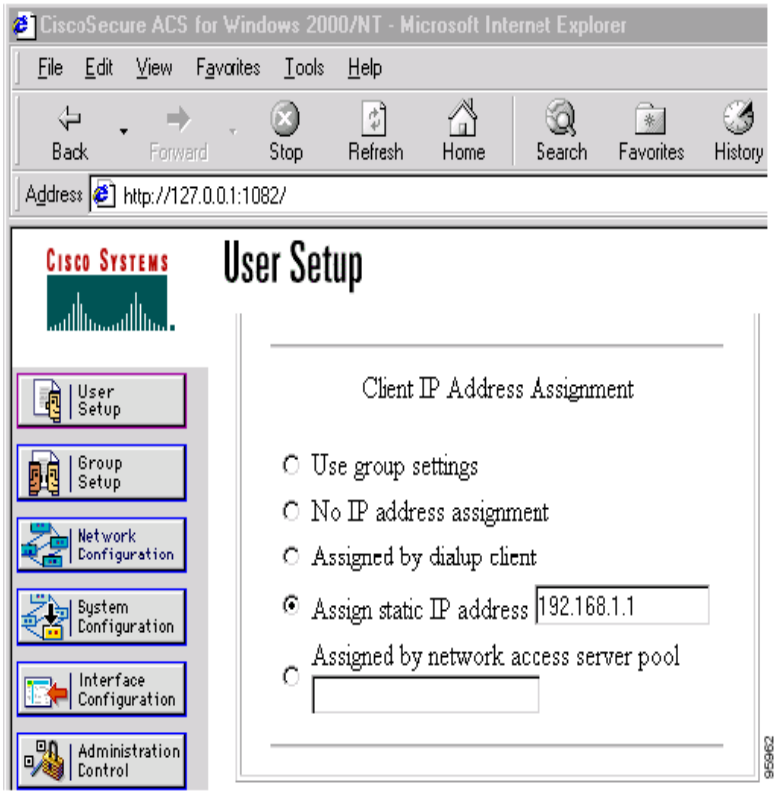
User-based attributes are available only if RADIUS is being used for user authentication.

To define user policy attributes for RADIUS, you must do the following task on your RADIUS server:

- Define a user or add attributes to the existing profile of a user in your RADIUS database. The password for the user will be used during Xauth user authentication, or you may proxy to a third-party server, such as a token card server.

Figure 76 shows how CiscoSecure ACS may be used for user authentication and for the assignment of a Framed-IP-Address attribute that may be pushed to the client. The presence of this attribute means that the local address pool defined for the group to which that user belongs will be overridden.

Figure 76 CiscoSecure ACS User Profile Setup



For All Other RADIUS Servers

Ensure that your RADIUS server allows you to define AV pairs. (For an example, see the "Configuring Cisco IOS for Easy VPN Server: Example" section later in this document.

Supported Protocols

Table 51 outlines supported IPSec protocol options and attributes that can be configured for this feature. (See Table 50 for nonsupported options and attributes.)

Table 51 Supported IPSec Protocol Options and Attributes

Options	Attributes
Authentication Algorithms	• Hashed Message Authentication Codes with Message Digest 5 (HMAC-MD5) • HMAC-Secure Hash Algorithm 1 (HMAC-SHA1)
Authentication Types	• Preshared keys • RSA digital signatures
D-H groups	• 2 • 5
Encryption Algorithms (IKE)	• Data Encryption Standard (DES) • Triple Data Encryption Standard (3DES)
Encryption Algorithms (IPSec)	• DES • 3DES • NULL
IPSec Protocol Identifiers	• Encapsulating Security Payload (ESP) • IP LZS compression (IPCOMP-LZS)
IPSec Protocol Mode	Tunnel mode

Functions Supported by Easy VPN Server

- Mode Configuration Version 6 Support, page 1084
- Xauth Version 6 Support, page 1085
- IKE DPD, page 1085
- Split Tunneling Control, page 1085
- Initial Contact, page 1085
- Group-Based Policy Control, page 1085
- User-Based Policy Control, page 1085
- Session Monitoring for VPN Group Access, page 1087

Mode Configuration Version 6 Support

Mode Configuration version 6 is now supported for more attributes (as described in an IETF draft submission).

Xauth Version 6 Support

Cisco IOS has been enhanced to support version 6 of Xauth. Xauth for user authentication is based on an IETF draft submission.

IKE DPD

The client implements a new keepalives scheme—IKE DPD.

DPD allows two IPSec peers to determine whether the other is still "alive" during the lifetime of a VPN connection. DPD is useful because a host may reboot, or the dialup link of a remote user may disconnect without notifying the peer that the VPN connection has gone away. When an IPSec host determines that a VPN connection no longer exists, the host can notify a user, attempt to switch to another IPSec host, or clean up valuable resources that were allocated for the peer that no longer exists.

A Cisco IOS VPN device can be configured to send and reply to DPD messages. DPD messages are sent if no other traffic is being passed through the VPN tunnel. If a configured amount of time has lapsed since the last inbound data was received, DPD will send a message ("DPD R-U-THERE") the next time it sends outbound IPSec data to the peer. DPD messages are unidirectional and are automatically sent by Cisco VPN clients. DPD *must* be configured on the router *only* if the router wishes to send DPD messages to the VPN client to determine the health of the client.

Split Tunneling Control

Remote clients can support split tunneling, which enables a client to have intranet and Internet access at the same time. If split tunneling is not configured, the client will direct all traffic through the tunnel, even traffic destined for the Internet.

Initial Contact

If a client is suddenly disconnected, the gateway may not be notified. Consequently, removal of connection information (IKE and IPSec SAs) for that client will not immediately occur. Thus, if the client attempts to reconnect to the gateway again, the gateway will refuse the connection because the previous connection information is still valid.

To avoid such a scenario, a new capability called initial contact has been introduced; it is supported by all Cisco VPN products. If a client or router is connecting to another Cisco gateway for the first time, an initial contact message is sent that tells the receiver to ignore and delete any old connection information that has been maintained for that newly connecting peer. Initial contact ensures that connection attempts are not refused because of SA synchronization problems, which are often identified via invalid security parameter index (SPI) messages and which require devices to have their connections cleared.

Group-Based Policy Control

Policy attributes such as IP addresses, DNS, and split tunnel access can be provided on a per-group or per-user basis.

User-Based Policy Control

Attributes may also be applied on a per-user basis. You can override a group attribute value with an individual user attribute. The attributes are retrieved at the time that user authentication via Xauth occurs. They are then combined with group attributes and applied during Mode Configuration.

From Cisco IOS Release 12.3(4)T forward, attributes can be applied on a per-user basis after the user has been authenticated. These attributes can override any similar group attributes. User-based attributes are available only if RADIUS is used as the database.

Framed-IP-Address

To select the Framed-IP-Address attribute: For CiscoSecure for NT, under the user profile, select the "use this IP address"option under addressing and manually enter the address. (You should check the method of configuring a framed IP address with your own RADIUS server as this will vary.)

Note If a framed IP address is present, and there is also a local pool address configured for the group that the user belongs to, the framed IP address will override the local pool setting.

User-Save-Password

As per the group description, the User-Save-Password attribute can be received in addition to the group variant (Save-Password), but if it is received, it will override the value asserted by the group.

The following is an output example of a RADIUS AV pair for the User-Save-Password attribute:

```
ipsec:user-save-password=1
```

User-Include-Local-LAN

As per the group description, the User-Include-Local-LAN attribute can be received in addition to the group variant (Include-Local-LAN), but if it is received, it will override the value asserted by the group.

The following is an output example of a RADIUS AV pair for the User-Include-Local LAN attribute:

```
ipsec:user-include-local-lan=1
```

User-VPN-Group

The User-VPN-Group attribute is a replacement for the Group-Lock attribute. It allows support for both preshared key and RSA signature authentication mechanisms such as certificates.

If you need to check that the group a user is attempting to connect to is indeed the group the user belongs to, use the User-VPN-Group attribute. The administrator sets this attribute to a string, which is the group that the user belongs to. The group the user belongs to is matched against the VPN group as defined by group name (ID_KEY_ID) for preshared keys or by the OU field of a certificate. If the groups do not match, the client connection is terminated.

This feature works only with AAA RADIUS. Local Xauth authentication must still use the Group-Lock attribute.

The following is an output example of a RADIUS AV pair for the Use-VPN-Group attribute:

```
ipsec:user-vpn-group=cisco
```

Group-Lock

If you are only using pre-shared keys (no certificates or other RSA signature authentication mechanisms) with RADIUS or local AAA you can continue to use the Group-Lock attribute. If you are only using pre-shared keys (no certificates or other RSA signature authentication mechanisms) with RADIUS you can either continue to use the Group-Lock attribute or you can use the new User-VPN-Group attribute.

Caution Do not use the Group-Lock attribute if you are using RSA signature authentication mechanisms such as certificates. Use the User-VPN-Group attribute instead.

Session Monitoring for VPN Group Access

It is possible to mimic the functionality provided by some RADIUS servers for limiting the maximum number of connections to a specific server group and also for limiting the number of simultaneous logins for users in that group. After user-defined thresholds are defined in each VPN group, connections will be denied until counts drop below these thresholds.

If you use a RADIUS server, such as CiscoSecure ACS, it is recommended that you enable this session control on the RADIUS server if the functionality is provided. In this way, usage can be controlled across a number of servers by one central repository. When enabling this feature on the router itself, only connections to groups on that specific device are monitored. Load-sharing scenarios are not accurately accounted for.

To configure session monitoring using command-line interface (CLI), use the **crypto isakmp client configuration group** command and the **max-users** and **max-logins** subcommands.

The following is an output example of RADIUS AV pairs that have been added to the relevant group:

```
ipsec:max-users=1000
ipsec:max-logins=1
```

How to Configure Easy VPN Server

This section includes the following procedures:

- Enabling Policy Lookup via AAA, page 1088 (required)
- Defining Group Policy Information for Mode Configuration Push, page 1089 (required)
- Enabling VPN Session Monitoring, page 1092 (optional)
- Verifying a VPN Session, page 1093 (optional)
- Applying Mode Configuration and Xauth, page 1094 (required)
- Enabling Reverse Route Injection for the Client, page 1095 (optional)
- Enabling IKE Dead Peer Detection, page 1096 (optional)
- Configuring RADIUS Server Support, page 1097 (optional)
- Verifying Easy VPN Server, page 1097 (optional)

Enabling Policy Lookup via AAA

To enable policy lookup via AAA, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication password-prompt** *text-string*
5. **aaa authentication username prompt** *text-string*
6. **aaa authentication login** [*list-name method1*] [*method2...*]
7. **aaa authorization network** *list-name* **local group radius**
8. **username** *name* **password** *encryption-type encrypted-password*

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router (config)# aaa new-model`	Enables AAA.
Step 4	**aaa authentication password-prompt** *text-string* **Example:** `Router (config)# aaa authentication password-prompt "Enter your password now:"`	(Optional) Changes the text displayed when users are prompted for a password.
Step 5	**aaa authentication username-prompt** *text-string* **Example:** `Router (oonfig)# aaa authentication username-prompt "Enter your name here:"`	(Optional) Changes the text displayed when users are prompted to enter a username.

	Command	Purpose
Step 6	**aaa authentication login** [*list-name method1*] [*method2...*] **Example:** `Router (config)# aaa authentication login userlist local group radius`	Sets AAA authentication at login. • A local and RADIUS server may be used together and will be tried in order. **Note** This command must be enabled to enforce Xauth.
Step 7	**aaa authorization network** *list-name* **local group radius** **Example:** `Router (config)# aaa authorization network grouplist local group radius`	Enables group policy lookup. • A local and RADIUS server may be used together and will be tried in order.
Step 8	**username** *name* **password** *encryption-type encrypted-password* **Example:** `Router (config)# username server_r password 7 121F0A18`	(Optional) Defines local users for Xauth if RADIUS or TACACS+ is not used. **Note** Use this command only if no external validation repository will be used.

Defining Group Policy Information for Mode Configuration Push

Although users can belong to only one group per connection, they may belong to specific groups with different policy requirements. Thus, users may decide to connect to the client using a different group ID by changing their client profile on the VPN device. To define the policy attributes that are pushed to the client via Mode Configuration, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp client configuration group** {*group-name* | **default**}
4. **key** *name*
5. **dns** *primary-server secondary-server*
6. **wins** *primary-server secondary-server*
7. **domain** *name*
8. **pool** *name*
9. **acl** *number*
10. **split-dns** *domain-name*
11. **access-restrict** {*interface-name*}
12. **firewall are-u-there**
13. **group-lock**
14. **include-local-lan**

15. **save-password**
16. **backup-gateway**
17. **pfs**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp client configuration group** {*group-name* | **default**} **Example:** `Router (config)# crypto isakmp client configuration group group1`	Specifies the policy profile of the group that will be defined and enters Internet Security Association Key Management Protocol (ISAKMP) group configuration mode. • If no specific group matches and a default group is defined, users will automatically be given the policy of a default group.
Step 4	**key** *name* **Example:** `Router (config-isakmp-group)# key group1`	Specifies the IKE preshared key for group policy attribute definition. **Note** This command *must* be enabled if the client identifies itself with a preshared key.
Step 5	**dns** *primary-server secondary-server* **Example:** `Router (config-isakmp-group)# dns 10.2.2.2 10.3.3.3`	(Optional) Specifies the primary and secondary DNS servers for the group.
Step 6	**wins** *primary-server secondary-server* **Example:** `Router (config-isakmp-group)# wins 10.10.10.10 10.12.12.12`	(Optional) Specifies the primary and secondary WINS servers for the group.
Step 7	**domain** *name* **Example:** `Router (config-isakmp-group)# domain domain.com`	(Optional) Specifies the DNS domain to which a group belongs.
Step 8	**pool** *name* **Example:** `Router (config-isakmp-group)# pool green`	Defines a local pool address. • Although a user must define at least one pool name, a separate pool may be defined for each group policy. **Note** This command *must* be defined and refer to a valid IP local pool address or the client connection will fail

	Command	Purpose
Step 9	**acl** *number* **Example:** `Router (config-isakmp-group)# acl 199`	(Optional) Configures split tunneling. • The *number* argument specifies a group of access control list (ACL) rules that represent protected subnets for split tunneling purposes.
Step 10	**split-dns** *domain-name* **Example:** `Router (config-isakmp-group)# split-dns green.com`	Specifies a domain name that must be tunneled or resolved to the private network.
Step 11	**access-restrict** {*interface-name*} **Example:** `Router (config-isakmp-group)# access-restrict fastethernet0/0`	Restricts clients in a group to an interface.
Step 12	**firewall are-u-there** **Example:** `Router (config-isakmp-group)# firewall are-u-there`	(Optional) Adds the firewall are-u-there attribute to the server group if your PC is running the Black Ice or Zone Alarm personal firewalls.
Step 13	**group-lock** **Example:** `Router (config-isakmp-group)# group-lock`	Enforces the group lock feature.
Step 14	**include-local-lan** **Example:** `Router (config-isakmp-group)# include-local-lan`	(Optional) Configures the Include-Local-LAN attribute to allow a nonsplit-tunneling connection to access the local subnetwork at the same time as the client.
Step 15	**save-password** **Example:** `Router (config-isakmp-group)# save-password`	(Optional) Saves your Xauth password locally on your PC.

	Command	Purpose
Step 16	**backup-gateway** **Example:** `Router (config-isakmp-group)# backup gateway`	(Optional) Rather than have backup gateways added to client configurations manually, it is possible to have the server "push down" a list of backup gateways to the client device. • These gateways are tried in order in the case of a failure of the previous gateway. The gateways may be specified using IP addresses or host names.
Step 17	**pfs** **Example:** `Router (config-isakmp-group)# pfs`	(Optional) Notifies the client of the central-site policy regarding whether PFS is required for any IPSec SA. • Because the client device does not have a user interface option to enable or disable PFS negotiation, the server will notify the client device of the central site policy using this parameter. The Diffie-Hellman (D-H) group that is proposed for PFS will be the same that was negotiated in Phase 1 of the IKE negotiation.

Enabling VPN Session Monitoring

If you wish to set restrictions on the maximum number of connections to the router per VPN group and the maximum number of simultaneous logins per user, add the following attributes to the VPN group.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp client configuration group** *group-name*
4. **max-logins** *number-of-logins*
5. **max-users** *number-of-users*

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command	Purpose
Step 3	**crypto isakmp client configuration group** *group-name* **Example:** `Router (config-isakmp-group)# crypto isakmp client configuration group group1`	Specifies the policy profile of the group that will be defined and enters ISAKMP group configuration mode. • *group-name*—Group definition that identifies which policy is enforced for users.
Step 4	**max-logins** *number-of-logins* **Example:** `Router (config-isakmp-group)# max-logins 10`	(Optional) Limits the number of simultaneous logins for users in a specific server group.
Step 5	**max-users** *number-of-users* **Example:** `Router (config)# max-users 1000`	(Optional) Limits the number of connections to a specific server group.

Verifying a VPN Session

To verify a VPN session, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show crypto session group**
3. **show crypto session summary**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto session group** **Example:** `Router# show crypto session group`	Displays groups that are currently active on the VPN device.
Step 3	**show crypto session summary** **Example:** `Router# show crypto session summary`	Displays groups that are currently active on the VPN device and the users that are connected for each of those groups.

Applying Mode Configuration and Xauth

Mode Configuration and Xauth must be applied to a crypto map to be enforced. To apply Mode Configuration and Xauth to a crypto map, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *tag* **client configuration address** [**initiate** | **respond**]
4. **crypto map** *map-name* **isakmp authorization list** *list-name*
5. **crypto map** *map-name* **client authentication list** *list-name*

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *tag* **client configuration address** [**initiate** | **respond**] **Example:** `Router (config)# crypto map dyn client configuration address initiate`	Configures the router to initiate or reply to Mode Configuration requests. **Note** Cisco clients require the **respond** keyword to be used; however, if the Cisco Secure VPN Client 1.x is used, the **initiate** keyword must be used; **initiate** and **respond** keywords may be used simultaneously.
Step 4	**crypto map** *map-name* **isakmp authorization list** *list-name* **Example:** `Router (config)# crypto map ikessaaamap isakmp authorization list ikessaaalist`	Enables IKE querying for group policy when requested by the client. • The *list-name* argument is used by AAA to determine which storage source is used to find the policy (local or RADIUS) as defined in the **aaa authorization network** command.
Step 5	**crypto map** *map-name* **client authentication list** *list-name* **Example:** `Router (config)# crypto map xauthmap client authentication list xauthlist`	Enforces Xauth. • The *list-name* argument is used to determine the appropriate username and password storage location (local or RADIUS) as defined in the **aaa authentication login** command.

Enabling Reverse Route Injection for the Client

To enable RRI on the crypto map (static or dynamic) for VPN client support, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto dynamic** *map-name seq-num*

   or

   crypto map *map-name seq-num* **ipsec-isakmp**
4. **set peer** *ip-address*
5. **set transform-set** *transform-set-name*
6. **reverse-route**
7. **match-address**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto dynamic** *map-name seq-num* or **crypto map** *map-name seq-num* **ipsec-isakmp** **Example:** `Router (config)# crypto dynamic mymap 10` or `Router (config)# crypto map yourmap 15 ipsec-isakmp`	Creates a dynamic crypto map entry and enters crypto map configuration mode. or Adds a dynamic crypto map set to a static crypto map set and enters crypto map configuration mode.
Step 4	**set peer** *ip-address* **Example:** `Router (config-crypto-map)# set peer 10.20.20.20`	Specifies an IPSec peer IP address in a crypto map entry. • This step is optional when configuring dynamic crypto map entries.

	Command	Purpose
Step 5	**set transform-set** *transform-set-name* **Example:** `Router (config-crypto-map)# set transform-set dessha`	Specifies which transform sets are allowed for the crypto map entry. • Lists multiple transform sets in order of priority (highest priority first). **Note** This list is the only configuration statement required in dynamic crypto map entries.
Step 6	**reverse-route** **Example:** `Router (config-crypto-map)# reverse-route`	Creates source proxy information.
Step 7	**match address** **Example:** `Router (config-crypto-map)# match address`	Specifies an extended access list for a crypto map entry. • This step is optional when configuring dynamic crypto map entries.

Enabling IKE Dead Peer Detection

To enable a Cisco IOS VPN gateway (instead of the client) to send IKE DPD messages, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp keepalive** *secs retries*

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp keepalive** *secs retries* **Example:** `Router (config)# crypto isakmp keepalive 20 10`	Allows the gateway to send DPD messages to the router. • The *secs* argument specifies the number of seconds between DPD messages (the range is from 1 to 3600 seconds); the *retries* argument specifies the number of seconds between retries if DPD messages fail (the range is from 2 to 60 seconds).

Configuring RADIUS Server Support

To configure access to the server and allow the Cisco IOS VPN device to send requests to the server, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius server host** *ip-address* [**auth-port** *port-number*] [**acct-port** *port-number*] [**key** *string*]

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**radius server host** *ip-address* **[auth-port** *port-number*] [**acct-port** *port-number*] [**key** *string*] **Example:** `Router (config)# radius server host 192.168.1.1. auth-port 1645 acct-port 1646 key XXXX`	Specifies a RADIUS server host. **Note** This step is required if you choose to store group policy information in a RADIUS server.

Verifying Easy VPN Server

To verify your configurations for this feature, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show crypto map** [**interface** *interface* **|** **tag** *map-name*]

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto map** [**interface** *interface* | **tag** *map-name*] **Example:** `Router# show crypto map interface ethernet 0`	Displays the crypto map configuration.

Configuration Examples for Easy VPN Server

This section provides the following configuration examples:

- Configuring Cisco IOS for Easy VPN Server: Example, page 1098
- RADIUS Group Profile with IPSec AV Pairs: Example, page 1100
- RADIUS User Profile with IPSec AV Pairs: Example, page 1100
- Backup Gateway with Maximum Logins and Maximum Users: Example, page 1100

Configuring Cisco IOS for Easy VPN Server: Example

The following example shows how to define group policy information locally for mode configuration. In this example, a group name is named “cisco” and another group name is named “default.” The policy is enforced for all users who do not offer a group name that matches “cisco.”

```
! Enable policy look-up via AAA. For authentication and authorization, send requests to
! RADIUS first, then try local policy.
aaa new-model
aaa authentication login userlist group radius local
aaa authorization network grouplist group radius local
enable password XXXX
!
username cisco password 0 cisco
clock timezone PST -8
ip subnet-zero
! Configure IKE policies, which are assessed in order so that the first policy that
! matches the client’s proposal will be used.
crypto isakmp policy 1
 group 2
!
crypto isakmp policy 3
 hash md5
 authentication pre-share
 group 2
crypto isakmp identity hostname
!
! Define “cisco” group policy information for mode config push.
crypto isakmp client configuration group cisco
 key cisco
```

```
 dns 10.2.2.2 10.2.2.3
 wins 10.6.6.6
 domain cisco.com
 pool green
 acl 199
! Define default group policy for mode config push.
crypto isakmp client configuration group default
 key cisco
 dns 10.2.2.2 10.3.2.3
 pool green
 acl 199
!
!
crypto ipsec transform-set dessha esp-des esp-sha-hmac
!
crypto dynamic-map mode 1
 set transform-set dessha
!
! Apply mode config and xauth to crypto map "mode." The list names that are defined here
! must match the list names that are defined in the AAA section of the config.
crypto map mode client authentication list userlist
crypto map mode isakmp authorization list grouplist
crypto map mode client configuration address respond
crypto map mode 1 ipsec-isakmp dynamic mode
!
!
controller ISA 1/1
!
!
interface FastEthernet0/0
 ip address 10.6.1.8 255.255.0.0
 ip route-cache
 ip mroute-cache
 duplex auto
 speed auto
 crypto map mode
!
interface FastEthernet0/1
 ip address 192.168.1.28 255.255.255.0
 no ip route-cache
 no ip mroute-cache
 duplex auto
 speed auto
! Specify IP address pools for internal IP address allocation to clients.
ip local pool green 192.168.2.1 192.168.2.10
ip classless
ip route 0.0.0.0 0.0.0.0 5.6.0.1
!
! Define access lists for each subnet that should be protected.
access-list 199 permit ip 192.168.1.0 0.0.0.255 any
access-list 199 permit ip 192.168.3.0 0.0.0.255 any
!
! Specify a RADIUS server host and configure access to the server.
radius-server host 192.168.1.1 auth-port 1645 acct-port 1646 key XXXXX
radius-server retransmit 3
!
!
line con 0
 exec-timeout 0 0
 length 25
 transport input none
line aux 0
line vty 5 15
!
```

RADIUS Group Profile with IPSec AV Pairs: Example

The following is an example of a standard RADIUS group profile that includes RADIUS IPSec AV pairs. To get the group authorization attributes, “cisco” must be used as the password.

```
client_r Password = "cisco"
 Service-Type = Outbound

 cisco-avpair = "ipsec:tunnel-type*ESP"
 cisco-avpair = "ipsec:key-exchange=ike"
 cisco-avpair = "ipsec:tunnel-password=lab"
 cisco-avpair = "ipsec:addr-pool=pool1"
 cisco-avpair = "ipsec:default-domain=cisco"
 cisco-avpair = "ipsec:inacl=101"
 cisco-avpair = “ipsec:access-restrict=fastethernet 0/0”
 cisco-avpair = “ipsec:group-lock=1”
 cisco-avpair = "ipsec:dns-servers=10.1.1.1 10.2.2.2"
 cisco-avpair = “ipsec:firewall=1”
 cisco-avpair = “ipsec:include-local-lan=1”
 cisco-avpair = “ipsec:save-password=1”
 cisco-avpair = "ipsec:wins-servers=3.3.3.3 4.4.4.4"
 cisco-avpair = “ipsec:split-dns=green.com”
 cisoc-avpair = “ipsec:ipsec-backup-gateway=1.1.1.1”
 cisoc-avpair = “ipsec:ipsec-backup-gateway=2.1.1.2”
 cisoc-avpair = “ipsec:pfs=1”
```

RADIUS User Profile with IPSec AV Pairs: Example

The following is an example of a standard RADIUS user profile that includes RADIUS IPSec AV pairs. These user attributes will be obtained during Xauth.

```
ualluall Password = "uall1234"
        cisco-avpair = "ipsec:user-vpn-group=unity"
        cisco-avpair = "ipsec:user-include-local-lan=1"
        cisco-avpair = "ipsec:user-save-password=1"
        Framed-IP-Address = 10.10.10.10
```

Backup Gateway with Maximum Logins and Maximum Users: Example

The following example shows that two backup gateways have been configured, that the maximum users have been set to 250, and that maximum logins have been set to 2:

crypto isakmp client configuration group sdm

```
 key 6 RMZPPMRQMSdiZNJg`EBbCWTKSTi\d[
 pool POOL1
 acl 150
 backup-gateway 172.12.12.12
 backup-gateway 172.12.12.13
 backup-gateway 172.12.12.14
 backup-gateway 172.12.12.130
 backup-gateway 172.12.12.131
 max-users 250
 max-logins 2
```

Additional References

The following sections provide references related to Easy VPN Server.

Related Documents

Related Topic	Document Title
Configuring a router as a VPN client	*Easy VPN Remote Enhancements*, Cisco IOS Release 12.3(2)T feature module
General information on IPSec and VPN	Refer to the following information in the product literature and in IP technical tips sections on Cisco.com: • *Cisco IOS Security Configuration Guide* • *Cisco IOS Security Command Reference*, Release 12.3 T • *An Introduction to IP Security (IPSec) Encryption* • *Deploying IPSec* • *Certificate Authority Support for IPSec Overview* • *Cisco Secure VPN Client* • *IPSec VPN High Availability Enhancements*, Cisco IOS Release 12.2(8)T feature module
IPSec Protocol options and attributes	"Configuring Internet Key Exchange Security Protocol" chapter in the *Cisco IOS Security Configuration Guide*
RRI	*IPSec VPN High Availability Enhancements*, Cisco IOS Release 12.2(8)T feature module

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

New Commands

- **access-restrict**
- **acl (ISAKMP)**
- **backup-gateway**
- **dns**
- **domain (isakmp-group)**
- **firewall are-u-there**
- **include-local-lan**
- **include-local-lan**
- **key (isakmp-group)**

- **max-logins**
- **max-users**
- **pfs**
- **pool (isakmp-group)**
- **save-password**
- **show crypto session group**
- **show crypto session summary**
- **split-dns**
- **wins**

Modified Commands

- **crypto isakmp client configuration group**

Glossary

AAA—authentication, authorization, and accounting. Framework of security services that provides the method for identifying users (authentication), for remote access control (authorization), and for collecting and sending security server information used for billing, auditing, and reporting (accounting).

aggressive mode (AM)—Mode during Internet Key Exchange negotiation. Compared to main mode (MM), AM eliminates several steps, which makes it faster but less secure than MM. Cisco IOS software will respond in aggressive mode to an Internet Key Exchange (IKE) peer that initiates aggressive mode.

AV pair—attribute-value pair. Additional authentication and authorization information in the following format: Cisco:AVPair="protocol:attribute=value".

IKE—Internet Key Exchange. Hybrid protocol that implements Oakley key exchange and Skeme key exchange inside the ISAKMP framework. Although IKE can be used with other protocols, its initial implementation is with IPSec. IKE provides authentication of the IPSec peers, negotiates IPSec keys, and negotiates IPSec security associations.

IPSec—IP Security Protocol. Framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer. IPSec uses IKE to handle negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPSec. IPSec can be used to protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

ISAKMP—Internet Security Association Key Management Protocol. Protocol framework that defines payload formats, the mechanics of implementing a key exchange protocol, and the negotiation of a security association.

MM—main mode. Mode that is slower than aggressive mode but more secure and more flexible than aggressive mode because it can offer an IKE peer more security proposals. The default action for IKE authentication (Rivest, Shamir, and Adelman signature (rsa-sig), RSA encryption (rsa-encr), or preshared) is to initiate main mode.

reverse route injection (RRI)—Simplified network design for VPNs on which there is a requirement for redundancy or load balancing. RRI works with both dynamic and static crypto maps.

In the dynamic case, as remote peers establish IPSec security associations with an RRI enabled router, a static route is created for each subnet or host protected by that remote peer. For static crypto maps, a static route is created for each destination of an extended access-list rule.

SA—security association. Description of how two or more entities will utilize security services to communicate securely. For example, an IPSec SA defines the encryption algorithm (if used), the authentication algorithm, and the shared session key to be used during the IPSec connection.

Both IPSec and IKE require and use SAs to identify the parameters of their connections. IKE can negotiate and establish its own SA. The IPSec SA is established either by IKE or by manual user configuration.

VPN—Virtual Private Network. Framework that consists of multiple peers transmitting private data securely to one another over an otherwise public infrastructure. In this framework, inbound and outbound network traffic is protected using protocols that tunnel and encrypt all data. This framework permits networks to extend beyond their local topology, while remote users are provided with the appearance and functionality of a direct network connection.

Note Refer to *Internetworking Terms and Acronyms* for terms not included in this glossary.

Invalid Security Parameter Index Recovery

When an invalid security parameter index error (shown as “Invalid SPI”) occurs in IP Security (IPSec) packet processing, the Invalid Security Parameter Index Recovery feature allows for an Internet Key Exchange (IKE) security association (SA) to be established. The “IKE” module sends notification of the “Invalid SPI” error to the originating IPSec peer so that Security Association Databases (SADBs) can be resynchronized and successful packet processing can be resumed.

Feature History for Invalid Security Parameter Index Recovery

Release	Modification
12.3(2)T	This feature was introduced.
12.2(18)SXE	This feature was integrated into Cisco IOS Release 12.2(18)SXE.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for Invalid Security Parameter Index Recovery, page 1105
- Restrictions for Invalid Security Parameter Index Recovery, page 1106
- Information About Invalid Security Parameter Index Recovery, page 1106
- How to Configure Invalid Security Parameter Index Recovery, page 1107
- Configuration Examples for Invalid Security Parameter Index Recovery, page 1114
- Additional References, page 1119
- Command Reference, page 1121

Prerequisites for Invalid Security Parameter Index Recovery

Before configuring the Invalid Security Parameter Index Recovery feature, you must have enabled Internet Key Exchange (IKE) and IPSec on your router.

Restrictions for Invalid Security Parameter Index Recovery

If an IKE SA is being initiated to notify an IPSec peer of an "Invalid SPI" error, there is the risk that a denial-of-service (DoS) attack can occur. The Invalid Security Parameter Index Recovery feature has a built-in mechanism to minimize such a risk, but because there is a risk, the Invalid Security Parameter Index Recovery feature is not enabled by default. You must enable the command using command-line interface (CLI).

Information About Invalid Security Parameter Index Recovery

To use the Invalid Security Parameter Index Recovery feature, you should understand the following concept.

- How the Invalid Security Parameter Index Recovery Feature Works, page 1106

How the Invalid Security Parameter Index Recovery Feature Works

An IPSec "black hole" occurs when one IPSec peer "dies" (for example, a peer can "die" if a reboot occurs or if an IPSec peer somehow gets reset). Because one of the peers (the receiving peer) is completely reset, it loses its IKE SA with the other peer. Generally, when an IPSec peer receives a packet for which it cannot find an SA, it tries to send an IKE "INVALID SPI NOTIFY" message to the data originator. This notification is sent using the IKE SA. If there is no IKE SA available, the receiving peer drops the packet.

Note A single security association (SA) has only two peers. However, a SADB can have multiple SAs, whereby each SA has an association with a different peer.

When an invalid security parameter index (SPI) is encountered, the Invalid Security Parameter Index feature provides for the setting up of an IKE SA with the originator of the data, and the IKE "INVALID SPI NOTIFY" message is sent. The peer that originated the data "sees" the "INVALID SPI NOTIFY" message and deletes the IPSec SA that has the invalid SPI. If there is further traffic from the originating peer, there will not be any IPSec SAs, and new SAs will be set up. Traffic will flow again. The default behavior (that is, without configuring the Invalid Security Parameter Index Recovery feature) is that the data packet that caused the invalid SPI error is dropped. The originating peer keeps on sending the data using the IPSec SA that has the invalid SPI, and the receiving peer keeps dropping the traffic (thus creating the "black hole").

The IPSec module uses the IKE module to send an IKE "INVALID SPI NOTIFY" message to the other peer. Once the invalid SPI recovery is in place, there should not be any significant dropping of packets although the IPSec SA setup can itself result in the dropping of a few packets.

To configure your router for the Invalid Security Parameter Index Recovery feature, use the **crypto isakmp invalid-spi-recovery** command. The IKE SA will not be initiated unless you have configured this command.

How to Configure Invalid Security Parameter Index Recovery

This section contains the following procedure.

- Configuring Invalid Security Parameter Index Recovery, page 1107

Configuring Invalid Security Parameter Index Recovery

To configure the Invalid Security Parameter Index Recovery feature, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp invalid-spi-recovery**

DETAILED STEPS

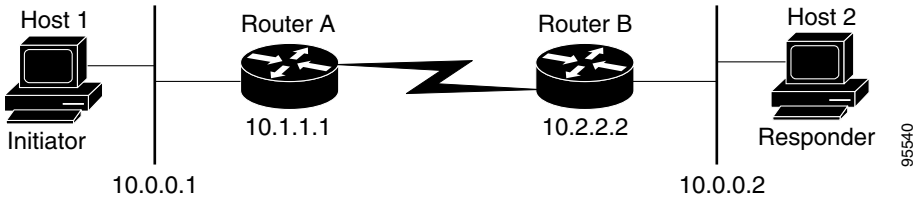
	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp invalid-spi-recovery** **Example:** `Router (config)# crypto isakmp invalid-spi-recovery`	Initiates the IKE module process whereby the IKE module notifies the receiving peer that an "Invalid SPI" error has occurred.

Verifying an Invalid Security Parameter Index Recovery Configuration

To determine the status of the IPSec SA for traffic between two peers, you can use the **show crypto ipsec sa** command. If the IPSec SA is available on one peer and not on the other, there is a "black hole" situation, in which case you will see the invalid SPI errors being logged for the receiving peer. If you turn console logging on or check the syslog server, you will see that these errors are also being logged.

Figure 77 shows the topology of a typical preshared configuration setup. Host 1 is the initiating peer (initiator), and Host 2 is the receiving peer (responder).

Figure 77 Preshared Configuration Topology



To verify the preshared configuration, perform the following steps.

SUMMARY STEPS

1. **Initiate the IKE and IPSec SAs between Host 1 and Host 2**
2. **Clear the IKE and IPSec SAs on Router B**
3. **Send traffic from Host 1 to Host 2 and ensure that IKE and IPSec SAs are correctly established**
4. **Check for an invalid SPI message on Router B**

DETAILED STEPS

Step 1 Initiate the IKE and IPSec SAs between Host 1 and Host 2

Router A

```
Router# show crypto isakmp sa

  f_vrf/i_vrf   dst            src             state       conn-id slot
       / 10.2.0.1              10.2.0.0        QM_IDLE               1       0
```

Router B

```
Router# show crypto isakmp sa

  f_vrf/i_vrf   dst            src             state       conn-id slot
       /        10.2.0.1       10.2.0.0        QM_IDLE               1       0
```

Router A

```
Router# show crypto ipsec sa interface fastethernet0/0

interface: FastEthernet0/0
    Crypto map tag: testtag1, local addr. 10.2.0.0

   protected vrf:
   local  ident (addr/mask/prot/port): (10.0.0.1/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (10.0.0.3/255.255.255.255/0/0)
   current_peer: 10.0.0.2:500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 10, #pkts encrypt: 10, #pkts digest: 10
    #pkts decaps: 10, #pkts decrypt: 10, #pkts verify: 10
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0
```

```
    local crypto endpt.: 10.2.0.0, remote crypto endpt.: 10.2.0.1
    path mtu 1500, media mtu 1500
    current outbound spi: 7AA69CB7

    inbound esp sas:
     spi: 0x249C5062(614223970)
       transform: esp-des esp-sha-hmac ,
       in use settings ={Tunnel, }
       slot: 0, conn id: 5123, flow_id: 1, crypto map: testtag1
       crypto engine type: Hardware
       sa timing: remaining key lifetime (k/sec): (4537831/3595)
       IV size: 8 bytes
       replay detection support: Y

    inbound ah sas:
     spi: 0xB16D1587(2976716167)
       transform: ah-sha-hmac ,
       in use settings ={Tunnel, }
       slot: 0, conn id: 5121, flow_id: 1, crypto map: testtag1
       crypto engine type: Hardware
       sa timing: remaining key lifetime (k/sec): (4537831/3595)
       replay detection support: Y

    inbound pcp sas:

    outbound esp sas:
     spi: 0x7AA69CB7(2057739447)
       transform: esp-des esp-sha-hmac ,
       in use settings ={Tunnel, }
       slot: 0, conn id: 5124, flow_id: 2, crypto map: testtag1
       crypto engine type: Hardware
       sa timing: remaining key lifetime (k/sec): (4537835/3595)
       IV size: 8 bytes
       replay detection support: Y

    outbound ah sas:
     spi: 0x1214F0D(18960141)
       transform: ah-sha-hmac ,
       in use settings ={Tunnel, }
       slot: 0, conn id: 5122, flow_id: 2, crypto map: testtag1
       crypto engine type: Hardware
       sa timing: remaining key lifetime (k/sec): (4537835/3594)
       replay detection support: Y

    outbound pcp sas:
```

Router B

```
Router# show crypto ipsec sa interface ethernet1/0

interface: Ethernet1/0
    Crypto map tag: testtag1, local addr. 10.2.0.1

   protected vrf:
   local  ident (addr/mask/prot/port): (10.0.0.3/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (10.0.0.1/255.255.255.255/0/0)
   current_peer: 10.2.0.0:500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 10, #pkts encrypt: 10, #pkts digest: 10
    #pkts decaps: 10, #pkts decrypt: 10, #pkts verify: 10
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
```

```
#send errors 0, #recv errors 0

 local crypto endpt.: 10.2.0.1, remote crypto endpt.: 10.2.0.0
 path mtu 1500, media mtu 1500
 current outbound spi: 249C5062

 inbound esp sas:
  spi: 0x7AA69CB7(2057739447)
    transform: esp-des esp-sha-hmac ,
    in use settings ={Tunnel, }
    slot: 0, conn id: 5123, flow_id: 1, crypto map: testtag1
    crypto engine type: Hardware
    sa timing: remaining key lifetime (k/sec): (4421281/3593)
    IV size: 8 bytes
    replay detection support: Y

 inbound ah sas:
  spi: 0x1214F0D(18960141)
    transform: ah-sha-hmac ,
    in use settings ={Tunnel, }
    slot: 0, conn id: 5121, flow_id: 1, crypto map: testtag1
    crypto engine type: Hardware
    sa timing: remaining key lifetime (k/sec): (4421281/3593)
    replay detection support: Y

 inbound pcp sas:

 outbound esp sas:
  spi: 0x249C5062(614223970)
    transform: esp-des esp-sha-hmac ,
    in use settings ={Tunnel, }
    slot: 0, conn id: 5124, flow_id: 2, crypto map: testtag1
    crypto engine type: Hardware
    sa timing: remaining key lifetime (k/sec): (4421285/3593)
    IV size: 8 bytes
    replay detection support: Y

 outbound ah sas:
  spi: 0xB16D1587(2976716167)
    transform: ah-sha-hmac ,
    in use settings ={Tunnel, }
    slot: 0, conn id: 5122, flow_id: 2, crypto map: testtag1
    crypto engine type: Hardware
    sa timing: remaining key lifetime (k/sec): (4421285/3592)
    replay detection support: Y

 outbound pcp sas:
```

Step 2 Clear the IKE and IPSec SAs on Router B

```
Router# clear crypto isakmp

Router# clear crypto sa

Router# show crypto isakmp sa

  f_vrf/i_vrf   dst            src             state       conn-id slot
       /        10.2.0.1       10.2.0.0      MM_NO_STATE           1       0 (deleted)

Router# show crypto ipsec sa

interface: Ethernet1/0
    Crypto map tag: testtag1, local addr. 10.2.0.1
```

```
   protected vrf:
   local  ident (addr/mask/prot/port): (10.0.0.3/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (10.0.0.1/255.255.255.255/0/0)
   current_peer: 10.2.0.0:500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
    #pkts decaps: 0, #pkts decrypt: 0, #pkts verify: 0
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0

     local crypto endpt.: 10.2.0.1, remote crypto endpt.: 10.2.0.0
     path mtu 1500, media mtu 1500
     current outbound spi: 0

     inbound esp sas:

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:

     outbound ah sas:

     outbound pcp sas:
```

Step 3 Send traffic from Host 1 to Host 2 and ensure that new IKE and IPSec SAs are correctly established

```
ping
Protocol [ip]: ip
Target IP address: 10.0.0.3
Repeat count [5]: 30
Datagram size [100]: 100
Timeout in seconds [2]:
Extended commands [n]: no
Sweep range of sizes [n]: n
Type escape sequence to abort.
Sending 30, 100-byte ICMP Echos to 10.0.0.3, timeout is 2 seconds:
..!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 93 percent (28/30), round-trip min/avg/max = 1/3/8 ms

Router# show crypto isakmp sa

  f_vrf/i_vrf   dst            src             state       conn-id slot
       /        10.2.0.0       10.2.0.1     QM_IDLE              3       0
       /        10.2.0.1       10.2.0.0      MM_NO_STATE          1       0 (deleted)

Router# show crypto ipsec sa

interface: Ethernet1/0
    Crypto map tag: testtag1, local addr. 10.2.0.1

   protected vrf:
   local  ident (addr/mask/prot/port): (10.0.0.3/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (10.0.0.1/255.255.255.255/0/0)
   current_peer: 10.2.0.0:500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 28, #pkts encrypt: 28, #pkts digest: 28
    #pkts decaps: 28, #pkts decrypt: 28, #pkts verify: 28
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
```

```
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0

     local crypto endpt.: 10.2.0.1, remote crypto endpt.: 10.2.0.0
     path mtu 1500, media mtu 1500
     current outbound spi: D763771F

     inbound esp sas:
      spi: 0xE7AB4256(3886760534)
        transform: esp-des esp-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5127, flow_id: 3, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4502463/3596)
        IV size: 8 bytes
        replay detection support: Y

     inbound ah sas:
      spi: 0xF9205CED(4179647725)
        transform: ah-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5125, flow_id: 3, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4502463/3596)
        replay detection support: Y

     inbound pcp sas:

     outbound esp sas:
      spi: 0xD763771F(3613619999)
        transform: esp-des esp-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5128, flow_id: 4, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4502468/3596)
        IV size: 8 bytes
        replay detection support: Y

     outbound ah sas:
      spi: 0xEB95406F(3952427119)
        transform: ah-sha-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 5126, flow_id: 4, crypto map: testtag1
        crypto engine type: Hardware
        sa timing: remaining key lifetime (k/sec): (4502468/3595)
        replay detection support: Y

     outbound pcp sas:

Router# show crypto isakmp sa

  f_vrf/i_vrf   dst            src             state       conn-id slot
       /        10.2.0.1       10.2.0.0      MM_NO_STATE           1       0 (deleted)
       /        10.2.0.0       10.2.0.1      QM_IDLE               2       0

Router# show crypto isakmp sa

  f_vrf/i_vrf   dst            src             state       conn-id slot
       /        10.2.0.0       10.2.0.1       QM_IDLE               3       0
       /        10.2.01        10.2.0.0        MM_NO_STATE           1       0 (deleted)
```

Step 4 Check for an invalid SPI message on Router B

```
Router# show logging
```

```
Syslog logging: enabled (10 messages dropped, 13 messages rate-limited, 0 flushes, 0
overruns, xml disabled)
    Console logging: disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled
    Buffer logging: level debugging, 43 messages logged, xml disabled
    Logging Exception size (8192 bytes)
    Count and timestamp logging messages: disabled
    Trap logging: level informational, 72 message lines logged

Log Buffer (8000 bytes):

*Mar 24 20:55:45.739: %CRYPTO-4-RECVD_PKT_INV_SPI: decaps: rec'd IPSEC packet has invalid
spi for
        destaddr=2.0.0.2, prot=51, spi=0x1214F0D(18960141), srcaddr=2.0.0.1
*Mar 24 20:55:47.743: IPSEC(validate_proposal_request): proposal part #1,
  (key eng. msg.) INBOUND local= 10.2.0.1, remote= 10.2.0.0,
    local_proxy= 10.0.0.3/255.255.255.255/0/0 (type=1),
    remote_proxy= 10.0.0.1/255.255.255.255/0/0 (type=1),
    protocol= AH, transform= ah-sha-hmac ,
    lifedur= 0s and 0kb,
    spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x2
*Mar 24 20:55:47.743: IPSEC(validate_proposal_request): proposal part #2,
  (key eng. msg.) INBOUND local= 10.2.0.1, remote= 10.2.0.0,
    local_proxy= 10.0.0.3/255.255.255.255/0/0 (type=1),
    remote_proxy= 10.0.0.1/255.255.255.255/0/0 (type=1),
    protocol= ESP, transform= esp-des esp-sha-hmac ,
    lifedur= 0s and 0kb,
    spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x2
*Mar 24 20:55:47.743: IPSEC(kei_proxy): head = testtag1, map->ivrf = , kei->ivrf =
*Mar 24 20:55:47.743: IPSEC(key_engine): got a queue event with 2 kei messages
*Mar 24 20:55:47.743: IPSEC(spi_response): getting spi 4179647725 for SA
        from 10.2.0.1        to 10.2.0.0     for prot 2
*Mar 24 20:55:47.747: IPSEC(spi_response): getting spi 3886760534 for SA
        from 10.2.0.1     to 10.2.0.0        for prot 3
*Mar 24 20:55:48.071: IPSec: Flow_switching Allocated flow for flow_id 939524099
*Mar 24 20:55:48.071: IPSec: Flow_switching Allocated flow for flow_id 939524100
*Mar 24 20:55:48.135: IPSEC(key_engine): got a queue event with 4 kei messages
*Mar 24 20:55:48.135: IPSEC(initialize_sas): ,
  (key eng. msg.) INBOUND local= 10.2.0.1, remote= 10.2.0.0,
    local_proxy= 10.0.0.3/0.0.0.0/0/0 (type=1),
    remote_proxy= 10.0.0.1/0.0.0.0/0/0 (type=1),
    protocol= AH, transform= ah-sha-hmac ,
    lifedur= 3600s and 4608000kb,
    spi= 0xF9205CED(4179647725), conn_id= 939529221, keysize= 0, flags= 0x2
*Mar 24 20:55:48.135: IPSEC(initialize_sas): ,
  (key eng. msg.) OUTBOUND local= 10.2.0.1, remote= 10.2.0.0,
    local_proxy= 10.0.0.3/0.0.0.0/0/0 (type=1),
    remote_proxy= 10.0.0.1/0.0.0.0/0/0 (type=1),
    protocol= AH, transform= ah-sha-hmac ,
    lifedur= 3600s and 4608000kb,
    spi= 0xEB95406F(3952427119), conn_id= 939529222, keysize= 0, flags= 0xA
*Mar 24 20:55:48.135: IPSEC(initialize_sas): ,
  (key eng. msg.) INBOUND local= 2.0.0.2, remote= 2.0.0.1,
    local_proxy= 10.0.0.3/0.0.0.0/0/0 (type=1),
    remote_proxy= 10.0.0.1/0.0.0.0/0/0 (type=1),
    protocol= ESP, transform= esp-des esp-sha-hmac ,
    lifedur= 3600s and 4608000kb,
    spi= 0xE7AB4256(3886760534), conn_id= 939529223, keysize= 0, flags= 0x2
*Mar 24 20:55:48.135: IPSEC(initialize_sas): ,
  (key eng. msg.) OUTBOUND local= 2.0.0.2, remote= 2.0.0.1,
    local_proxy= 10.0.0.3/0.0.0.0/0/0 (type=1),
    remote_proxy= 10.0.0.1/0.0.0.0/0/0 (type=1),
    protocol= ESP, transform= esp-des esp-sha-hmac ,
```

```
    lifedur= 3600s and 4608000kb,
    spi= 0xD763771F(3613619999), conn_id= 939529224, keysize= 0, flags= 0xA
*Mar 24 20:55:48.139: IPSEC(kei_proxy): head = testtag1, map->ivrf = , kei->ivrf =
*Mar 24 20:55:48.139: IPSEC(mtree_add_ident): src 10.0.0.3, dest 10.0.0.1, dest_port 0

*Mar 24 20:55:48.139: IPSEC(create_sa): sa created,
  (sa) sa_dest= 10.2.0.1, sa_prot= 51,
    sa_spi= 0xF9205CED(4179647725),
    sa_trans= ah-sha-hmac , sa_conn_id= 939529221
*Mar 24 20:55:48.139: IPSEC(create_sa): sa created,
  (sa) sa_dest= 10.2.0.1, sa_prot= 51,
    sa_spi= 0xEB95406F(3952427119),
    sa_trans= ah-sha-hmac , sa_conn_id= 939529222
*Mar 24 20:55:48.139: IPSEC(create_sa): sa created,
  (sa) sa_dest= 10.2.0.1, sa_prot= 50,
    sa_spi= 0xE7AB4256(3886760534),
    sa_trans= esp-des esp-sha-hmac , sa_conn_id= 939529223
*Mar 24 20:55:48.139: IPSEC(create_sa): sa created,
  (sa) sa_dest= 10.2.0.0, sa_prot= 50,
    sa_spi= 0xD763771F(3613619999),
    sa_trans= esp-des esp-sha-hmac , sa_conn_id= 939529224
ipseca-72a#
```

Configuration Examples for Invalid Security Parameter Index Recovery

This section provides the following configuration example.

- Invalid Security Parameter Index Recovery: Example, page 1114

Invalid Security Parameter Index Recovery: Example

The following example shows that invalid security parameter index recovery has been configured on Router A and Router B. Figure 77 shows the topology used for this example.

Router 1

```
Router# show running-config

Building configuration...

Current configuration : 2048 bytes
!
version 12.3
no service pad
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
no service password-encryption
service tcp-small-servers
!
hostname ipseca-71a
!
logging queue-limit 100
no logging console
enable secret 5 $1$4GZB$L2YOmnenOCNAu0jgFxebT/
enable password lab
!
clock timezone PST -8
```

```
clock summer-time PDT recurring
ip subnet-zero
!
!
no ip domain lookup
!
ip cef
ip audit notify log
ip audit po max-events 100
mpls ldp logging neighbor-changes
no ftp-server write-enable
!
!
no voice hpi capture buffer
no voice hpi capture destination
!
!
crypto isakmp policy 1
 authentication pre-share
 lifetime 180
crypto isakmp key 0 1234 address 10.2.0.1
crypto isakmp invalid-spi-recovery
!
!
crypto ipsec transform-set auth2 ah-sha-hmac esp-des esp-sha-hmac
!
crypto map testtag1 10 ipsec-isakmp
 set peer 10.0.0.2
 set transform-set auth2
 match address 150
!
!
controller ISA 5/1
!
!
interface FastEthernet0/0
 ip address 10.2.0.1 255.0.0.0
 no ip route-cache cef
 duplex full
 speed 100
 crypto map testtag1
!
interface FastEthernet0/1
 ip address 10.2.2.2 255.0.0.0
 no ip route-cache cef
 duplex auto
 speed auto
!
interface Serial1/0
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 serial restart_delay 0
 clockrate 128000
!
interface Serial1/1
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 serial restart_delay 0
 clockrate 128000
!
```

```
interface Serial1/2
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 serial restart_delay 0
!
interface Serial1/3
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 no keepalive
 serial restart_delay 0
 clockrate 128000
!
ip classless
ip route 10.3.3.3 255.0.0.0 10.2.0.1
no ip http server
no ip http secure-server
!
!
access-list 150 permit ip host 10.0.0.1 host 10.0.0.3
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
!
call rsvp-sync
!
!
mgcp profile default
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 password lab
 login
!
!
end

ipseca-71a#
```

Router 2

```
Router# show running-config

Building configuration...

Current configuration : 2849 bytes
!
version 12.3
no service pad
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
no service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname ipseca-72a
!
```

```
logging queue-limit 100
no logging console
enable secret 5 $1$kKqL$5Th5Qhw1ubDkkK90KWFxi1
enable password lab
!
clock timezone PST -8
clock summer-time PDT recurring
ip subnet-zero
!
!
no ip domain lookup
!
ip cef
ip audit notify log
ip audit po max-events 100
mpls ldp logging neighbor-changes
no ftp-server write-enable
!
!
no voice hpi capture buffer
no voice hpi capture destination
!
!
mta receive maximum-recipients 0
!
!
crypto isakmp policy 1
 authentication pre-share
 lifetime 180
crypto isakmp key 0 1234 address 10.2.0.0
crypto isakmp invalid-spi-recovery
!
!
crypto ipsec transform-set auth2 ah-sha-hmac esp-des esp-sha-hmac
!
crypto map testtag1 10 ipsec-isakmp
 set peer 10.2.0.0
 set transform-set auth2
 match address 150
!
!
controller ISA 5/1
!
!
interface FastEthernet0/0
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface Ethernet1/0
 ip address 10.2.0.1 255.0.0.0
 no ip route-cache cef
 duplex half
 crypto map testtag1
!
interface Ethernet1/1
 ip address 10.3.2.2. 255.0.0.0
 no ip route-cache cef
 duplex half
!
interface Ethernet1/2
 no ip address
```

```
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface Ethernet1/3
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface Ethernet1/4
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface Ethernet1/5
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface Ethernet1/6
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface Ethernet1/7
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 duplex half
!
interface Serial3/0
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 serial restart_delay 0
!
interface Serial3/1
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 serial restart_delay 0
 clockrate 128000
!
interface Serial3/2
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 serial restart_delay 0
!
interface Serial3/3
 no ip address
```

```
 no ip route-cache
 no ip mroute-cache
 shutdown
 no keepalive
 serial restart_delay 0
 clockrate 128000
!
ip classless
ip route 10.0.0.0 255.0.0.0 10.2.0.1
no ip http server
no ip http secure-server
!
!
access-list 150 permit ip host 10.0.0.3 host 10.0.0.1
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
!
call rsvp-sync
!
!
mgcp profile default
!
dial-peer cor custom
!
!
gatekeeper
 shutdown
!
!
line con 0
 exec-timeout 0 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 password lab
 login
!
!
end
```

Additional References

The following sections provide references related to Invalid Security Parameter Index Recovery.

Related Documents

Related Topic	Document Title
Configuring IKE	"Configuring Internet Key Exchange Security Protocol" section of the *Cisco IOS Security Configuration Guide*
Configuring IPSec	"Part 4: IP Security and Encryption" of the *Cisco IOS Security Configuration Guide*
Interface commands	The *Cisco IOS Interface and Hardware Component Command Reference*, Release 12.3

Standards

Standards	Title
This feature has no new or modified standards.	—

MIBs

MIBs	MIBs Link
This feature has no new or modified MIBs.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
This feature has no new or modified RFCs.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **crypto isakmp invalid-spi-recovery**

IP Security VPN Monitoring

The IP Security VPN Monitoring feature provides VPN session monitoring enhancements that will allow you to troubleshoot the Virtual Private Network (VPN) and monitor the end-user interface. Session monitoring enhancements include the following:

- Ability to specify an Internet Key Exchange (IKE) peer description in the configuration file
- Summary listing of crypto session status
- Syslog notification for crypto session up or down status
- Ability to clear both IKE and IP Security (IPSec) security associations (SAs) using one command-line interface (CLI)

Feature History for IP Security VPN Monitoring

Release	Modification
12.3(4)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for IP Security VPN Monitoring, page 1124
- Restrictions for IP Security VPN Monitoring, page 1124
- Information About IPSec VPN Monitoring, page 1124
- How to Configure IP Security VPN Monitoring, page 1126
- Configuration Examples for IP Security VPN Monitoring, page 1128
- Additional References, page 1129
- Command Reference, page 1130

Prerequisites for IP Security VPN Monitoring

- You should be familiar with IPSec and encryption.
- Your router must support IPSec, and before using the IP Security VPN Monitoring feature, you must have configured IPSec on your router.

Restrictions for IP Security VPN Monitoring

- You must be running Cisco IOS k8 or k9 crypto images on your router.

Information About IPSec VPN Monitoring

To troubleshoot the IPSec VPN and monitor the end-user interface, you should understand the following concepts:

- Background: Crypto Sessions, page 1124
- Per-IKE Peer Description, page 1124
- Summary Listing of Crypto Session Status, page 1125
- Syslog Notification for Crypto Session Up or Down Status, page 1125
- IKE and IPSec Security Exchange Clear Command, page 1125

Background: Crypto Sessions

A crypto session is a set of IPSec connections (flows) between two crypto endpoints. If the two crypto endpoints use IKE as the keying protocol, they are IKE peers to each other. Typically, a crypto session consists of one IKE security association (for control traffic) and at least two IPSec security associations (for data traffic—one per each direction). There may be duplicated IKE security associations (SAs) and IPSec SAs or duplicated IKE SAs or IPSec SAs for the same session in the duration of rekeying or because of simultaneous setup requests from both sides.

Per-IKE Peer Description

The Per-IKE Peer Description function allows you to enter a description of your choosing for an IKE peer. (Before Cisco IOS Release 12.3(4)T, you could use only the IP address or fully qualified domain name [FQDN] to identify the peer; there was no way to configure a description string.) The unique peer description, which can include up to 80 characters, can be used whenever you are referencing that particular IKE peer. To add the peer description, use the **description** command.

Note IKE peers that "sit" behind a Network Address Translation (NAT) device cannot be uniquely identified; therefore, they have to share the same peer description.

The primary application of this description field is for monitoring purposes (for example, when using **show** commands or for logging [syslog messages]). The description field is purely informational (for example, it cannot act as a substitute for the peer address or FQDN when defining crypto maps).

Summary Listing of Crypto Session Status

You can get a list of all the active VPN sessions by entering the **show crypto session** command. The listing will include the following:

- Interface
- IKE peer description, if available
- IKE SAs that are associated with the peer by whom the IPSec SAs are created
- IPSec SAs serving the flows of a session

Multiple IKE or IPSec SAs may be established for the same peer (for the same session), in which case IKE peer descriptions will be repeated with different values for the IKE SAs that are associated with the peer and for the IPSec SAs that are serving the flows of the session.

You can also use the **show crypto session detail** variant of this command to obtain more detailed information about the sessions.

Syslog Notification for Crypto Session Up or Down Status

The Syslog Notification for Crypto Session Up or Down Status function provides syslog notification every time the crypto session comes up or goes down.

The following is a sample syslog notification showing that a crypto session is up:

```
%CRYPTO-5-SESSION_STATUS: Crypto session is UP. Peer 10.6.6.1:500 fvrf=name10 ivrf=name20
Description: SJC24-2-VPN-Gateway Id: 10.5.5.2
```

The following is a sample syslog notification showing that a crypto session is down:

```
%CRYPTO-5-SESSION_STATUS: Crypto session is DOWN. Peer 10.6.6.1:500 fvrf=name10
ivrf=name20 Description: SJC24-2-VPN-Gateway Id: 10.5.5.2
```

IKE and IPSec Security Exchange Clear Command

In previous IOS versions, there was no single command to clear both IKE and IPSec connections (that is, SAs). Instead, you had to use the **clear crypto isakmp** command to clear IKE and the **clear crypto ipsec** command to clear IPSec. The new **clear crypto session** command allows you to clear both IKE and IPSec with a single command. To clear a specific crypto session or a subset of all the sessions (for example, a single tunnel to one remote site), you need to provide session-specific parameters, such as a local or remote IP address, a local or remote port, a front door VPN routing and forwarding (FVRF) name, or an inside VRF (IVRF) name. Typically, the remote IP address will be used to specify a single tunnel to be deleted.

If a local IP address is provided as a parameter when you use the **clear crypto session** command, all the sessions (and their IKE SAs and IPSec SAs) that share the IP address as a local crypto endpoint (IKE local address) will be cleared. If you do not provide a parameter when you use the **clear crypto session** command, all IPSec SAs and IKE SAs that are in the router will be deleted.

How to Configure IP Security VPN Monitoring

See the following sections for configuration tasks for this feature. Each task in the list is identified as either required or optional.

- Adding the Description of an IKE Peer, page 1126 (optional)
- Verifying Peer Descriptions, page 1127 (optional)
- Clearing a Crypto Session, page 1127 (optional)

Adding the Description of an IKE Peer

To add the description of an IKE peer to an IPSec VPN session, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp peer** {**ip-address** *ip-address*}
4. **description**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp peer** {**ip-address** *ip-address*} **Example:** `Router (config)# crypto isakmp peer address 10.2.2.9`	Enables an IPSec peer for IKE querying of authentication, authorization, and accounting (AAA) for tunnel attributes in aggressive mode and enters ISAKMP peer configuration mode.
Step 4	**description** **Example:** `Router (config-isakmp-peer)# description connection from site A`	Adds a description for an IKE peer.

Verifying Peer Descriptions

To verify peer descriptions, use the **show crypto isakmp peer** command.

SUMMARY STEPS

1. **enable**
2. **show crypto isakmp peer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto isakmp peer** **Example:** `Router# show crypto isakmp peer`	Displays peer descriptions.

Examples

The following output example verifies that the description “connection from site A” has been added for IKE peer 10.2.2.9:

```
Router# show crypto isakmp peer

Peer: 10.2.2.9 Port: 500
 Description: connection from site A
  flags: PEER_POLICY
```

When the peer at address 10.2.2.9 connects and the session comes up, the syslog status will be shown as follows:

```
%CRYPTO-5-SESSION_STATUS: Crypto tunnel is UP. Peer 10.2.2.9:500 Description: connection
from site A Id: ezvpn
```

Clearing a Crypto Session

To clear a crypto session, use the **clear crypto session** command from the router command line. No configuration statements are required in the configuration file to use this command.

SUMMARY STEPS

1. **enable**
2. **clear crypto session**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear crypto session** **Example:** `Router# clear crypto session`	Deletes crypto sessions (IPSec and IKE SAs).

Configuration Examples for IP Security VPN Monitoring

This section provides the following configuration example:

• show crypto session Command Output: Examples, page 1128

show crypto session Command Output: Examples

The following is sample output for the **show crypto session** output without the **detail** keyword:

```
Router# show crypto session

Crypto session current status

Interface: FastEthernet0/1
Session status: UP-ACTIVE
Peer: 172.0.0.2/500
   IKE SA: local 172.0.0.1/500 remote 172.0.0.2/500 Active
   IPSEC FLOW: permit ip 10.10.10.0/255.255.255.0 10.30.30.0/255.255.255.0
         Active SAs: 2, origin: crypto map
```

The following is sample output using the **show crypto session** command and the **detail** keyword:

```
Router# show crypto session detail

Interface: Tunnel0
Session status: UP-ACTIVE
Peer: 10.1.1.3 port 500 fvrf: (none) ivrf: (none)
      Desc: this is my peer at 10.1.1.3:500 Green
      Phase1_id: 10.1.1.3
  IKE SA: local 10.1.1.4/500 remote 10.1.1.3/500 Active
          Capabilities:(none) connid:3 lifetime:22:03:24
  IPSEC FLOW: permit 47 host 10.1.1.4 host 10.1.1.3
        Active SAs: 0, origin: crypto map
        Inbound:  #pkts dec'ed 0 drop 0 life (KB/Sec) 0/0
        Outbound: #pkts enc'ed 0 drop 0 life (KB/Sec) 0/0
  IPSEC FLOW: permit ip host 10.1.1.4 host 10.1.1.3
        Active SAs: 4, origin: crypto map
        Inbound:  #pkts dec'ed 4 drop 0 life (KB/Sec) 4605665/2949
        Outbound: #pkts enc'ed 4 drop 1 life (KB/Sec) 4605665/2949
```

Additional References

The following sections provide references related to IP Security VPN Monitoring.

Related Documents

Related Topic	Document Title
IP security, encryption, and IKE	"IP Security and Encryption" section of the *Cisco IOS Security Configuration Guide*
Security commands	*Cisco IOS Security Command Reference*, Release 12.3 T

Standards

Standards	Title
No new or modified standards are supported by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

New Commands

- **clear crypto session**
- **description (isakmp peer)**
- **show crypto isakmp peer**
- **show crypto session**

IPSec and Quality of Service

The IPSec and Quality of Service feature allows Cisco IOS quality of service (QoS) policies to be applied to IP Security (IPSec) packet flows on the basis of a QoS group that can be added to the current Internet Security Association and Key Management Protocol (ISAKMP) profile.

Feature History for IPSec and Quality of Service

Release	Modification
12.3(8)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for IPSec and Quality of Service, page 1131
- Restrictions for IPSec and Quality of Service, page 1132
- Information About IPSec and Quality of Service, page 1132
- How to Configure IPSec and Quality of Service, page 1132
- Configuration Examples for IPSec and Quality of Service, page 1134
- Additional References, page 1137
- Command Reference, page 1138

Prerequisites for IPSec and Quality of Service

- You should be familiar with IPSec and the concept of ISAKMP profiles.
- You should be familiar with Cisco IOS QoS.

Restrictions for IPSec and Quality of Service

- This feature can be applied only via the ISAKMP profile. The limit of 128 QoS groups that exists for QoS applications applies to this feature as well.
- You can apply an IPSec QoS group only to outbound service policies.

Information About IPSec and Quality of Service

To configure the IPSec and Quality of Service feature, you should understand the following concept:

- IPSec and Quality of Service Overview, page 1132

IPSec and Quality of Service Overview

The IPSec and Quality of Service feature allows you to apply QoS policies, such as traffic policing and shaping, to IPSec-protected packets by adding a QoS group to ISAKMP profiles. After the QoS group has been added, this group value will be mapped to the same QoS group as defined in QoS class maps. Any current QoS method that makes use of this QoS group tag can be applied to IPSec packet flows. Common groupings of packet flows can have specific policy classes applied by having the IPSec QoS group made available to the QoS mechanism. Marking IPSec flows allows QoS mechanisms to be applied to classes of traffic that could provide support for such things as restricting the amount of bandwidth that is available to specific groups or devices or marking the type of service (ToS) bits on certain flows.

The application of the QoS group is applied at the ISAKMP profile level because it is the profile that can uniquely identify devices through its concept of match identity criteria. These criteria are on the basis of the Internet Key Exchange (IKE) identity that is presented by incoming IKE connections and includes such things as IP address, fully qualified domain name (FQDN), and group (that is, the virtual private network [VPN] remote client grouping). The granularity of the match identity criteria will impose the granularity of the specified QoS policy, for example, to mark all traffic belonging to the VPN client group named “Engineering” as “TOS 5”. Another example of having the granularity of a specified QoS policy imposed would be to allocate 30 percent of the bandwidth on an outbound WAN link to a specific group of remote VPN devices.

How to Configure IPSec and Quality of Service

This section includes the following procedures:

- Configuring IPSec and Quality of Service, page 1132 (required)
- Verifying IPSec and Quality of Service Sessions, page 1133 (optional)
- Troubleshooting Tips, page 1134 (optional)

Configuring IPSec and Quality of Service

To apply QoS policies to an ISAKMP profile, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp-profile** *profile-name*
4. **qos-group** *group-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp-profile** *profile-number* **Example:** `Router (config)# crypto isakmp-profile vpnprofile`	Defines an ISAKMP profile, audits IPSec user sessions, and enters ISAKMP profile configuration mode.
Step 4	**qos-group** *group-number* **Example:** `Router(config-isa-prof)# qos-group 1`	Applies a QoS group value to an ISAKMP profile.

Verifying IPSec and Quality of Service Sessions

To verify your IPSec and QoS sessions, perform the following steps. The **show** commands can be used in any order or independent of each other.

SUMMARY STEPS

1. **enable**
2. **show crypto isakmp profile**
3. **show crypto ipsec sa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto isakmp profile** **Example:** `Router# show crypto isakmp profile`	Shows that the QoS group is applied to the profile.
Step 3	**show crypto ipsec sa** **Example:** `Router# show crypto ipsec sa`	Shows that the QoS group is applied to a particular pair of IPSec security associations (SAs).

Troubleshooting Tips

If you have a problem with your IPSec and QoS sessions, ensure that you have done the following:

- Validated the application of QoS by the QoS service using the QoS-specific commands in the *Cisco IOS Quality of Service Solutions Command Reference*, Release 12.3 T.
- Configured a QoS policy on the router that matches the same QoS group as that specified for the class map match criterion.
- Applied the service policy to the same interface to which a crypto map is applied.

Configuration Examples for IPSec and Quality of Service

This section provides the following output examples:

- QoS Policy Applied to Two Groups of Remote Users: Example, page 1134
- show crypto isakmp profile Command: Example, page 1136
- show crypto ipsec sa Command: Example, page 1136

QoS Policy Applied to Two Groups of Remote Users: Example

In the following example, a specific QoS policy is applied to two groups of remote users. Two ISAKMP profiles are configured so that upon initial connection via IKE, remote users are mapped to a specific profile. From that profile, all IPSec SAs that have been created for that remote will be marked with the specific QoS group. As traffic leaves the outbound interface, the QoS service will map the IPSec set QoS group with the QoS group that is specified in the class maps that comprise the service policy that is applied on that outbound interface.

```
version 12.3
!
aaa authentication login group group radius
aaa authorization network autho local
aaa accounting update periodic 1
```

```
aaa session-id common
ip subnet-zero
!
!
ip cef
no ip domain lookup
!
class-map match-all yellow
 match qos-group 3
class-map match-all blue
match qos-group 2
!
!
policy-map clients
 class blue
  set precedence 5
 class yellow
  set precedence 7
!
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
 group 2
 lifetime 300
!
crypto isakmp keepalive 10 periodic
crypto isakmp xauth timeout 20
!
crypto isakmp client configuration group blue
 key cisco
 dns 10.2.2.2 10.2.2.3
 wins 10.6.6.6
 pool blue
 save-password
 include-local-lan
 backup-gateway corky1.cisco.com
!
crypto isakmp client configuration group yellow
 dns 10.2.2.2 10.2.2.3
 wins 10.6.6.5
 pool yellow
!
crypto isakmp profile blue
   match identity group cisco
   client authentication list autho
   isakmp authorization list autho
   client configuration address respond
   qos-group 2
crypto isakmp profile yellow
   match identity group yellow
   match identity address 10.0.0.11 255.255.255.255
   client authentication list autho
   isakmp authorization list autho
   client configuration address respond
   qos-group 3
!
!
crypto ipsec transform-set combo ah-sha-hmac esp-3des esp-sha-hmac
crypto ipsec transform-set client esp-3des esp-sha-hmac comp-lzs
!
crypto dynamic-map mode 1
 set security-association lifetime seconds 180
```

```
 set transform-set client
 set isakmp-profile blue
 reverse-route
crypto dynamic-map mode 2
 set transform-set combo
 set isakmp-profile yellow
 reverse-route
!
crypto map mode 1 ipsec-isakmp dynamic mode
!
interface FastEthernet0/0
 ip address 10.0.0.110 255.255.255.0
 no ip redirects
 no ip proxy-arp
 no ip mroute-cache
 duplex half
 no cdp enable
 crypto map mode
 service-policy out clients
!
ip local pool yellow 192.168.2.1 192.168.2.10
ip local pool blue 192.168.6.1 192.168.6.6
no ip classless
!
radius-server host 10.0.0.13 auth-port 1645 acct-port 1646
radius-server key XXXXXX
radius-server vsa send accounting
radius-server vsa send authentication
```

show crypto isakmp profile Command: Example

The following output shows that QoS group “2” has been applied to the ISAKMP profile “blue” and that QoS group “3” has been applied to the ISAKMP profile “yellow”:

```
Router# show crypto isakmp profile

ISAKMP PROFILE blue
   Identities matched are:
    group blue
   QoS Group 2 is applied

ISAKMP PROFILE yellow
   Identities matched are:
    ip-address 10.0.0.13 255.255.255.255
    group yellow
   QoS Group 3 is applied
```

show crypto ipsec sa Command: Example

The following output shows that the QoS group has been applied to a particular pair of IPSec SAs:

```
Router# show crypto ipsec sa

interface: FastEthernet0/0
    Crypto map tag: mode, local addr. 10.0.0.110

   protected vrf:
   local  ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
   remote ident (addr/mask/prot/port): (10.12.12.0/255.255.255.0/0/0)
   current_peer: 10.0.0.11:500
```

```
 PERMIT, flags={}
#pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
#pkts decaps: 0, #pkts decrypt: 0, #pkts verify: 0
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. failed: 0
#pkts not decompressed: 0, #pkts decompress failed: 0
#send errors 0, #recv errors 0

 qos group is set to 2
```

Additional References

The following sections provide references related to the IPSec and Quality of Service feature.

Related Documents

Related Topic	Document Title
IPSec	"IP Security and Encryption" chapter of the *Cisco IOS Security Configuration Guide*, Release 12.3
QoS options	*Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.3
QoS commands	*Cisco IOS Quality of Service Solutions Command Reference*, Release 12.3 T
Security commands	*Cisco IOS Security Command Reference*, Release 12.3 T

Standards

Standards	Title
No new or modified standards are supported by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **qos-group**

IPSec Anti-Replay Window: Expanding and Disabling

Cisco IP Security (IPSec) authentication provides anti-replay protection against an attacker duplicating encrypted packets by assigning a unique sequence number to each encrypted packet. The decryptor keeps track of which packets it has seen on the basis of these numbers. Because the decryptor has limited memory, it can presently track only the last 64 packets. Generally, this number (window size) is sufficient, but there are times when you may want to expand this window size. The IPSec Anti-Replay Window: Expanding and Disabling feature allows you to expand the window size, allowing the decryptor to keep more than 64 packets in its memory.

Feature History for IPSec Anti-Replay Window: Expanding and Disabling

Release	Modification
12.3(14)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for IPSec Anti-Replay Window: Expanding and Disabling, page 1140
- Information About IPSec Anti-Replay Window: Expanding and Disabling, page 1140
- How to Configure IPSec Anti-Replay Window: Expanding and Disabling, page 1140
- Configuration Examples for IPSec Anti-Replay Window: Expanding and Disabling, page 1143
- Additional References, page 1145
- Command Reference, page 1146

Prerequisites for IPSec Anti-Replay Window: Expanding and Disabling

- Before configuring this feature, you should have already created a crypto map or crypto profile.

Information About IPSec Anti-Replay Window: Expanding and Disabling

To configure the IPSec Anti-Replay Window: Expanding and Disabling feature, you should understand the following concept:

- IPSec Anti-Replay Window, page 1140

IPSec Anti-Replay Window

Cisco IPSec authentication provides anti-replay protection against an attacker duplicating encrypted packets by assigning a unique sequence number to each encrypted packet. (Security association [SA] anti-replay is a security service in which the receiver can reject old or duplicate packets to protect itself against replay attacks.) The decryptor checks off the sequence numbers that it has seen before. The encryptor assigns sequence numbers in an increasing order. The decryptor remembers the value X of the highest sequence number that it has already seen. N is the window size, or the number of packets that can be held in the memory of the decryptor. Any packet with the sequence number X–N is discarded. Currently, N is set at 64, so only 64 packets can be kept in the memory of the decryptor.

At times, however, the 64-packet window size is not sufficient. For example, Cisco quality of service (QoS) gives priority to high-priority packets, which could cause some low-priority packets to be discarded even though they could be one of the last 64 packets received by the decryptor. The IPSec Anti-Replay Window: Expanding and Disabling feature allows you to expand the window size, allowing the decryptor to keep more than 64 packets in its memory.

How to Configure IPSec Anti-Replay Window: Expanding and Disabling

This section contains the following procedures:

- Configuring IPSec Anti-Replay Window: Expanding and Disabling Globally, page 1140 (optional)
- Configuring IPSec Anti-Replay Window: Expanding and Disabling on a Crypto Map, page 1141 (optional)

Configuring IPSec Anti-Replay Window: Expanding and Disabling Globally

To configure IPSec Anti-Replay Window: Expanding and Disabling globally (so that it affects all SAs that are created— except for those that are specifically overridden on a per-crypto map basis), perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec security-association replay window-size** [*N*]
4. **crypto ipsec security-association replay disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec security-association replay window-size** [*N*] **Example:** `Router (config)# crypto ipsec security-association replay window-size 256`	Sets the size of the SA replay window globally. **Note** Configure this command or the **crypto ipsec security-association replay disable** command. The two commands are not used at the same time.
Step 4	**crypto ipsec security-association replay disable** **Example:** `Router (config)# crypto ipsec security-association replay disable`	Disables checking globally. **Note** Configure this command or the **crypto ipsec security-association replay window-size** command. The two commands are not used at the same time.

Configuring IPSec Anti-Replay Window: Expanding and Disabling on a Crypto Map

To configure IPSec Anti-Replay Window: Expanding and Disabling on a crypto map so that it affects those SAs that have been created using a specific crypto map or profile, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* [**ipsec-isakmp**]
4. **set security-association replay window-size**
5. **set security-association replay disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* [**ipsec-isakmp**] **Example:** `Router (config)# crypto map ETHO 17 ipsec-isakmp`	Enters crypto map configuration mode and creates a crypto profile that provides a template for configuration of dynamically created crypto maps.
Step 4	**set security-association replay window-size** [*N*] **Example:** `Router (crypto-map)# set security-association replay window-size 128`	Controls the SAs that are created using the policy specified by a particular crypto map, dynamic crypto map, or cyrpto profile. **Note** Configure this command or the **set security-association replay disable** command. The two commands are not used at the same time.
Step 5	**set security-association replay disable** **Example:** `Router (crypto-map)# set security-association replay disable`	Disables replay checking for a particular crypto map, dynamic crypto map, or crypto profile. **Note** Configure this command or the **set security-association replay window-size** command. The two commands are not used at the same time.

Troubleshooting Tips

- If your replay window size has not been set to a number that is high enough for the number of packets received, you will receive a system message such as the following:

  ```
  *Nov 17 19:27:32.279: %CRYPTO-4-PKT_REPLAY_ERR: decrypt: replay check failed
  connection id=1
  ```

  The above message is generated when a received packet is judged to be outside the anti-replay window.

Configuration Examples for IPSec Anti-Replay Window: Expanding and Disabling

This section includes the following configuration examples:

- Global Expanding and Disabling of an Anti-Replay Window: Example, page 1143
- Expanding and Disabling of an Anti-Replay Window for a Particular Crypto Map, Dynamic Crypto Map, or Crypto Profile: Example, page 1144

Global Expanding and Disabling of an Anti-Replay Window: Example

The following example shows that the anti-replay window size has been set globally to 1024:

```
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname VPN-Gateway1
!

boot-start-marker
boot-end-marker
!
!
clock timezone EST 0
no aaa new-model
ip subnet-zero
!
!
ip audit po max-events 100
no ftp-server write-enable
!
!
crypto isakmp policy 10
 authentication pre-share
crypto isakmp key cisco123 address 192.165.201.2 !
crypto ipsec security-association replay window-size 1024 !
crypto ipsec transform-set basic esp-des esp-md5-hmac !
crypto map mymap 10 ipsec-isakmp
 set peer 192.165.201.2
 set transform-set basic
 match address 101
!
!
interface Ethernet0/0
 ip address 192.168.1.1 255.255.255.0
!
interface Serial1/0
 ip address 192.165.200.2 255.255.255.252  serial restart-delay 0  crypto map mymap !
ip classless
ip route 0.0.0.0 0.0.0.0 192.165.200.1
no ip http server
no ip http secure-server
!
!
access-list 101 permit ip 192.168.1.0 0.0.0.255 172.16.2.0 0.0.0.255 access-list 101
remark Crypto ACL
!
```

```
!
control-plane
!
!
line con 0
line aux 0
line vty 0 4
!
!
end
```

Expanding and Disabling of an Anti-Replay Window for a Particular Crypto Map, Dynamic Crypto Map, or Crypto Profile: Example

The following example shows that anti-replay checking is disabled for IPSec connections to 172.150.150.2 but enabled (and the default window size is 64) for IPSec connections to 172.150.150.3 and 172.150.150.4:

```
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname dr_whoovie
!
enable secret 5 $1$KxKv$cbqKsZtQTLJLGPN.tErFZ1 enable password ww !
ip subnet-zero
!
cns event-service server

crypto isakmp policy 1
authentication pre-share

crypto isakmp key cisco170 address 172.150.150.2 crypto isakmp key cisco180 address
172.150.150.3 crypto isakmp key cisco190 address 172.150.150.4

crypto ipsec transform-set 170cisco esp-des esp-md5-hmac crypto ipsec transform-set
180cisco esp-des esp-md5-hmac crypto ipsec transform-set 190cisco esp-des esp-md5-hmac

crypto map ETH0 17 ipsec-isakmp
 set peer 172.150.150.2
 set security-association replay disable  set transform-set 170cisco  match address 170
crypto map ETH0 18 ipsec-isakmp  set peer 150.150.150.3  set transform-set 180cisco  match
address 180 crypto map ETH0 19 ipsec-isakmp  set peer 150.150.150.4  set transform-set
190cisco  match address 190 !
interface Ethernet0
 ip address 172.150.150.1 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 no mop enabled
 crypto map ETH0
!
interface Serial0
 ip address 172.160.160.1 255.255.255.0
 no ip directed-broadcast
 no ip mroute-cache
 no fair-queue
!
ip classless
```

```
ip route 172.170.170.0 255.255.255.0 172.150.150.2 ip route 172.180.180.0 255.255.255.0
172.150.150.3 ip route 172.190.190.0 255.255.255.0 172.150.150.4 no ip http server !

access-list 170 permit ip 172.160.160.0 0.0.0.255 172.170.170.0 0.0.0.255 access-list 180
permit ip 172.160.160.0 0.0.0.255 172.180.180.0 0.0.0.255 access-list 190 permit ip
172.160.160.0 0.0.0.255 172.190.190.0 0.0.0.255 !
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
line con 0
transport input none
line aux 0
line vty 0 4
password ww
login
end
```

Additional References

The following sections provide references related to IPSec Anti-Replay Window: Expanding and Disabling.

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Security Command Reference*, Release 12.3T
IP security and encryption	"IP Security and Encryption" section of *Cisco IOS Security Configuration Guide*, Release 12.3

Standards

Standards	Title
No new of modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new of modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new of modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **crypto ipsec security-association replay window-size**
- **crypto ipsec security-association replay window-size**
- **set security-association replay disable**
- **set security-association replay window-size**

IPSec Dead Peer Detection Periodic Message Option

The IPSec Dead Peer Detection Periodic Message Option feature allows you to configure your router to query the liveliness of its Internet Key Exchange (IKE) peer at regular intervals. The benefit of this approach over the default approach (on-demand dead peer detection) is earlier detection of dead peers.

Feature History for IPSec Dead Peer Detection Periodic Message Option

Release	Modification
12.3(7)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for IPSec Dead Peer Detection Periodic Message Option, page 1148
- Restrictions for IPSec Dead Peer Detection Periodic Message Option, page 1148
- Information About IPSec Dead Peer Detection Periodic Message Option, page 1148
- How to Configure IPSec Dead Peer Detection Periodic Message Option, page 1149
- Configuration Examples for IPSec Dead Peer Detection Periodic Message Option, page 1153
- Additional References, page 1158
- Command Reference, page 1159

Prerequisites for IPSec Dead Peer Detection Periodic Message Option

Before configuring the IPSec Dead Peer Detection Periodic Message Option feature, you should have the following:

- Familiarity with configuring IP Security (IPSec).
- An IKE peer that supports DPD (dead peer detection). Implementations that support DPD include the Cisco VPN 3000 concentrator, Cisco PIX Firewall, Cisco VPN Client, and Cisco IOS software in all modes of operation—site-to-site, Easy VPN remote, and Easy VPN server.

Restrictions for IPSec Dead Peer Detection Periodic Message Option

Using periodic DPD potentially allows the router to detect an unresponsive IKE peer with better response time when compared to on-demand DPD. However, use of periodic DPD incurs extra overhead. When communicating to large numbers of IKE peers, you should consider using on-demand DPD instead.

Information About IPSec Dead Peer Detection Periodic Message Option

To configure IPSec Dead Peer Detection Periodic Message Option, you should understand the following concepts:

- How DPD and Cisco IOS Keepalive Features Work, page 1148
- Using the IPSec Dead Peer Detection Periodic Message Option, page 1149
- Using DPD and Cisco IOS Keepalive Features with Multiple Peers in the Crypto Map, page 1149
- Using DPD in an Easy VPN Remote Configuration, page 1149

How DPD and Cisco IOS Keepalive Features Work

DPD and Cisco IOS keepalives function on the basis of the timer. If the timer is set for 10 seconds, the router will send a “hello” message every 10 seconds (unless, of course, the router receives a “hello” message from the peer). The benefit of IOS keepalives and periodic DPD is earlier detection of dead peers. However, IOS keepalives and periodic DPD rely on periodic messages that have to be sent with considerable frequency. The result of sending frequent messages is that the communicating peers must encrypt and decrypt more packets.

DPD also has an on-demand approach. The contrasting on-demand approach is the default. With on-demand DPD, messages are sent on the basis of traffic patterns. For example, if a router has to send outbound traffic and the liveliness of the peer is questionable, the router sends a DPD message to query the status of the peer. If a router has no traffic to send, it never sends a DPD message. If a peer is dead, and the router never has any traffic to send to the peer, the router will not find out until the IKE or IPSec

security association (SA) has to be rekeyed (the liveliness of the peer is unimportant if the router is not trying to communicate with the peer). On the other hand, if the router has traffic to send to the peer, and the peer does not respond, the router will initiate a DPD message to determine the state of the peer.

Using the IPSec Dead Peer Detection Periodic Message Option

With the IPSec Dead Peer Detection Periodic Message Option feature, you can configure your router so that DPD messages are "forced" at regular intervals. This forced approach results in earlier detection of dead peers. For example, if a router has no traffic to send, a DPD message is still sent at regular intervals, and if a peer is dead, the router does not have to wait until the IKE SA times out to find out.

If you want to configure the DPD periodic message option, you should use the **crypto isakmp keepalive** command with the **periodic** keyword. If you do not configure the **periodic** keyword, the router defaults to the on-demand approach.

Note When the **crypto isakmp keepalive** command is configured, the IOS software negotiates the use of IOS keepalives or DPD, depending on which protocol the peer supports.

Using DPD and Cisco IOS Keepalive Features with Multiple Peers in the Crypto Map

DPD and IOS keepalive features can be used in conjunction with multiple peers in the crypto map to allow for stateless failover. DPD allows the router to detect a dead IKE peer, and when the router detects the dead state, the router deletes the IPSec and IKE SAs to the peer. If you configure multiple peers, the router will switch over to the next listed peer for a stateless failover.

Using DPD in an Easy VPN Remote Configuration

DPD can be used in an Easy VPN remote configuration. See the section "Configuring DPD for an Easy VPN Remote."

How to Configure IPSec Dead Peer Detection Periodic Message Option

This section contains the following prodedures:

- Configuring a Periodic DPD Message, page 1150
- Configuring DPD and Cisco IOS Keepalives with Multiple Peers in the Crypto Map, page 1150
- Configuring DPD for an Easy VPN Remote, page 1151
- Verifying That DPD Is Enabled, page 1152

Configuring a Periodic DPD Message

To configure a periodic DPD message, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp keepalive** *seconds* [*retries*] [**periodic** | **on-demand**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp keepalive** *seconds* [*retries*] [**periodic** | **on-demand**] **Example:** `Router (config)# crypto isakmp keepalive 10 periodic`	Allows the gateway to send DPD messages to the peer. • *seconds*—Number of seconds between DPD messages. • *retries*—(Optional) Number of seconds between DPD retries if the DPD message fails. • **periodic**—(Optional) DPD messages are sent at regular intervals. • **on-demand**—(Optional) DPD retries are sent on demand. This is the default behavior.

Configuring DPD and Cisco IOS Keepalives with Multiple Peers in the Crypto Map

To configure DPD and IOS keepalives to be used in conjunction with the crypto map to allow for stateless failover, perform the following steps. This configuration will cause a router to cycle through the peer list when it detects that the first peer is dead.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* **ipsec-isakmp**
4. **set peer** {*host-name* [**dynamic**] | *ip-address*}

5. **set transform-set** *transform-set-name*
6. **match address** [*access-list-id* | *name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* **ipsec-isakmp** **Example:** `Router (config)# crypto map green 1 ipsec-isakmp`	Enters crypto map configuration mode and creates or modifies a crypto map entry. • The **ipsec-isakmp** keyword indicates that IKE will be used to establish the IPSec SAs for protecting the traffic specified by this crypto map entry.
Step 4	**set peer** {*host-name* [**dynamic**] | *ip-address*} **Example:** `Router (config-crypto-map)# set peer 12.12.12.12`	Specifies an IPSec peer in a crypto map entry. • You can specify multiple peers by repeating this command.
Step 5	**set transform-set** *transform-set-name* **Example:** `Router (config-crypto-map)# set transform-set txfm`	Specifies which transform sets can be used with the crypto map entry. • You can specify more than one transform set name by repeating this command.
Step 6	**match address** [*access-list-id* | *name*] **Example:** `Router (config-crypto-map)# match address 101`	Specifies an extended access list for a crypto map entry.

Configuring DPD for an Easy VPN Remote

To configure DPD in an Easy VPN remote configuration, perform the following steps. This configuration also will cause a router to cycle through the peer list when it detects that the first peer is dead.

Note IOS keepalives are not supported for Easy VPN remote configurations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **crypto ipsec client ezvpn** *name*
4. **connect** {**auto** | **manual**}
5. **group** *group-name* **key** *group-key*
6. **mode** {**client** | **network-extension**}
7. **peer** {*ipaddress* | *hostname*}

DETAILED STEPS

Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec client ezvpn** *name* **Example:** `Router (config)# crypto ipsec client ezvpn ezvpn-config1`	Creates a Cisco Easy VPN remote configuration and enters the Cisco Easy VPN Remote configuration mode.
Step 4	**connect** {**auto** | **manual**} **Example:** `Router (config-crypto-ezvpn)# connect manual`	Manually establishes and terminates an IPSec VPN tunnel on demand. • The **auto** keyword option is the default setting.
Step 5	**group** *group-name* **key** *group-key* **Example:** `Router (config-crypto-ezvpn)# group unity key preshared`	Specifies the group name and key value for the Virtual Private Network (VPN) connection.
Step 6	**mode** {**client** | **network-extension**} **Example:** `Router (config-crypto-ezvpn)# mode client`	Specifies the VPN mode of operation of the router.
Step 7	**peer** {*ipaddress* | *hostname*} **Example:** `Router (config-crypto-ezvpn)# peer 10.10.10.10`	Sets the peer IP address or host name for the VPN connection. • A hostname can be specified only when the router has a DNS server available for host-name resolution. • This command can be repeated multiple times.

Verifying That DPD Is Enabled

DPD allows the router to clear the IKE state when a peer becomes unreachable. If DPD is enabled and the peer is unreachable for some time, you can use the **clear crypto session** command to manually clear IKE and IPSec SAs.

The **debug crypto isakmp** command can be used to verify that DPD is enabled.

SUMMARY STEPS

1. **enable**
2. **clear crypto session** [**local** *ip-address* [**port** *local-port*]] [**remote** *ip-address* [**port** *remote-port*]] | [**fvrf** *vrf-name*] [**ivrf** *vrf-name*]
3. **debug crypto isakmp**

DETAILED STEPS

Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear crypto session** [**local** ip-address [**port** local-port]] [**remote** ip-address [**port** remote-port]] | [**fvrf** vrf-name] [**ivrf** vrf-name] **Example:** `Router# clear crypto session`	Deletes crypto sessions (IPSec and IKE SAs).
Step 3	**debug crypto isakmp** **Example:** `Router# debug crypto isakmp`	Displays messages about IKE events.

Configuration Examples for IPSec Dead Peer Detection Periodic Message Option

This section provides the following configuration examples:

- Site-to-Site Setup with Periodic DPD Enabled: Example, page 1153
- Easy VPN Remote with DPD Enabled: Example, page 1154
- Verifying DPD Configuration Using the debug crypto isakmp Command: Example, page 1154
- DPD and IOS Keepalives Used in Conjunction with Multiple Peers in a Crypto Map: Example, page 1157
- DPD Used in Conjunction with Multiple Peers for an Easy VPN Remote: Example, page 1157

Site-to-Site Setup with Periodic DPD Enabled: Example

The following configurations are for a site-to-site setup with no periodic DPD enabled. The configurations are for the IKE Phase 1 policy and for the IKE preshared key.

IKE Phase 1 Policy

```
crypto isakmp policy 1
  encryption 3des
```

```
  authentication pre-share
  group 2
!
```

IKE Preshared Key

```
crypto isakmp key kd94j1ksldz address 10.2.80.209 255.255.255.0
crypto isakmp keepalive 10 periodic
crypto ipsec transform-set esp-3des-sha esp-3des esp-sha-hmac
crypto map test 1 ipsec-isakmp
  set peer 10.2.80.209
  set transform-set esp-3des-sha
  match address 101
!
!
interface FastEthernet0
  ip address 10.1.32.14 255.255.255.0
  speed auto
  crypto map test
!
```

Easy VPN Remote with DPD Enabled: Example

The following configuration tells the router to send a periodic DPD message every 30 seconds. If the peer fails to respond to the DPD R_U_THERE message, the router will resend the message every 20 seconds (four transmissions altogether).

```
crypto isakmp keepalive 30 20 periodic
crypto ipsec client ezvpn ezvpn-config
  connect auto
  group unity key preshared
  mode client
  peer 10.2.80.209
!
!
interface Ethernet0
  ip address 10.2.3.4 255.255.255.0
  half-duplex
  crypto ipsec client ezvpn ezvpn-config inside
!
interface FastEthernet0
  ip address 10.1.32.14 255.255.255.0
  speed auto
  crypto ipsec client ezvpn ezvpn-config outside
```

Verifying DPD Configuration Using the debug crypto isakmp Command: Example

The following sample output from the **debug crypto isakmp** command verifies that IKE DPD is enabled:

```
*Mar 25 15:17:14.131: ISAKMP:(0:1:HW:2):IKE_DPD is enabled, initializing timers
```

To see that IKE DPD is enabled (and that the peer supports DPD): when periodic DPD is enabled, you should see the following debug messages at the interval specified by the command:

```
*Mar 25 15:18:52.107: ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
*Mar 25 15:18:52.107: ISAKMP:(0:1:HW:2):purging node 899852982 *Mar 25 15:18:52.111:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_IM_ALIVE
*Mar 25 15:18:52.111: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE
```

The above message corresponds to sending the DPD R_U_THERE message.

```
*Mar 25 15:18:52.123: ISAKMP (0:268435457): received packet from 10.2.80.209
dport 500 sport 500 Global (I) QM_IDLE
*Mar 25 15:18:52.123: ISAKMP: set new node -443923643 to QM_IDLE *Mar 25 15:18:52.131:
ISAKMP:(0:1:HW:2): processing HASH payload. message ID =
-443923643
*Mar 25 15:18:52.131: ISAKMP:(0:1:HW:2): processing NOTIFY R_U_THERE_ACK protocol 1
    spi 0, message ID = -443923643, sa = 81BA4DD4
*Mar 25 15:18:52.135: ISAKMP:(0:1:HW:2): DPD/R_U_THERE_ACK received from peer
10.2.80.209, sequence 0x9
*Mar 25 15:18:52.135: ISAKMP:(0:1:HW:2):deleting node -443923643 error FALSE
reason "informational (in) state 1"
*Mar 25 15:18:52.135: ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_PEER, IKE_INFO_NOTIFY *Mar
25 15:18:52.135: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE
```

The above message corresponds to receiving the acknowledge (ACK) message from the peer.

```
Router#
*Mar 25 15:47:35.335: ISAKMP: set new node -90798077 to QM_IDLE *Mar 25 15:47:35.343:
ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
*Mar 25 15:47:35.343: ISAKMP:(0:1:HW:2):purging node -90798077 *Mar 25 15:47:35.347:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_IM_ALIVE
*Mar 25 15:47:35.347: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE

*Mar 25 15:47:36.611: ISAKMP:(0:1:HW:2):purging node 1515050537 *Mar 25 15:47:37.343:
ISAKMP:(0:1:HW:2):incrementing error counter on sa:
PEERS_ALIVE_TIMER
*Mar 25 15:47:37.343: ISAKMP: set new node -1592471565 to QM_IDLE *Mar 25 15:47:37.351:
ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
*Mar 25 15:47:37.351: ISAKMP:(0:1:HW:2):purging node -1592471565 *Mar 25 15:47:37.355:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_PEERS_ALIVE
*Mar 25 15:47:37.355: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE

*Mar 25 15:47:39.355: ISAKMP:(0:1:HW:2):incrementing error counter on sa:
PEERS_ALIVE_TIMER
*Mar 25 15:47:39.355: ISAKMP: set new node 1758739401 to QM_IDLE *Mar 25 15:47:39.363:
ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
*Mar 25 15:47:39.363: ISAKMP:(0:1:HW:2):purging node 1758739401 *Mar 25 15:47:39.367:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_PEERS_ALIVE
*Mar 25 15:47:39.367: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE
```

```
*Mar 25 15:47:41.367: ISAKMP:(0:1:HW:2):incrementing error counter on sa:
PEERS_ALIVE_TIMER
*Mar 25 15:47:41.367: ISAKMP: set new node 320258858 to QM_IDLE *Mar 25 15:47:41.375:
ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
*Mar 25 15:47:41.379: ISAKMP:(0:1:HW:2):purging node 320258858 *Mar 25 15:47:41.379:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_PEERS_ALIVE
*Mar 25 15:47:41.379: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE

*Mar 25 15:47:43.379: ISAKMP:(0:1:HW:2):incrementing error counter on sa:
PEERS_ALIVE_TIMER
*Mar 25 15:47:43.379: ISAKMP: set new node -744493014 to QM_IDLE *Mar 25 15:47:43.387:
ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
*Mar 25 15:47:43.387: ISAKMP:(0:1:HW:2):purging node -744493014 *Mar 25 15:47:43.391:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_PEERS_ALIVE
*Mar 25 15:47:43.391: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE

*Mar 25 15:47:45.391: ISAKMP:(0:1:HW:2):incrementing error counter on sa:
PEERS_ALIVE_TIMER
*Mar 25 15:47:45.391: ISAKMP:(0:1:HW:2):peer 10.2.80.209 not responding! *Mar 25
15:47:45.391: ISAKMP:(0:1:HW:2):peer does not do paranoid keepalives.

*Mar 25 15:47:45.391: ISAKMP:(0:1:HW:2):deleting SA reason "peers alive" state
(I) QM_IDLE       (peer 10.2.80.209) input queue 0
*Mar 25 15:47:45.395: ISAKMP: Unlocking IPSEC struct 0x81E5C4E8 from
delete_siblings, count 0
*Mar 25 15:47:45.395: %CRYPTO-5-SESSION_STATUS: Crypto tunnel is DOWN.  Peer
10.2.80.209:500       Id: 10.2.80.209
*Mar 25 15:47:45.399: ISAKMP: set new node -2061951065 to QM_IDLE *Mar 25 15:47:45.411:
ISAKMP:(0:1:HW:2): sending packet to 10.2.80.209 my_port
500 peer_port 500 (I) QM_IDLE
*Mar 25 15:47:45.411: ISAKMP:(0:1:HW:2):purging node -2061951065 *Mar 25 15:47:45.411:
ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_TIMER,
IKE_TIMER_PEERS_ALIVE
*Mar 25 15:47:45.411: ISAKMP:(0:1:HW:2):Old State = IKE_P1_COMPLETE  New State =
IKE_DEST_SA

*Mar 25 15:47:45.415: ISAKMP:(0:1:HW:2):deleting SA reason "peers alive" state
(I) QM_IDLE       (peer 10.2.80.209) input queue 0
*Mar 25 15:47:45.415: ISAKMP: Unlocking IKE struct 0x81E5C4E8 for
isadb_mark_sa_deleted(), count 0
*Mar 25 15:47:45.415: ISAKMP: Deleting peer node by peer_reap for 10.2.80.209:
81E5C4E8
*Mar 25 15:47:45.415: ISAKMP:(0:1:HW:2):deleting node -1067612752 error TRUE
reason "peers alive"
*Mar 25 15:47:45.415: ISAKMP:(0:1:HW:2):deleting node -114443536 error TRUE
reason "peers alive"
*Mar 25 15:47:45.419: ISAKMP:(0:1:HW:2):deleting node 2116015069 error TRUE
reason "peers alive"
*Mar 25 15:47:45.419: ISAKMP:(0:1:HW:2):deleting node -1981865558 error TRUE
reason "peers alive"
*Mar 25 15:47:45.419: ISAKMP:(0:1:HW:2):Input = IKE_MESG_INTERNAL, IKE_PHASE1_DEL *Mar 25
15:47:45.419: ISAKMP:(0:1:HW:2):Old State = IKE_DEST_SA  New State =
IKE_DEST_SA

*Mar 25 15:47:45.419: ISAKMP: received ke message (4/1)
*Mar 25 15:47:45.419: ISAKMP: received ke message (3/1)
*Mar 25 15:47:45.423: ISAKMP: ignoring request to send delete notify (no ISAKMP
sa) src 10.1.32.14 dst 10.2.80.209 for SPI 0x3A7B69BF
```

```
*Mar 25 15:47:45.423: ISAKMP:(0:1:HW:2):deleting SA reason "" state (I)
MM_NO_STATE (peer 10.2.80.209) input queue 0
*Mar 25 15:47:45.423: ISAKMP:(0:1:HW:2):deleting node -1067612752 error FALSE
reason ""
*Mar 25 15:47:45.423: ISAKMP:(0:1:HW:2):deleting node -114443536 error FALSE
reason ""
*Mar 25 15:47:45.423: ISAKMP:(0:1:HW:2):deleting node 2116015069 error FALSE
reason ""
*Mar 25 15:47:45.427: ISAKMP:(0:1:HW:2):deleting node -1981865558 error FALSE
reason ""
*Mar 25 15:47:45.427: ISAKMP:(0:1:HW:2):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH *Mar 25
15:47:45.427: ISAKMP:(0:1:HW:2):Old State = IKE_DEST_SA  New State =
IKE_DEST_SA
```

The above message shows what happens when the remote peer is unreachable. The router sends one DPD R_U_THERE message and four retransmissions before it finally deletes the IPSec and IKE SAs.

DPD and IOS Keepalives Used in Conjunction with Multiple Peers in a Crypto Map: Example

The following example shows that DPD and IOS keepalives are used in conjunction with multiple peers in a crypto map configuration when IKE will be used to establish the security associations (SAs). In this example, an SA could be set up to the IPSec peer at 10.0.0.1, 10.0.0.2, or 10.0.0.3.

```
crypto map green 1 ipsec-isakmp
  set peer 10.0.0.1
  set peer 10.0.0.2
  set peer 10.0.0.3
  set transform-set txfm
  match address 101
```

DPD Used in Conjunction with Multiple Peers for an Easy VPN Remote: Example

The following example shows that DPD is used in conjunction with multiple peers in an Easy VPN remote configuration. In this example, an SA could be set up to the IPSec peer at 10.10.10.10, 10.2.2.2, or 10.3.3.3.

```
crypto ipsec client ezvpn ezvpn-config
  connect auto
  group unity key preshared
  mode client
  peer 10.10.10.10
  peer 10.2.2.2
  peer 10.3.3.3
```

Additional References

The following sections provide references related to IPSec Dead Peer Detection Periodic Message Option.

Related Documents

Related Topic	Document Title
Configuring IPSec	"IP Security and Encryption" section of *Cisco IOS Security Configuration Guide*
IPSec commands	*Cisco IOS Security Command Reference*, Release 12.3 T

Standards

Standards	Title
No new or modified standards are supported by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
DPD conforms to the Internet draft "draft-ietf-ipsec-dpd-04.txt," which is pending publication as an Informational RFC (a number has not yet been assigned).	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following modified command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **crypto isakmp keepalive**

IPSec NAT Transparency

The IPSec NAT Transparency feature introduces support for IP Security (IPSec) traffic to travel through Network Address Translation (NAT) or Point Address Translation (PAT) points in the network by addressing many known incompatabilites between NAT and IPSec.

Feature Specifications for the IPSec NAT Transparency feature

Feature History	
Release	**Modification**
12.2(13)T	This feature was introduced.
Supported Platforms	
For platforms supported in Cisco IOS Release 12.2(13)T, consult Cisco Feature Navigator.	

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that are supported on specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Contents

- Restrictions for IPSec NAT Transparency, page 1162
- Information About IPSec NAT Transparency, page 1162
- How to Configure NAT and IPSec, page 1167
- Configuration Examples for IPSec and NAT, page 1169
- Additional References, page 1169
- Command Reference, page 1171
- Glossary, page 1172

Restrictions for IPSec NAT Transparency

Although this feature addresses many incompatibilities between NAT and IPSec, the following problems still exist:

Internet Key Exchange (IKE) IP Address and NAT

This incompatibility applies only when IP addresses are used as a search key to find a preshared key. Modification of the IP source or destination addresses by NAT or reverse NAT results in a mismatch between the IP address and the preshared key.

Embedded IP Addresses and NAT

Because the payload is integrity protected, any IP address enclosed within IPSec packets cannot be translated by NAT. Protocols that use embedded IP addresses include FTP, Internet Relay Chat (IRC), Simple Network Mangement Protocol (SNMP), Lightweight Directory Access Protocol (LDAP), H.323, and Session Initiation Protocol (SIP).

Information About IPSec NAT Transparency

To configure the IPSec NAT Transparency feature, you must understand the following concepts:

- Benefit of IPSec NAT Transparency, page 1163
- Feature Design of IPSec NAT Traversal, page 1163
- NAT Keepalives, page 1166

Benefit of IPSec NAT Transparency

Before the introduction of this feature, a standard IPSec virtual private network (VPN) tunnel would not work if there were one or more NAT or PAT points in the delivery path of the IPSec packet. This feature makes NAT IPSec-aware, thereby, allowing remote access users to build IPSec tunnels to home gateways.

Feature Design of IPSec NAT Traversal

The IPSec NAT Transparency feature introduces support for IPSec traffic to travel through NAT or PAT points in the network by encapsulating IPSec packets in a User Datagram Protocol (UDP) wrapper, which allows the packets to travel across NAT devices. The following sections define the details of NAT traversal:

- IKE Phase 1 Negotiation: NAT Detection
- IKE Phase 2 Negotiation: NAT Traversal Decision
- UDP Encapsulation of IPSec Packets for NAT Traversal
- UDP Encapsulated Process for Software Engines: Transport Mode and Tunnel Mode ESP Encapsulation

IKE Phase 1 Negotiation: NAT Detection

During Internet Key Exchange (IKE) phase 1 negotiation, two types of NAT detection occur before IKE Quick Mode begins—NAT support and NAT existence along the network path.

To detect NAT support, you should exchange the vendor identification (ID) string with the remote peer. During Main Mode (MM) 1 and MM 2 of IKE phase 1, the remote peer sends a vendor ID string payload to its peer to indicate that this version supports NAT traversal. Thereafter, NAT existence along the network path can be determined.

Detecting whether NAT exists along the network path allows you to find any NAT device between two peers and the exact location of NAT. A NAT device can translate the private IP address and port to public value (or from public to private). This translation changes the IP address and port if the packet goes through the device. To detect whether a NAT device exists along the network path, the peers should send a payload with hashes of the IP address and port of both the source and destination address from each end. If both ends calculate the hashes and the hashes match, each peer knows that a NAT device does not exist on the network path between them. If the hashes do not match (that is, someone translated the address or port), then each peer needs to perform NAT traversal to get the IPSec packet through the network.

The hashes are sent as a series of NAT discovery (NAT-D) payloads. Each payload contains one hash; if multiple hashes exist, multiple NAT-D payloads are sent. In most environments, there are only two NAT-D payloads—one for the source address and port and one for the destination address and port. The destination NAT-D payload is sent first, followed by the source NAT-D payload, which implies that the receiver should expect to process the local NAT-D payload first and the remote NAT-D payload second. The NAT-D payloads are included in the third and fourth messages in Main Mode and in the second and third messages in Aggressive Mode (AM).

IKE Phase 2 Negotiation: NAT Traversal Decision

While IKE phase 1 detects NAT support and NAT existence along the network path, IKE phase 2 decides whether or not the peers at both ends will use NAT traversal. Quick Mode (QM) security association (SA) payload in QM1 and QM2 is used to for NAT traversal negotiation.

Because the NAT device changes the IP address and port number, incompatablities between NAT and IPSec can be created. Thus, exchanging the original source address bypasses any incompatablities.

UDP Encapsulation of IPSec Packets for NAT Traversal

In addition to allowing IPSec packets to traverse across NAT devices, UDP encapsulation also addresses many incompatability issues between IPSec and NAT and PAT. The resolved issues are as follows:

Incompatability Between IPSec ESP and PAT—Resolved

If PAT found a legislative IP address and port, it would drop the Encapsulating Security Payload (ESP) packet. To prevent this scenario, UDP encapulation is used to hide the ESP packet behind the UDP header. Thus, PAT treats the ESP packet as a UDP packet, processing the ESP packet as a normal UDP packet.

Incompatability Between Checksums and NAT—Resolved

In the new UDP header, the checksum value is always assigned to zero. This value prevents an intermediate device from validating the checksum against the packet checksum, thereby, resolving the TCP UDP checksum issue because NAT changes the IP source and destination addresses.

Incompatability Between Fixed IKE Destination Ports and PAT—Resolved

PAT changes the port address in the new UDP header for translation and leaves the original payload unchanged.

To see how UDP encapsulation helps to send IPSec packets see Figure 78 and Figure 79.

Figure 78 Standard IPSec Tunnel Through a NAT/PAT Point (No UDP Encapsulation)

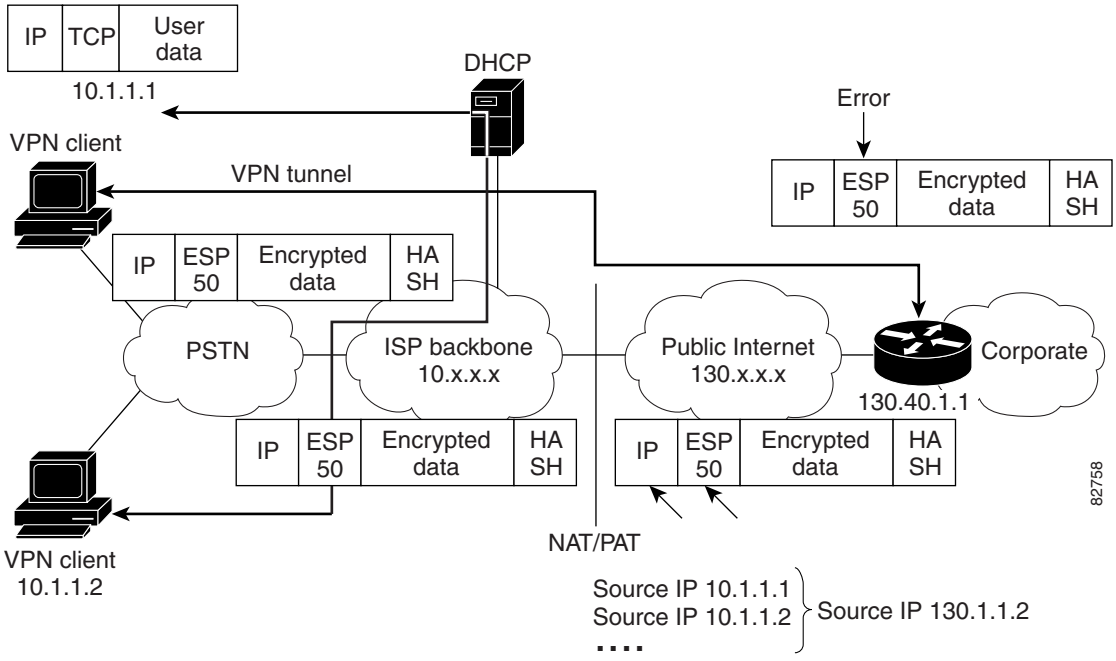
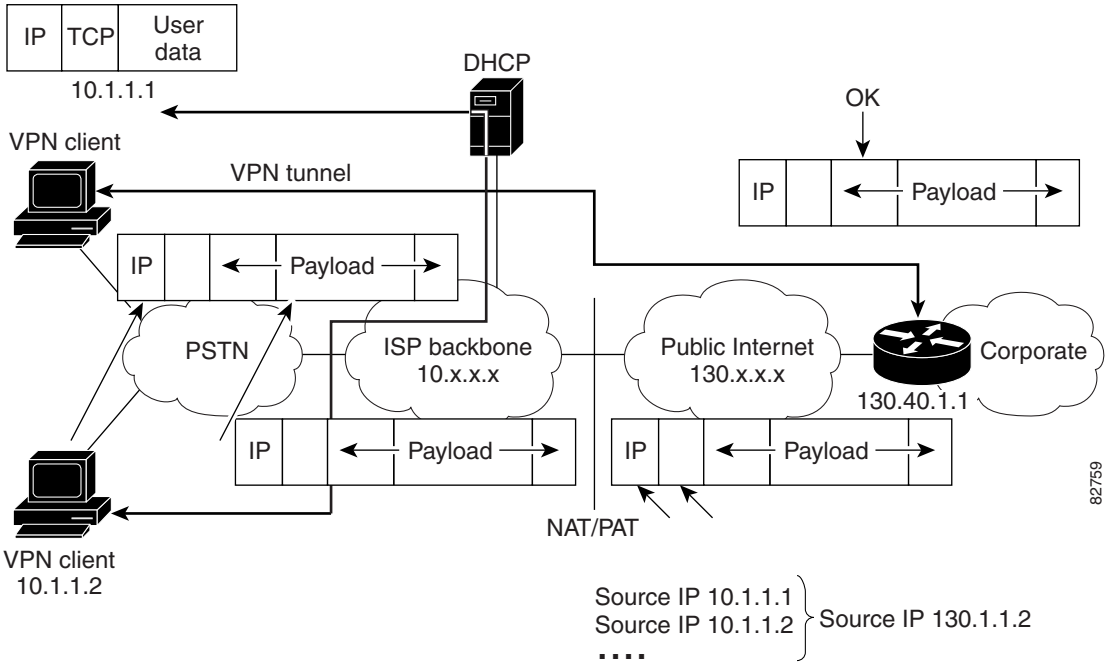


Figure 79 IPSec Packet with UDP Encapsulation



UDP Encapsulated Process for Software Engines: Transport Mode and Tunnel Mode ESP Encapsulation

After the IPSec packet is encrypted by a hardware accelerator or a software crypto engine, a UDP header and a non-IKE marker (which is 8 bytes in length) are inserted between the original IP header and ESP header. The total length, protocol, and checksum fields are changed to match this modification. Figure 80 shows an IPSec packet before and after transport mode is applied; Figure 81 shows an IPSec packet before and after tunnel mode is applied.

Figure 80 Transport Mode—IPSec Packet Before and After ESP Encapsulation

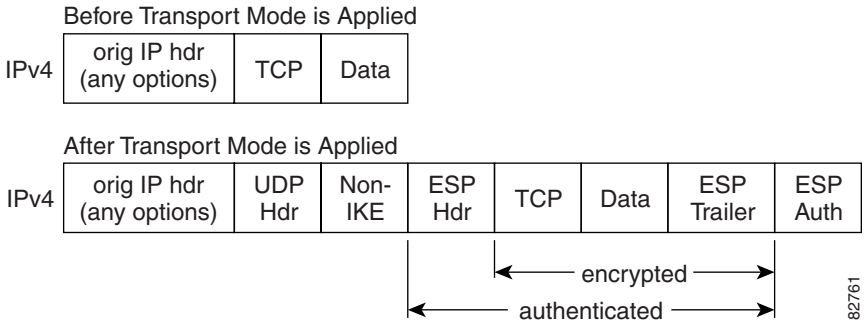
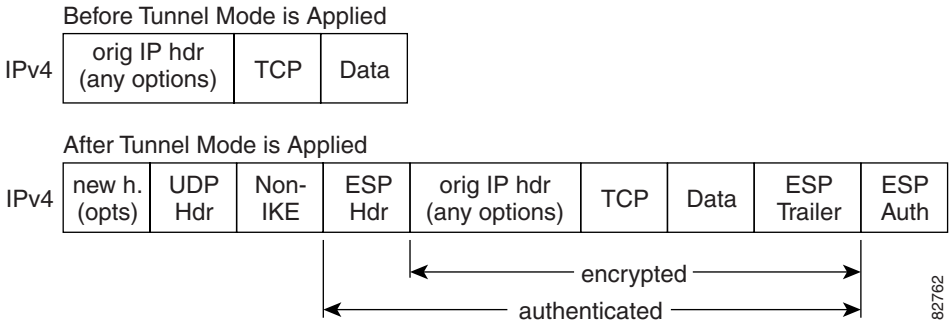


Figure 81 Tunnel Mode—IPSec Packet Before and After ESP Encapsualtion



NAT Keepalives

NAT keepalives are enabled to keep the dynamic NAT mapping alive during a connection between two peers. NAT keepalives are UDP packets with an unencrypted payload of 1 byte. Although the currrent dead peer detection (DPD) implementation is similar to NAT keepalives, there is a slight difference: DPD is used to detect peer status, while NAT keepalives are sent if the IPSec entity did not send or receive the packet at a specified period of time—valid range is between 5 to 3600 seconds.

If NAT keepalives are enabled (via the **crypto isamkp nat keepalive** command), users should ensure that the idle value is shorter than the NAT mapping expiration time, which is 20 seconds.

How to Configure NAT and IPSec

This section contains the following procedures:

- Configuring NAT Traversal, page 1167 (optional)
- Disabling NAT Traversal, page 1167 (optional)
- Configuring NAT Keepalives, page 1168 (optional)
- Verifying IPSec Configuration, page 1168 (optional)

Configuring NAT Traversal

NAT Traversal is a feature that is auto detected by VPN devices. There are no configuration steps for a router running Cisco IOS Release 12.2(13)T. If both VPN devices are NAT-T capable, NAT Traversal is auto detected and auto negotiated.

Disabling NAT Traversal

You may wish to disable NAT traversal if you already know that your network uses IPSec-awareness NAT (spi-matching scheme). To disable NAT traversal, use the following commands:

SUMMARY STEPS:

1. **enable**
2. **configure terminal**
3. **no crypto ipsec nat-transparency udp-encapsulation**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**no crypto ipsec nat-transparency udp-encapsulation** **Example:** `Router(config)# no crypto ipsec nat-transparency udp-encapsulation`	Disables NAT traversal.

Configuring NAT Keepalives

To configure your router to send NAT keepalives, use the following commands:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp nat keepalive** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp nat keepalive** *seconds* **Example:** `Router(config)# crypto isakmp nat keepalive 20`	Allows an IPSec node to send NAT keepalive packets. • *seconds*—The number of seconds between keepalive packets; range is between 5 to 3,600 seconds.

Verifying IPSec Configuration

To verify your configuration, perform the following optional steps:

SUMMARY STEPS

1. **enable**
2. **show crypto ipsec sa** [**map** *map-name* | **address** | **identity**] [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**show crypto ipsec sa** [**map** *map-name* | **address** | **identity**] [**detail**] **Example:** `Router# show crypto ipsec sa`	Displays the settings used by current SAs.

Configuration Examples for IPSec and NAT

This section provides the following configuration example:

- NAT Keepalives Configuration Example, page 1169

NAT Keepalives Configuration Example

The following example shows how to enable NAT keepalives to be sent every 20 seconds:

```
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key 1234 address 56.0.0.1
crypto isakmp nat keepalive 20
!
!
crypto ipsec transform-set t2 esp-des esp-sha-hmac
!
crypto map test2 10 ipsec-isakmp
 set peer 56.0.0.1
 set transform-set t2
 match address 101
```

Additional References

The following sections provide additional references related to IPSec NAT Transparency:

- Related Documents, page 1170
- Standards, page 1170
- MIBs, page 1170
- RFCs, page 1171
- Technical Assistance, page 1171

Related Documents

Related Topic	Document Title
Additional NAT configuration tasks.	The chapter "Configuring IP Addressing" in the *Cisco IOS IP Configuration Guide*, Release 12.2
Additional NAT commands	The chapter "IP Addressing Commands" in the *Cisco IOS IP Command Reference, Volume 1 of 3: Addressing and Services*, Release 12.2
Additional IPSec configuration tasks	The chapter "Configuring IPSec Network Security" in the *Cisco IOS Security Configuration Guide*, Release 12.2
Additional IPSec commands	The chapter "IPSec Network Security Commands" in the *Cisco IOS Security Command Reference*, Release 12.2
Information on IKE phase 1 and phase 2, Agressive Mode, and Main Mode.	The chapter "Configuring Internet Key Exchange Security Protocol" in the *Cisco IOS Security Configuration Guide*, Release 12.2
Additional information on IKE dead peer detection.	*Easy VPN Server*, Cisco IOS Release 12.2(8)T feature module

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

RFCs[1]	Title
RFC 2402	*IP Authentication Header*
RFC 2406	*IP Encapsulating Security Payload (ESP)*

1. Not all supported RFCs are listed.

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

New Command

- **crypto isamkp nat keepalive**

Modified Commands

- **access-list (IP extended)**
- **show crypto ipsec sa**

Glossary

IKE—Internet Key Exchange. Hybrid protocol that implements Oakley key exchange and Skeme key exchange inside the Internet Security Association Key Management Protocol (ISAKMP) framework. Although IKE can be used with other protocols, its initial implementation is with IPSec. IKE provides authentication of the IPSec peers, negotiates IPSec keys, and negotiates IPSec security associations (SAs).

IPSec—IP Security. Framework of open standards developed by the Internet Engineering Task Force (IETF). IPSec provides security for transmission of sensitive information over unprotected networks such as the Internet. IPSec acts at the network layer, protecting and authenticating IP packets between participating IPSec devices ("peers"), such as Cisco routers.

NAT—Network Address Translation. Translates a private IP address used inside the corporation to a public, routable address for use on the outside of the corporation, such as the Internet. NAT is considered a one-to-one mapping of addresses from private to public.

PAT—Port Address Translation. Like NAT, PAT also translated private IP address to public, routable addresses. Unlike NAT, PAT provides a many-to-one mapping of private addresses to a public address; each instance of the public address is associated with a particular port number to provide uniqueness. PAT can be used in environments where the cost of obtaining a range of public addresses is too expensive for an organization.

Note Refer to the *Internetworking Terms and Acronyms* for terms not included in this glossary.

IPSec Preferred Peer

The IP Security (IPSec) Preferred Peer feature allows you to control the circumstances by which multiple peers on a crypto map are tried in a failover scenario.

This feature includes the following capabilities:

- Default peer configuration
- IPSec idle-timer usage with default peer

If there is a default peer, the next time a connection is initiated, the connection is directed to the default peer instead of to the next peer in the peer list. If all connections to the current peer time out, the next time a connection is initiated it is directed to the default peer.

Feature History for IPSec Preferred Peer

Release	Modification
12.3(14)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- **Prerequisites for IPSec Preferred Peer, page 1174**
- **Restrictions for IPSec Preferred Peer, page 1174**
- **Information About IPSec Preferred Peer, page 1174**
- **How to Configure IPSec Preferred Peer, page 1176**
- **Configuration Examples for IPSec Preferred Peer, page 1179**
- **Additional References, page 1179**
- **Command Reference, page 1180**
- **Glossary, page 1181**

Prerequisites for IPSec Preferred Peer

- You must have a properly defined, complete crypto map. For detailed instructions, see "Configuring IPSec Network Security" in *Cisco IOS Security Configuration*.

Restrictions for IPSec Preferred Peer

Default peer:

- This feature must be used in conjunction with Dead Peer Detection (DPD). It is most effective on a remote site running DPD in periodic mode. DPD detects the failure of a device quickly and resets the peer list so that the default peer is tried for the next attempted connection.
- Only one peer can be designated as the default peer in a crypto map.
- The default peer must be the first peer in the peer list.

IPSec idle-timer usage with default peer:

- This feature works only on the crypto map for which it is configured. You cannot configure the capability globally for all crypto maps.
- If there is a global idle timer, the crypto map idle-timer value must be different from the global value; otherwise the idle timer is not added to the crypto map.

Information About IPSec Preferred Peer

To configure IPSec Preferred Peer, you need to understand the following concepts:

- IPSec, page 1174
- Dead Peer Detection, page 1175
- Default Peer Configuration, page 1175
- Idle Timers, page 1176
- IPSec Idle-Timer Usage with Default Peer, page 1176
- Peers on Crypto Maps, page 1176

IPSec

IPSec is a framework of open standards developed by the Internet Engineering Task Force (IETF). IPSec provides security for transmission of sensitive information over unprotected networks such as the Internet. IPSec acts at the network layer, protecting and authenticating IP packets between participating IPSec devices ("peers"), such as Cisco routers.

IPSec provides the following network security services. These services are optional. In general, local security policy dictates the use of one or more of these services:

- Data Confidentiality—The IPSec sender can encrypt packets before transmitting them across a network.
- Data Integrity—The IPSec receiver can authenticate packets sent by the IPSec sender to ensure that the data has not been altered during transmission.
- Data Origin Authentication—The IPSec receiver can authenticate the source of the IPSec packets sent.
- Anti-Replay—The IPSec receiver can detect and reject replayed packets.

With IPSec, data can be transmitted across a public network without fear of observation, modification, or spoofing. This enables applications such as Virtual Private Networks (VPNs), including intranets, extranets, and remote user access.

IPSec provides secure tunnels between two peers, such as two routers. You define which packets are considered sensitive and should be sent through these secure tunnels, and you define the parameters that should be used to protect these sensitive packets, by specifying characteristics of these tunnels. When the IPSec peer sees such a sensitive packet, it sets up the appropriate secure tunnel and sends the packet through the tunnel to the remote peer.

Dead Peer Detection

The VPN Client uses a keepalive mechanism called Dead Peer Detection (DPD) to check the availability of the VPN device on the other side of an IPSec tunnel. If the network is unusually busy or unreliable, you can increase the number of seconds that the VPN Client will wait before deciding whether the peer is no longer active.

Keepalive packets are not sent if traffic is received. This lowers the overhead associated with DPD, because on a heavily loaded network very few keepalive packets will be sent because traffic is being received on the tunnels. In addition, DPD sends keepalive packets only if there is user traffic to send (and no user traffic is received).

You can configure IKE DPD so that DPD sends the keepalive packets whether or not there is outbound user data. That is, as long as there is no inbound user data, the keepalive packets are sent at the configured keepalive interval.

For more information about dead peer detection, see the *Cisco IOS Security Configuration* manual.

Default Peer Configuration

If a connection timeout occurs, the connection to the current peer is closed. The **set peer** command allows you to configure the first peer as the default peer. If there is a default peer, the next time a connection is initiated, the connection is directed to the default peer instead of to the next peer in the peer list. If the default peer is unresponsive, the next peer in the peer list becomes the current peer and future connections through the crypto map try that peer.

This capability is useful when traffic on a physical link stops due to the failure of a remote peer. DPD indicates that the remote peer is unavailable, but that peer remains the current peer.

A default peer facilitates the failover to a preferred peer that was previously unavailable, but has returned to service. Users can give preference to certain peers in the event of a failover. This is useful if the original failure was due to a network connectivity problem rather than failure of the remote peer.

To configure a default peer, see the "Configuring a Default Peer" section on page 1177.

Idle Timers

When a router running Cisco IOS software creates an IPSec security association (SA) for a peer, resources must be allocated to maintain the SA. The SA requires both memory and several managed timers. For idle peers, these resources are wasted. If enough resources are wasted by idle peers, the router could be prevented from creating new SAs with other peers.

IPSec SA idle timers increase the availability of resources by deleting SAs associated with idle peers. Because IPSec SA idle timers prevent the wasting of resources by idle peers, more resources are available to create new SAs when required.

If IPSec SA idle timers are not configured, only the global lifetimes for IPSec SAs are applied. SAs are maintained until the global timers expire, regardless of peer activity.

IPSec Idle-Timer Usage with Default Peer

If all connections to the current peer time out, the next time a connection is initiated it is directed to the default peer configured in the **set peer** command. If a default peer is not configured and there is a connection timeout, the current peer remains the one that timed out.

This enhancement helps facilitate a failover to a preferred peer that was previously unavailable but is in service now.

To configure an IPSec idle-timer, see the "Configuring the Idle Timer" section on page 1178.

Peers on Crypto Maps

A crypto map set can contain multiple entries, each with a different access list. The router searches the crypto map entries in order, and attempts to match the packet to the access list specified in that entry.

When a packet matches a **permit** entry in a particular access list, and the corresponding crypto map entry is tagged as Cisco, connections are established with the remote peer as specified in the set peer statements within the crypto map.

For more information about crypto maps, see the *Cisco IOS Security Configuration* manual.

How to Configure IPSec Preferred Peer

This section contains the following procedures:

- Configuring a Default Peer, page 1177 (required)
- Configuring the Idle Timer, page 1178 (optional)

Configuring a Default Peer

To configure a default peer, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* [**ipsec-isakmp**] [**dynamic** *dynamic-map-name*] [**discover**] [**profile** *profile-name*]
4. **set peer** {*host-name* [**dynamic**] [**default**] | *ip-address* [**default**] }
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* [**ipsec-isakmp**] [**dynamic** *dynamic-map-name*] [**discover**] [**profile** *profile-name*] **Example:** `Router(config)# crypto map mymap 10 ipsec-isakmp`	Enters crypto map configuration mode. Creates or modifies a crypto map entry, creates a crypto profile that provides a template for configuration of dynamically created crypto maps, or configures a client accounting list.
Step 4	**set peer** {*host-name* [**dynamic**] [**default**] | *ip-address* [**default**] } **Example:** `Router(config-crypto-map)# set peer 10.0.0.2 default`	Specifies an IPSec peer in a crypto map entry. Ensures that the first peer specified is defined as the default peer.
Step 5	**exit** **Example:** `Router(config-crypto-map)# exit`	Exits crypto map configuration mode and returns to global configuration mode.

Configuring the Idle Timer

To configure the idle timer, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* [**ipsec-isakmp**] [**dynamic** *dynamic-map-name*] [**discover**] [**profile** *profile-name*]
4. **set security-association idletime** *seconds* [**default**]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* [**ipsec-isakmp**] [**dynamic** *dynamic-map-name*] [**discover**] [**profile** *profile-name*] **Example:** `Router(config)# crypto map mymap 10 ipsec-isakmp`	Enters crypto map configuration mode. Creates or modifies a crypto map entry, creates a crypto profile that provides a template for configuration of dynamically created crypto maps, or configures a client accounting list.
Step 4	**set security-association idletime** *seconds* [**default**] **Example:** `Router(config-crypto-map)# set security-association idletime 120 default`	Specifies the maximum amount of time for which the current peer can be idle before the default peer is used.
Step 5	**exit** **Example:** `Router(config-crypto-map)# exit`	Exits crypto map configuration mode and returns to global configuration mode.

Configuration Examples for IPSec Preferred Peer

- Configuring a Default Peer: Example, page 1179
- Configuring the IPSec Idle Timer: Example, page 1179

Configuring a Default Peer: Example

The following example shows that the first peer, at IP address 1.1.1.1, is the default peer:

```
crypto map tohub 1 ipsec-isakmp
 set peer 1.1.1.1 default
 set peer 2.2.2.2
```

Configuring the IPSec Idle Timer: Example

In the following example, if the current peer is idle for 120 seconds, the default peer 1.1.1.1 (which was specified in the **set peer** command) is used for the next attempted connection:

```
crypto map tohub 1 ipsec-isakmp
 set peer 1.1.1.1 default
 set peer 2.2.2.2
 set security-association idletime 120 default
```

Additional References

The following sections provide references related to IPSec Preferred Peer.

Related Documents

Related Topic	Document Title
IPSec	*Cisco IOS Security Configuration*, Release 12.3 *Cisco IOS Security Command Reference*, Release 12.3T
Crypto map	*Cisco IOS Security Configuration*, Release 12.3 *Cisco IOS Security Command Reference*, Release 12.3T
DPD	*IPSec Dead Peer Detection Periodic Message Option*, Release 12.3(7)T *Cisco IOS Security Configuration*, Release 12.3

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **set peer (IPSec)**
- **set security-association idletime**

Glossary

crypto access list—A list that defines which IP traffic will be protected by crypto and which traffic will not be protected by crypto.

crypto map—A map that specifies which traffic should be protected by IPSec, where IPSec-protected traffic should be sent, and what IPSec transform sets should be applied to this traffic.

dead peer detection—A feature that allows the router to detect an unresponsive peer.

keepalive message—A message sent by one network device to inform another network device that the virtual circuit between the two is still active.

peer—Router or other device that participates in IPSec and IKE. In IPSec, peers are devices or entities that communicate securely either through the exchange of keys or the exchange of digital certificates.

SA—Security association. An instance of security policy and keying material applied to a data flow. Both IKE and IPSec use SAs, although SAs are independent of one another. IPSec SAs are unidirectional and are unique in each security protocol. An IKE SA is used by IKE only, and unlike the IPSec SA, it is bidirectional. IKE negotiates and establishes SAs on behalf of IPSec. A user also can establish IPSec SAs manually. A set of SAs are needed for a protected data pipe, one per direction per protocol. For example, if you have a pipe that supports Encapsulating Security Payload (ESP) between peers, one ESP SA is required for each direction. SAs are identified uniquely by destination (IPSec endpoint) address, security protocol (AH or ESP), and security parameter index (SPI).

transform set—An acceptable combination of security protocols, algorithms, and other settings to apply to IPSec-protected traffic. During the IPSec security association negotiation, the peers agree to use a particular transform set when protecting a particular data flow.

Note Refer to *Internetworking Terms and Acronyms* for terms not included in this glossary.

IPSec Security Association Idle Timers

When a router running the Cisco IOS software creates an IPSec security association (SA) for a peer, resources must be allocated to maintain the SA. The SA requires both memory and several managed timers. For idle peers, these resources are wasted. If enough resources are wasted by idle peers, the router could be prevented from creating new SAs with other peers. The IPSec Security Association Idle Timers feature introduces a configurable idle timer to monitor SAs for activity, allowing SAs for idle peers to be deleted. Benefits of this feature include:

- Increased availability of resources
- Improved scalability of Cisco IOS IPSec deployments

Feature Specifications for IPSec Security Association Idle Timers

Feature History	
Release	**Modification**
12.2(15)T	This feature was introduced.
Supported Platforms	
Cisco 1700 series access routers, Cisco 2400 series integrated access devices, Cisco 2600 series multiservice platforms, Cisco 3600 series multiservice platforms, Cisco 3700 series multiservice access routers, Cisco 7100 series VPN routers, Cisco 7200 series routers, Cisco 7400 series routers, Cisco 7500 series routers, Cisco 801–804 ISDN routers, Cisco 805 serial router, Cisco 806 broadband router, Cisco 811, Cisco 813, Cisco 820, Cisco 827 ADSL router, Cisco 828 G.SHDSL router, Cisco 8850-RPM, Cisco 950, Cisco AS5350 universal gateway, Cisco AS5400 series universal gateways, Cisco integrated communications system 7750, Cisco MC3810 series multiservice access concentrators, Cisco ubr7200, Cisco ubr900 series cable access routers	

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for IPSec Security Association Idle Timers, page 1184
- Information About IPSec Security Association Idle Timers, page 1184
- Information About IPSec Security Association Idle Timers, page 1184
- How to Configure IPSec Security Association Idle Timers, page 1185
- Configuration Examples for IPSec Security Association Idle Timers, page 1187
- Additional References, page 1187
- Command Reference, page 1189

Prerequisites for IPSec Security Association Idle Timers

You must configure Internet Key Exchange (IKE) as described in the "Configuring Internet Key Exchange Security Protocol" chapter of the *Cisco IOS Security Configuration Guide*, Release 12.2.

Information About IPSec Security Association Idle Timers

To configure the IPSec Security Association Idle Timers feature, you must understand the following concepts:

- Lifetimes for IPSec Security Associations, page 1184
- IPSec Security Association Idle Timers, page 1184
- Benefits of IPSec Security Association Idle Timers, page 1185

Lifetimes for IPSec Security Associations

The Cisco IOS software currently allows the configuration of lifetimes for IPSec SAs. Lifetimes can be configured globally or per crypto map. There are two lifetimes: a "timed" lifetime and a "traffic-volume" lifetime. A security association expires after the first of these lifetimes is reached.

IPSec Security Association Idle Timers

The IPSec SA idle timers are different from the global lifetimes for IPSec SAs. The expiration of the global lifetime is independent of peer activity. The IPSec SA idle timer allows SAs associated with inactive peers to be deleted before the global lifetime has expired.

If the IPSec SA idle timers are not configured, only the global lifetimes for IPSec SAs are applied. SAs are maintained until the global timers expire, regardless of peer activity.

Note If the last IPSec SA to a given peer is deleted due to idle timer expiration, the Internet Key Exchange (IKE) SA to that peer will also be deleted.

Benefits of IPSec Security Association Idle Timers

Increased Availability of Resources

Configuring the IPSec Security Association Idle Timers feature increases the availability of resources by deleting SAs associated with idle peers.

Improved Scalability of Cisco IOS IPSec Deployments

Because the IPSec Security Association Idle Timers feature prevents the wasting of resources by idle peers, more resources will be available to create new SAs as required.

How to Configure IPSec Security Association Idle Timers

- Configuring the IPSec SA Idle Timer Globally, page 1185
- Configuring the IPSec SA Idle Timer per Crypto Map, page 1186

Configuring the IPSec SA Idle Timer Globally

This task configures the IPSec SA idle timer globally. The idle timer configuration will be applied to all SAs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto set security-association idle-time** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto set security-association idle-time** *seconds* **Example:** `Router(config)# crypto set security-association idle-time 600`	Configures the IPSec SA idle timer. • The *seconds* argument specifies the time, in seconds, that the idle timer will allow an inactive peer to maintain an SA. Valid values for the *seconds* argument range from 60 to 86400.

Configuring the IPSec SA Idle Timer per Crypto Map

This task configures the IPSec SA idle timer for a specified crypto map. The idle timer configuration will be applied to all SAs under the specified crypto map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-number* **ipsec-isakmp**
4. **crypto set security-association idle-time** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-number* **ipsec-isakmp** **Example:** `Router(config)# crypto map test 1 ipsec-isakmp`	Creates or modifies a crypto map entry and enters crypto map configuration mode.
Step 4	**crypto set security-association idle-time** *seconds* **Example:** `Router(config-crypto-map)# crypto set security-association idle-time 600`	Configures the IPSec SA idle timer. • The *seconds* argument specifies the time, in seconds, that the idle timer will allow an inactive peer to maintain an SA. Valid values for the *seconds* argument range from 60 to 86400.

Configuration Examples for IPSec Security Association Idle Timers

- Configuring the IPSec SA Idle Timer Globally Example, page 1187
- Configuring the IPSec SA Idle Timer per Crypto Map Example, page 1187

Configuring the IPSec SA Idle Timer Globally Example

The following example globally configures the IPSec SA idle timer to drop SAs for inactive peers after 600 seconds:

```
crypto set security-association idle-time 600
```

Configuring the IPSec SA Idle Timer per Crypto Map Example

The following example configures the IPSec SA idle timer for the crypto map named test to drop SAs for inactive peers after 600 seconds:

```
crypto map test 1 ipsec-isakmp
 crypto set security-association idle-time 600
```

Additional References

For additional information related to IPSec Security Association Idle Timers, see the following sections:

- Related Documents, page 1188
- Standards, page 1188
- MIBs, page 1188
- RFCs, page 1189
- Technical Assistance, page 1189

Related Documents

Related Topic	Document Title
Additional information about configuring IKE	"Configuring Internet Key Exchange Security Protocol" chapter of the *Cisco IOS Security Configuration Guide*, Release 12.2
Additional information about configuring global lifetimes for IPSec SAs	"Configuring IPSec Network Security" chapter ofthe *Cisco IOS Security Configuration Guide*, Release 12.2
Additional Security commands	*Cisco IOS Security Command Reference*, Release 12.2 T

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **crypto set security-association idle-time**

IPSec—SNMP Support

Feature History

Release	Modification
12.1(4)E	This feature was introduced on the Cisco 7100, 7200, and 7500 series.
12.1(5a)E	Support for CISCO-IPSEC-FLOW-MONITOR-MIB notifications was added.
12.2(4)T	Support for this feature was added for platforms in Release 12.2 T.
12.2(8)T, 12.1(11b)E	The following Command Line Interface (CLI) commands were added to enable and disable IP Security (IPSec) MIB notifications: • **snmp-server enable traps ipsec** • **snmp-server enable traps isakmp**
12.2(14)S	This feature was integrated into Cisco IOS Release 12.2(14)S.

This document describes the IPSec—SNMP Support feature in Cisco IOS Release 12.1 E, 12.2 T, and 12.2 S and includes the following sections:

- Feature Overview, page 1192
- Supported Platforms, page 1193
- Supported Standards, MIBs, and RFCs, page 1195
- Configuration Tasks, page 1195
- Monitoring and Maintaining IPSec MIB, page 1197
- Configuration Examples, page 1197
- Command Reference, page 1198
- Glossary, page 1199

Note This document focuses on Cisco IOS CLI support for the Cisco IPSec MIBs. This document also lists which elements of the MIBs are currently supported. This document does not describe SNMP configuration (from a Network Management Station) of the Cisco IPSec MIBs.

Feature Overview

The IP Security (IPSec) - SNMP Support feature introduces support for industry-standard IPSec MIBs and Cisco IOS-software specific IPSec MIBs.

The IPSec MIBs allow IPSec configuration monitoring and IPSec status monitoring using SNMP, and can be integrated in a variety of Virtual Private Network (VPN) management solutions.

For example, this feature allows you to specify the desired size of a tunnel history table or a tunnel failure table using the Cisco IOS CLI. The history table archives attribute and statistic information about the tunnel; the failure table archives tunnel failure reasons along with the time failure occurred. A failure history table can be used as a simple method to distinguish between a normal and an abnormal tunnel termination. That is, if a tunnel entry in the tunnel history table has no associated failure record, the tunnel must have terminated normally. However, a tunnel history table does not accompany every failure table because every failure does not correspond to a tunnel. Thus, supported setup failures are recorded in the failure table, but an associated history table is not recorded because a tunnel was never set up.

This feature also provides IPSec Simple Network Management Protocol (SNMP) notifications for use with network management systems.

Benefits

The commands in this feature allow you to examine the version of the IPSec MIB feature, to enable and disable SNMP traps, and to monitor and control the size of the buffers used by this feature.

Restrictions

Only the following tunnel setup failure logs are supported with the IPSec - SNMP Support feature:

- NOTIFY_MIB_IPSEC_PROPOSAL_INVALID

  “A tunnel could not be established because the peer did not supply an acceptable proposal.”
- NOTIFY_MIB_IPSEC_ENCRYPT_FAILURE

  “A tunnel could not be established because it failed to encrypt a packet to be sent to a peer.”
- NOTIFY_MIB_IPSEC_SYSCAP_FAILURE

  “A tunnel could not be established because the system ran out of resources.”
- NOTIFY_MIB_IPSEC_LOCAL_FAILURE

  “A tunnel could not be established because of an internal error.”

Note that these failure notices are recorded in the failure tables, but are not available as SNMP notifications (traps).

The following functions are not supported with the IPSec MIB feature:

- Checkpointing
- The Dynamic Cryptomap table of the CISCO-IPSEC-MIB

Note CISCO-IPSEC-FLOW-MONITOR-MIB notifications are not supported before Cisco IOS Release 12.1(5a)E.

The CISCO-IPSEC-POLICY-MAP-MIB (ciscoIpSecPolMap) defines no notifications (the "IPSec Policy Map Notifications Group" is empty).

Related Features and Technologies

The IPSec—SNMP Support feature was designed to support the VPN Device Manager (VDM). VDM enables network administrators to manage and configure site-to-site VPNs on a single device from a web browser and to see the effects of changes in real time. VDM implements a wizard-based graphical user interface (GUI) to simplify the process of configuring site-to-site VPNs using the IPSec protocol. VDM software is installed directly on Cisco VPN routers, and is designed for use and compatibility with future Device Manager products.

For more information on Cisco VDM, refer to the following URL:

http://www.cisco.com/warp/public/cc/pd/nemnsw/vpdvmn/

Related Documents

IPSec and Related Security Information

- *Cisco IOS Security Configuration Guide*
- *Cisco IOS Security Command Reference*

SNMP Configuration Information

- *Cisco IOS Configuration Fundamentals Configuration Guide*
- *Cisco IOS Configuration Fundamentals Command Reference*

For the Cisco IOS Release12.1 E implementation of security and SNMP features, refer to the Cisco IOS Release 12.1 versions of these documents. For Cisco IOS Release 12.2 T and 12.2 S implementation of these features, refer to the Cisco IOS Release 12.2 versions of these documents.

Supported Platforms

The IPSec MIB feature is supported on the following platforms in Cisco IOS Release 12.1(4)E:

- Cisco 7100 series
- Cisco 7200 series
- Cisco 7500 series (RSP7000 and 7500)

The IPSec MIB feature is supported on the following platforms in Cisco IOS Release 12.2(4)T:

- Cisco 800 series (800, 805, 806, 820, 827, 828)
- Cisco 900 series
- Cisco 1600 and 1600R series
- Cisco 1700 series (1710, 1720, 1750, 1751, 1760)
- Cisco 2400 series
- Cisco 2600 and 2600XM series
- Cisco 3600 series (Cisco 3620, 3640, and 3660)

- Cisco 3745
- Cisco 4000
- Cisco 4500
- Cisco 5300 series
- Cisco 5400 series
- Cisco 5800 series
- Cisco 7100 series
- Cisco 7200 series
- Cisco 7400 series
- Cisco 7500 series (Cisco IOS Release 12.2(4)T2 and later releases)
- Cisco 7700 series
- Cisco MC3810
- Cisco uBR900 series (uBR900, uBR904, uBR905, uBR910, uBR920, uBR925
- Cisco uBR7200

The IPSec MIB feature is supported on the following platforms in Cisco IOS Release 12.2(14)S:

- Cisco 7200 series
- Cisco 7400 series
- Cisco 7500 series

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that are supported on specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

The following MIBs are supported by the IPSec—SNMP Support feature:

- CISCO-IPSEC-FLOW-MONITOR- MIB
- CISCO-IPSEC-MIB
- CISCO-IPSEC-POLICY-MAP-MIB

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

No new or modified RFCs are supported by this feature.

Configuration Tasks

See the following sections for configuration tasks for the IPSec—SNMP Support feature. Each task in the list is identified as either required or optional:

- Enabling IPSec SNMP Notifications (required)
- Configuring IPSec Failure History Table Size (optional)
- Configuring IPSec Tunnel History Table Size (optional)

Enabling IPSec SNMP Notifications

To enable a router to send IPSec trap or inform notifications to a specified host, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# snmp-server enable traps ipsec cryptomap [add | delete | attach | detach]`	Enables a router to send IPSec SNMP notifications.
Step 2	`Router(config)# snmp-server enable traps isakmp [policy {add | delete} | tunnel {start | stop}]`	Enables a router to send IPSec ISAKMP SNMP notifications.
Step 3	`Router(config)# snmp-server host host-address traps community-string ipsec`	Specifies the recipient of IPSec SNMP notification operations.

For more information on configuring SNMP, refer to the chapter "Configuring SNMP Support" in the *Cisco IOS Configuration Fundamentals Configuration Guide.*

Configuring IPSec Failure History Table Size

The default failure history table size is 200. To change the size of the failure history table, use the following command in global configuration mode:

Command	Purpose
`Router(config)# crypto mib ipsec flowmib history failure size number`	Changes the size of the IPSec failure history table.

Configuring IPSec Tunnel History Table Size

The default tunnel history table size is 200. To change the size of the tunnel history table, use the following command in global configuration mode:

Command	Purpose
`Router(config)# crypto mib ipsec flowmib history tunnel size number`	Changes the size of the IPSec tunnel history table.

Verifying IPSec MIB Configuration

To verify that the IPSec MIB feature is configured properly, perform the following tasks:

- Enter the **show crypto mib ipsec flowmib history failure size** privileged EXEC command to display the size of the failure history table:

```
Router# show crypto mib ipsec flowmib history failure size
IPSec Failure Window Size: 140
```

- Enter the **show crypto mib ipsec flowmib history tunnel size** privileged EXEC command to display the size of the tunnel history table:

```
Router# show crypto mib ipsec flowmib history tunnel size
IPSec History Window Size: 130
```

- Enter the **show crypto mib ipsec flowmib version** privileged EXEC command to display the MIB version used by the management applications to identify the feature set:

```
Router# show crypto mib ipsec flowmib version
IPSec Flow MIB version: 1
```

- Enter the **debug crypto mib** command to display the IPSec MIB debug message notifications:

```
Router# debug crypto mib
Crypto IPSec Mgmt Entity debugging is on
```

Monitoring and Maintaining IPSec MIB

To monitor the status of IPSec MIB information, use any of the following commands in EXEC mode:

Command	Purpose
`Router# show crypto mib ipsec flowmib history failure size`	Displays the size of the IPSec failure history table.
`Router# show crypto mib ipsec flowmib history tunnel size`	Displays the size of the IPSec tunnel history table.
`Router# show crypto mib ipsec flowmib version`	Displays the IPSec Flow MIB version used by the router.

Configuration Examples

This section provides the following configuration examples:

- Enabling IPSec Notifications Examples
- Specifying History Table Size Examples

Enabling IPSec Notifications Examples

In the following example, IPSec notifications are enabled:

```
snmp-server enable traps ipsec isakmp
```

In the following example, the router is configured to send IPSec notifications to the host nms1.cisco.com:

```
snmp-server host nms1.cisco.com public ipsec isakmp
Translating "nms1.cisco.com"...domain server (171.00.0.01) [OK]
```

Specifying History Table Size Examples

In the following example, the specified failure history table size is 140:

```
crypto mib ipsec flowmib history failure size 140
```

In the following example, the specified tunnel history table size is 130:

```
crypto mib ipsec flowmib history tunnel size 130
```

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **crypto mib ipsec flowmib history failure size**
- **crypto mib ipsec flowmib history tunnel size**
- **debug crypto mib**
- **show crypto mib ipsec flowmib history failure size**
- **show crypto mib ipsec flowmib history tunnel size**
- **show crypto mib ipsec flowmib version**
- **snmp-server enable traps ipsec**
- **snmp-server enable traps isakmp**
- **snmp-server host**

Glossary

CA—certificate authority. A certificate authority (CA) is an entity in a network that issues and manages security credentials and public keys (in the form of X509v3 certificates) for message encryption. As part of a public key infrastructure (PKI), a CA checks with a registration authority (RA) to verify information provided by the requestor of a digital certificate. If the RA verifies the requestor's information, the CA can then issue a certificate. Certificates generally include the owner's public key, the expiration date of the certificate, the owner's name, and other information about the public key owner.

IP Security—See IPSec.

IPSec—Internet Protocol Security. A framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer. IPSec uses Internet Key Exchange (IKE) to handle negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPSec. IPSec can be used to protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

Management Information Base—See MIB.

MIB—Management Information Base. Database of network management information that is used and maintained by a network management protocol such as Simple Network Management Protocol (SNMP) or Common Management Information Protocol (MIP). The value of a MIB object can be changed or retrieved using SNMP or CMIP commands, usually through a graphical user interface (GUI) network management system (NMS). MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches.

Simple Network Management Protocol—See SNMP.

SNMP—Simple Network Management Protocol. An application-layer protocol that provides a message format for communication between SNMP managers and agents.

trap—Message sent by an SNMP agent to a network management system, console, or terminal to indicate the occurrence of a significant event, such as a specifically defined condition or a threshold that was reached.

IPSec Virtual Tunnel Interface

IPSec virtual tunnel interfaces (VTI) provide a routable interface type for terminating IPSec tunnels an easy way to define protection between sites to form an overlay network. IPSec virtual tunnel interfacees simplify configuration of IPSec for protection of remote links, supports multicast, and simplifies network management and load balancing.

History for the IPSec Virtual Tunnel Interface Feature

Release	Modification
12.3(14)T	This feature was introduced.
12.4(2)T	Dynamic Virtual Tunnel Interfaces were added.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Restrictions for IPSec Virtual Tunnel Interface, page 1202
- Information About IPSec Virtual Tunnel Interfaces, page 1202
- How to Configure IPSec Virtual Tunnels, page 1206
- Configuration Examples for IPSec Virtual Tunnel Interfaces, page 1210
- Additional References, page 1214
- Command Reference, page 1215

Restrictions for IPSec Virtual Tunnel Interface

Stateful Failover

IPSec stateful failover is not supported with IPSec virtual tunnel interfaces.

Proxy

Only strict IP ANY ANY proxy is supported.

IPSec Transform Set

The IPSec transform set must be configured in tunnel mode only.

IKE Security Association

The Internet Key Exchange (IKE) security association (SA) is bound to the virtual tunnel interface. Because IKE SA is bound to the virtual tunnel interface, the same IKE SA cannot be used for a crypto map.

VTI versus GRE Tunnels

The IPSec virtual tunnel interface is limited to IP unicast and multicast traffice only, as opposed to GRE tunnels, which have a wider application for IPSec implementation.

Information About IPSec Virtual Tunnel Interfaces

The IPSec virtual tunnel interface greatly simplifies the configuration process when you need to provide protection for remote access and provides an simpler alternative to using GRE or L2TP tunnels for encapsulation and crypto maps with IPSec. A major benefit associated with IPSec virtual tunnel interfaces is the reduction in overhead because the configuration does not require a static mapping of IPSec sessions to a physical interface: The IPSec VTI allows for the flexibility of sending and receiving both IP unicast and multicast encrypted traffic on any physical interface, such as in the case of multiple paths (multicast routing).

The following sections provide details about the IPSec virtual tunnel interface:

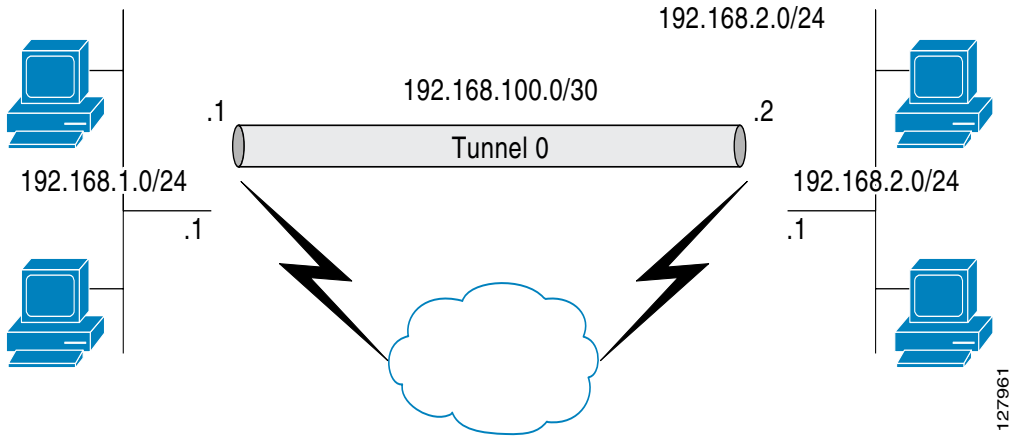
- Routing with IPSec Virtual Tunnel Interfaces, page 1202
- Traffic Encryption with the IPSec Virtual Tunnel Interface, page 1203
- IPSec Packet Flow, page 1203

Routing with IPSec Virtual Tunnel Interfaces

You can enable routing protocols on the tunnel interface so that routing information can be propagated over the virtual tunnel. The router can establish neighbor relationships over the virtual tunnel interface. Multicast packets can be encrypted, and interoperability with standard-based IPSec installations is possible through the use of IP ANY ANY proxy. The static IPSec interface, will negotiate and accept **permit IP ANY ANY** proxies.

Figure 82 illustrates how a static virtual tunnel interface is used.

Figure 82 IPSec Static Virtual Tunnel Interface



The IPSec virtual tunnel interface supports native IPSec tunneling and exhibits most of the properties of a physical interface.

Traffic Encryption with the IPSec Virtual Tunnel Interface

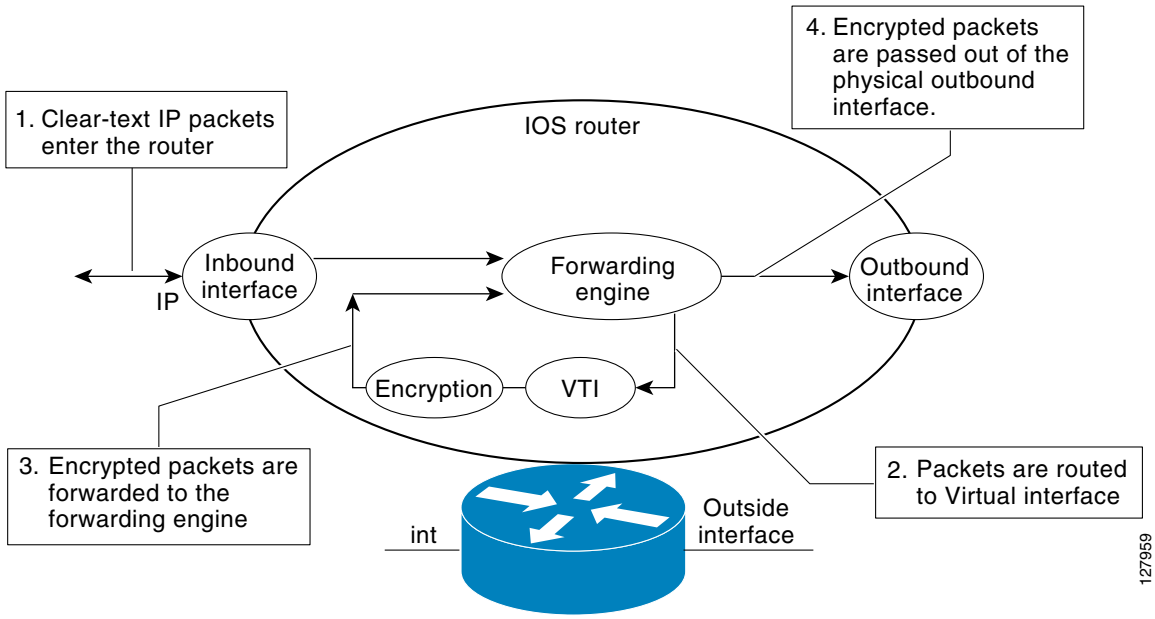
In the IPSec virtual tunnel interface encryption occurs in the tunnel. Traffic is encrypted when it is forwarded to the tunnel interface. Traffic forwarding is handled by the IP routing table, and dynamic or static IP routing can be used to route the traffic to the virtual tunnel interface. Using IP routing to forward the traffic to encryption simplifies the IPSec Virtual Private Network (VPN) configuration because the use of access control lists (ACLs) with a crypto map in native IPSec configurations not required. The IPSec virtual tunnel also allows you to encrypt multicast traffic with IPSec.

IPSec VTIs allow you to separate the interface context to apply pre- and post-encryption features. Features on the clear-text packets are configured on the VTI; Features for encrypted packets are applied on the physical outbound interface When IPSec virtual tunnel interfaces are used, you can separate application of Network Address Translation (NAT), ACLs, Quality of Service (QoS) and apply them to clear text or encrypted text, or both. When crypto maps are used, there is no easy way to specify forced encryption features.

IPSec Packet Flow

IPSec packet flow going into the IPSec tunnel is illustrated in Figure 83.

Figure 83 Packet Flow Going Into the IPSec Tunnel

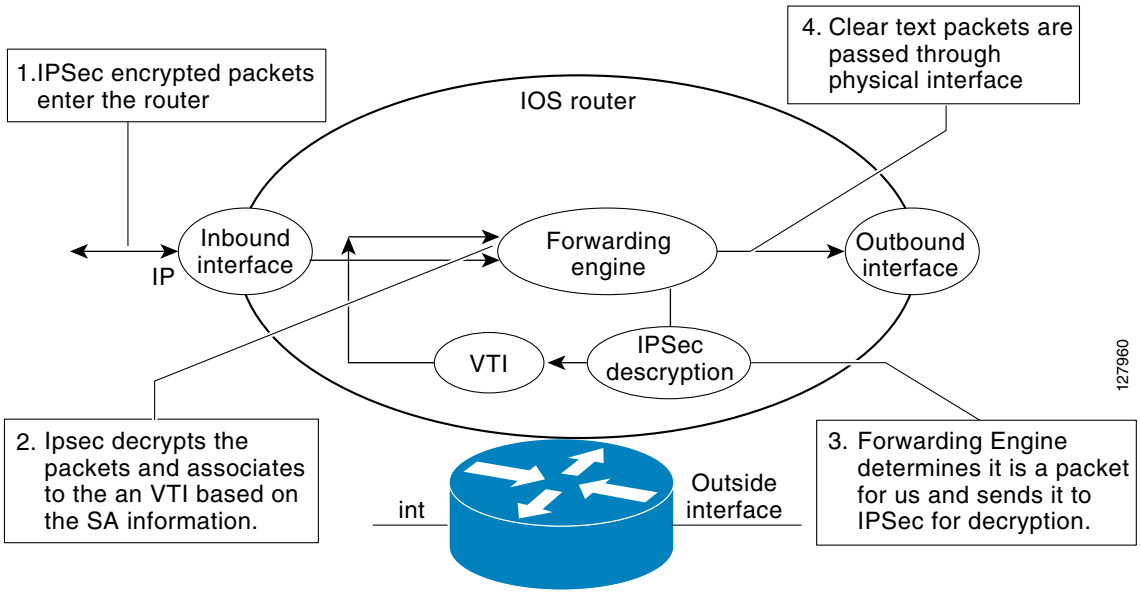


In Figure 83, shows the flow of packets in the egress path.

After packets arrive on the inbound interface, the forwarding engine switches the packets to the virtual tunnel interface where they are encrypted. The encrypted packets are handed back to the forwarding engine where they are switched through the outbound interface.

Figure 84 shows the packet flow out of the IPSec tunnel.

Figure 84 Packet Flow Out of the IPSec Tunnel



Dynamic Virtual Tunnel Interfaces

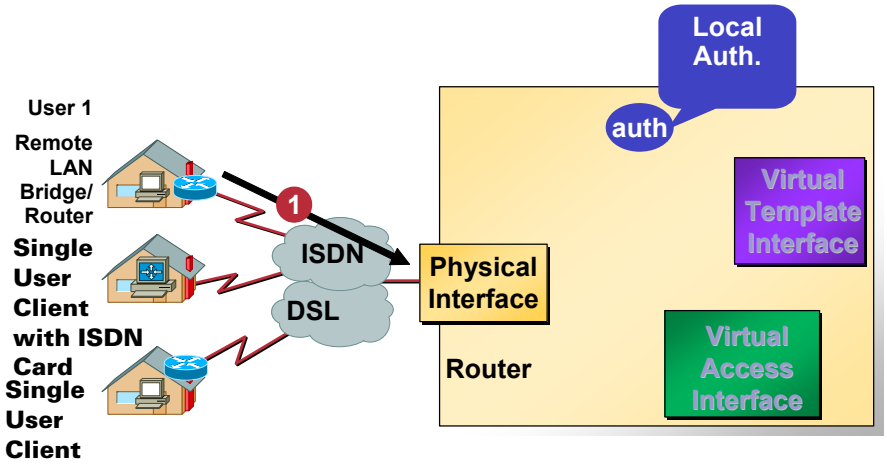
Dynmic VTIs provide efficiency in the use of IP addresses and provide secure connectivity. Dynamic VTIs allow dynamically downloadable per-group and per-user policies to be configured on a RADIUS server. The per-group or per-user definition can be created using Xauth User or Unity group, or it can be derived from a certificate. Dynamic VTI is standards-based, so interoperability in a multiple-vendor environment is supported. IPSec Dynamic VTIs allow you to create highly secure connectivity for remote access VPNs and can be combined with Cisco AVVID (Architecture for Voice, Video and Integrated Data) to deliver converged voice, video, and data over IP networks.The dynamic VTI simplifies VRF-aware IPSec deployment. The VRF is configured on the interface.

Quality of Service features can be used to improve the performance of various applications across the network. Traffic shaping is used between sites to limit the total amount of traffic that should be transmitted between two sites. Additionally, any combination of QoS features offered in Cisco IOS software can be used to support voice, video, or data applications.

Dynamic VTI reduces overhead by requiring minimal configuration on the router. A single virtual template can be configured and cloned, as opposed to the crypto requirement of one virtual template per VRF.

The dynamic VTI creates an interfce for IPSec sessions and uses the virtual template infrastructure for dynamic instantiation and management of IPSec interfaces. Dynamic VTIs ae used in hub-and-spoke configurations. A single dynamic VTI can support several staic VTIs. Decisions are made through routing updates. Figure 85 illustrates the dynamic VTI authentication path.

Figure 85 Dynamic IPSec Virtual Tunnel Interface



The authentication shown in Figure 85 follows this path:

1. User 1 calls the router.
2. Router 1 checks authentication locally.
3. Authentication succeeds.
4. Clones virtual access interface from virtual template Interface.

Profile Definitions and Policy Define the Dynamic Virtual Tunnel Interface Life Cycle

IPSec profiles define policy for dynamic VTIs. The dynamic interface is created at the end of IKE Phase 1. The IKE Phase 1.5 exchange is driven by the virtual template configuration in the ISAKAMP profile. The interface is deleted when the IPSec session to the peer is closed. The IPSec session is closed when both IKE and IPSec SAs to the peer are deletted.

How to Configure IPSec Virtual Tunnels

- Configuring IPSec Static Tunnels, page 1206
- Configuring Dynamic Virtual Tunnel Interfaces, page 1208

Configuring IPSec Static Tunnels

This configuration shows how to configure a static IPSec virtual tunnel interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec profile** *profile-name*
4. **set transform-set** *transform-set-name*
5. **interface** *type number*
6. **ip address** *address mask*
7. **tunnel mode** *mode*
8. **tunnel source** *interface*
9. **tunnel destination** *ip-address*
10. **tunnel protection ipsec profile** *profile-name* [**shared**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec profile** *profile-name* **Example:** `Router(config)# crypto ipsec profile PROF`	Defines the IP Security (IPSec) parameters that are to be used for IPSec encryption between two IPSec routers
Step 4	**set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*] **Example:** `Router(config)# set transform tset`	Specifies which transform sets can be used with the crypto map entry
Step 5	**interface** *type number* **Example:** `Router(config)# interface tunnel0`	Specifies the interface on which the tunnel will be configured and enters interface configuration mode.
Step 6	**ip address** *address mask* **Example:** `Router(config-if)# ip address 10.1.1.1 255.255.255.0`	Specifies the IP address and mask.
Step 7	**tunnel mode** *mode* **Example:** `Router(config-if)# tunnel mode ipsec ipv4`	Defines the mode for the tunnel.
Step 8	**tunnel source** *interface* **Example:** `Router(config-if)# tunnel source loopback0`	Specifies the tunnel source as a loopback interface.
Step 9	**tunnel destination** *ip-address* **Example:** `Router(config-if)# tunnel destination 172.1.1.1`	Identifies the IP address of the tunnel destination.
Step 10	**tunnel protection** ipsec profile *profile-name* [**shared**] **Example:** `Router(config-if)# tunnel protection ipsec profile PROF`	Associates a tunnel interface with an IP Security (IPSec) profile.

Configuring Dynamic Virtual Tunnel Interfaces

This configuration shows how to configure a dynamic IPSec virtual tunnel interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec profile** *profile-name*
4. **set transform-set** *transform-set-name*
5. **interface virtual-template** *number*
6. **ip unnumbered loopback** *number*
7. **tunnel mode** *mode*
8. **tunnel protection ipsec profile** *profile-name* [**shared**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec profile** *profile-name* **Example:** `Router(config)# crypto ipsec profile PROF`	Defines the IP Security (IPSec) parameters that are to be used for IPSec encryption between two IPSec routers
Step 4	**set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*] **Example:** `Router(config)# set transform tset`	Specifies which transform sets can be used with the crypto map entry
Step 5	**interface** *type number* **Example:** `Router(config)# interface tunnel0`	Specifies the interface on which the tunnel will be configured and enters interface configuration mode.
Step 6	**ip address** *address mask* **Example:** `Router(config-if)# ip address 10.1.1.1 255.255.255.0`	Specifies the IP address and mask.
Step 7	**tunnel mode** *mode* **Example:** `Router(config-if)# tunnel mode ipsec ipv4`	Defines the mode for the tunnel.
Step 8	**tunnel source** *interface* **Example:** `Router(config-if)# tunnel source loopback0`	Specifies the tunnel source as a loopback interface.
Step 9	**tunnel destination** *ip-address* **Example:** `Router(config-if)# tunnel destination 172.1.1.1`	Identifies the IP address of the tunnel destination.
Step 10	**tunnel protection** ipsec profile *profile-name* [**shared**] **Example:** `Router(config-if)# tunnel protection ipsec profile PROF`	Associates a tunnel interface with an IP Security (IPSec) profile.

Configuration Examples for IPSec Virtual Tunnel Interfaces

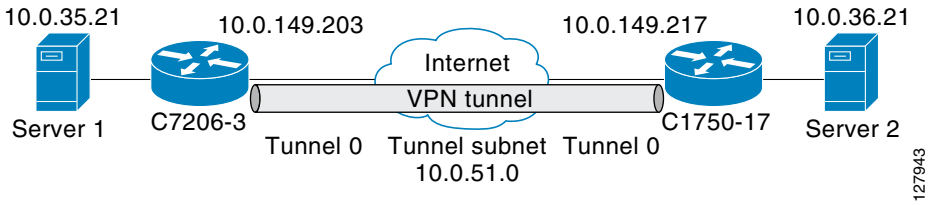
The following examples are provided to illustrate configuration scenarios for IPSec virtual tunnel interfaces:

- Static Virtual Tunnel Interface with IPSec: Example, page 1210
- Dynamic Virtual Tunnel Interface with IPSec for Simple Hub-and-Spoke Configuration: Example

Static Virtual Tunnel Interface with IPSec: Example

The following example configuration uses a preshared key for authentication between peers. VPN traffic is forwarded to the IPSec virtual tunnel interface for encryption and then sent out of the physical interface. The tunnel on subnet 10 checks packets for IPSec policy and passes them to the Crypto Engine (CE) for IPSec encapsulation. Figure 86 illustrates the IPSec VTI configuration.

Figure 86 Virtual Tunnel Interface with IPSec



C7206 Router Configuration

```
version 12.3

service timestamps debug datetime
service timestamps log datetime
hostname 7200-3
no aaa new-model
ip subnet-zero
ip cef
controller ISA 6/1
!
crypto isakmp policy 1
encr 3des
authentication pre-share
group 2
crypto isakmp key Cisco12345 address 0.0.0.0 0.0.0.0
crypto ipsec transform-set T1 esp-3des esp-sha-hmac
crypto ipsec profile P1
set transform-set T1
!
interface Tunnel0
 ip address 10.0.51.203 255.255.255.0
 ip ospf mtu-ignore
 load-interval 30
 tunnel source 10.0.149.203
 tunnel destination 10.0.149.217
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile P1
!
interface Ethernet3/0
```

```
 ip address 10.0.149.203 255.255.255.0
 duplex full
!
interface Ethernet3/3
 ip address 10.0.35.203 255.255.255.0
 duplex full
!
ip classless
ip route 10.0.36.0 255.255.255.0 Tunnel0
line con 0
line aux 0
line vty 0 4
end
```

C1750 Router Configuration

```
version 12.3

hostname c1750-17
no aaa new-model
ip subnet-zero
ip cef
crypto isakmp policy 1
encr 3des
authentication pre-share
group 2

crypto isakmp key Cisco12345 address 0.0.0.0 0.0.0.0
crypto ipsec transform-set T1 esp-3des esp-sha-hmac
crypto ipsec profile P1
set transform-set T1
!
interface Tunnel0
 ip address 10.0.51.217 255.255.255.0
 ip ospf mtu-ignore
 tunnel source 10.0.149.217
 tunnel destination 10.0.149.203
 tunnel mode ipsec ipv
 tunnel protection ipsec profile P1
!
interface FastEthernet0/0
 ip address 10.0.149.217 255.255.255.0
 speed 100
 full-duplex
!
interface Ethernet1/0
 ip address 10.0.36.217 255.255.255.0
 load-interval 30
 full-duplex
!
ip classless
ip route 10.0.35.0 255.255.255.0 Tunnel0
line con 0
line aux 0
line vty 0 4
end
```

Verifying the Results for IPSec Virtual Tunnel Interface Example

This section provides information you can use to confirm that your configuration is working properly. In this display, Tunnel 0 is “up,” and the line protocol is “up.” If the line protocol is “down”, the session is not active.

Verifying the C7206 Status

```
7200-3#show interface tunnel 0

Tunnel0 is up, line protocol is up
Hardware is Tunnel
Internet address is 10.0.51.203/24
MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
reliability 255/255, txload 103/255, rxload 110/255
Encapsulation TUNNEL, loopback not set
Keepalive not set
Tunnel source 10.0.149.203, destination 10.0.149.217
Tunnel protocol/transport IPSEC/IP, key disabled, sequencing disabled
Tunnel TTL 255

Checksumming of packets disabled, fast tunneling enabled
Tunnel transmit bandwidth 8000 (kbps)
Tunnel receive bandwidth 8000 (kbps)
Tunnel protection via IPSec (profile "P1")
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Input queue: 1/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/0 (size/max)
30 second input rate 13000 bits/sec, 34 packets/sec
30 second output rate 36000 bits/sec, 34 packets/sec
191320 packets input, 30129126 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
59968 packets output, 15369696 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 output buffer failures, 0 output buffers swapped out

7200-3#show crypto session

Crypto session current status
Interface: Tunnel0
Session status: UP-ACTIVE
Peer: 10.0.149.217 port 500
IKE SA: local 10.0.149.203/500 remote 10.0.149.217/500 Active
IPSEC FLOW: permit ip 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
Active SAs: 4, origin: crypto map

7200-3#show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C 10.0.35.0/24 is directly connected, Ethernet3/3
S 10.0.36.0/24 is directly connected, Tunnel0
C 10.0.51.0/24 is directly connected, Tunnel0
C 10.0.149.0/24 is directly connected, Ethernet3/0
```

Dynamic Virtual Tunnel Interface with IPSec for Simple Hub-and-Spoke Configuration: Example

This example shows the basic configuration of a dynamic VTI for a simple hub-and-spoke network configuration.

```
enable
configure terminal
crypto isakmp profile red
 virtual -template 1
!
crypto ipsec profile red
 set transform-set red
!
interface virtual-template1 tunnel
 ip unnumbered loopback0
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile red
```

VRF-Aware Ipsec with Dynamic VTI: Example

This example shows how to configure VRF-Aware IPSec to take advantage of the dynamic VTI.

```
!
Crypto isakmp profile BLUE
 …
 virtual-template 1
!
Crypto isakmp profile RED
 …
 virtual-template 2
!
Interface virtual-template1 type tunnel
 ip vrf forwarding BLUE
 …
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile BLUE
!
Interface virtual-template1 type tunnel
 ip vrf forwarding RED
 …
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile RED
```

QoS Service Policy Per Instance with Dynamic VTI: Example

```
Policy-map map1
 class class-default
  shape average 8000
!
Crypto isakmp profile map1
 virtual-template 1
 .
 .
 .
```

```
!
Interface Virtual-Template1 type tunnel
 …
 service-policy output map1
!
```

Additional References

The following sections provide references related to IPSec virtual tunnel interface.

Related Documents

Related Topic	Document Title
IPSec, security issues	*Cisco IOS Security Configuration Guide*
Security commands	*Cisco IOS Security Command Reference*
VPN configuration	*Cisco IOS Easy VPN Server*

Standards

Standards	Title
No standards were	

MIBs

MIBs	MIBs Link
• No MIBs were created or modified to support this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2401	*Security Architecture for The Internet Protocol*
RFC 2408	*Internet Security Association and Key Management Protocol*
RFC 2409	*The Internet Key Exchange (IKE)*

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **crypto isakmp profile**
- **tunnel mode**
- **virtual-template**

IPSec VPN Accounting

The IPSec VPN Accounting feature allows for a session to be accounted for by indicating when the session starts and when it stops.

A VPN session is defined as an Internet Key Exchange (IKE) security association (SA) and the one or more SA pairs that are created by the IKE SA. The session starts when the first IP Security (IPSec) pair is created and stops when all IPSec SAs are deleted.

Session identifying information and session usage information is passed to the Remote Authentication Dial-In User Service (RADIUS) server via standard RADIUS attributes and vendor-specific attributes (VSAs).

Feature Specifications for IPSec VPN Accounting

Feature History	
Release	**Modification**
12.2(15)T	This feature was introduced.
Supported Platforms	
Cisco 2610–2613, Cisco 2620–Cisco 2621, Cisco 2650–Cisco 2651, Cisco 3620, Cisco 3640, Cisco 3660, Cisco 3725, Cisco 3745, Cisco 7100, Cisco 7200, Cisco 7400, Cisco ubr7100, Cisco ubr7200.	

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for IPSec VPN Accounting, page 1218
- Information About IPSec VPN Accounting, page 1218
- How to Configure IPSec VPN Accounting, page 1222
- Configuration Examples for IPSec VPN Accounting, page 1228
- Additional References, page 1232

- Command Reference, page 1233
- Glossary, page 1235

Prerequisites for IPSec VPN Accounting

You need to understand how to configure RADIUS and authentication, authorization, and accounting (AAA) accounting. For information about configuring RADIUS and AAA, refer to the following documents:

- *Configuring Basic AAA RADIUS for Dial-In Clients*
- *How Does RADIUS Work?*
- The chapter "Configuring RADIUS" in the *Cisco IOS Security Configuration Guide*
- The chapter "RADIUS Commands" in the *Cisco IOS Security Command Reference*, Release 12.2
- The chapter "Configuring Accounting" in the *Cisco IOS Security Configuration Guide*, Release 12.2

You also need to know how to configure IPSec accounting. For information about configuring IPSec accounting, refer to the chapter "Configuring IPSec Network Security" in the *Cisco IOS Security Configuration Guide*, Release 12.2.

Information About IPSec VPN Accounting

To configure IPSec VPN accounting, you must understand the following concepts:

- RADIUS Accounting, page 1218
- IKE and IPSec Subsystem Interaction, page 1220

RADIUS Accounting

For many large networks, it is required that user activity be recorded for auditing purposes. The method that is used most is RADIUS accounting.

RADIUS accounting allows for a session to be accounted for by indicating when the session starts and when it stops. Additionally, session identifying information and session usage information will be passed to the RADIUS server via RADIUS attributes and VSAs.

RADIUS Start Accounting

The RADIUS Start packet contains many attributes that generally identify who is requesting the service and of what the property of that service consists. Table 52 represents the attributes required for the start.

Table 52 RADIUS Accounting Start Packet Attributes

RADIUS Attributes Value	Attribute	Description
1	user-name	Username used in extended authentication (XAUTH).The username may be NULL when XAUTH is not used.
4	nas-ip-address	Identifying IP address of the network access server (NAS) that serves the user. It should be unique to the NAS within the scope of the RADIUS server.
5	nas-port	Physical port number of the NAS that serves the user.
8	framed-ip-address	Private address allocated for the IP Security (IPSec) session.
40	acct-status-type	Status type. This attribute indicates whether this accounting request marks the beginning (start), the end (stop), or an update of the session.
41	acct-delay-time	Number of seconds the client has been trying to send a particular record.
44	acct-session-id	Unique accounting identifier that makes it easy to match start and stop records in a log file.
26	vrf-id	String that represents the name of the Virtual Route Forwarder (VRF).
26	isakmp-initiator-ip	Endpoint IP address of the remote Internet Key Exchange (IKE) initiator (V4).
26	isakmp-group-id	Name of the VPN group profile used for accounting.
26	isakmp-phase1-id	Phase 1 identification (ID) used by IKE (for example, domain name [DN], fully qualified domain name [FQDN], IP address) to help identify the session initiator.

RADIUS Stop Accounting

The RADIUS Stop packet contains many attributes that identify the usage of the session. Table 2 represents the additional attributes required for the RADIUS stop packet. It is possible that only the stop packet will be sent without the start if configured to do so. If only the stop packet is sent, this allows an easy way to reduce the number of records going to the AAA server.

Table 53 RADIUS Accounting Stop Packet Attributes

RADIUS Attributes Value	Attribute	Description
42	acct-input-octets	Number of octets that have been received from the Unity client over the course of the service that is being provided.
43	acct-output-octets	Number of octets that have been sent to the Unity client in the course of delivering this service.

Table 53 RADIUS Accounting Stop Packet Attributes (continued)

RADIUS Attributes Value	Attribute	Description
46	acct-session-time	Length of time (in seconds) that the Unity client has received service.
47	acct-input-packets	Quantity of packets that have been received from the Unity client in the course of delivering this service.
48	acct-output-packets	Quantity of packets that have been sent to the Unity client in the course of delivering this service.
49	acct-terminate-cause	For future use.
52	acct-input-gigawords	How many times the Acct-Input-Octets counter has wrapped around the 2^{32} (2 to the 32nd power) over the course of this service.
52	acct-output-gigawords	How many times the Acct-Input-Octets counter has wrapped around the 2^{32} (2 to the 32nd power) over the course of this service.

RADIUS Update Accounting

RADIUS accounting updates are supported. Packet and octet counts are shown in the updates. To learn more about AAA, refer to the following documents:

- *Configuring Basic AAA RADIUS for Dial-In Clients*
- The chapter "RADIUS Commands" in the *Cisco IOS Security Command Reference*, Release 12.2 T
- *How to Assign Privilege Levels with TACACS+ and RADIUS*
- Other AAA documentation at the Cisco.com website

IKE and IPSec Subsystem Interaction

Accounting Start

If IPSec accounting is configured, after IKE phases are complete, an accounting start record is generated for the session. New accounting records are not generated during a rekeying.

The following is an account start record that was generated on a router and that is to be sent to the AAA server that is defined:

```
*Aug 23 04:06:20.131: RADIUS(00000002): sending
*Aug 23 04:06:20.131: RADIUS(00000002): Send Accounting-Request to 10.1.1.4:1646 id 4, len
220
*Aug 23 04:06:20.131: RADIUS:  authenticator 38 F5 EB 46 4D BE 4A 6F - 45 EB EF 7D B7 19
FB 3F
*Aug 23 04:06:20.135: RADIUS:  Acct-Session-Id     [44]  10  "00000001"
*Aug 23 04:06:20.135: RADIUS:  Vendor, Cisco       [26]  31
*Aug 23 04:06:20.135: RADIUS:   Cisco AVpair       [1]   25  "isakmp-group-id=cclient"
*Aug 23 04:06:20.135: RADIUS:  Framed-IP-Address   [8]   6   10.13.13.1
*Aug 23 04:06:20.135: RADIUS:  Vendor, Cisco       [26]  20
*Aug 23 04:06:20.135: RADIUS:   Cisco AVpair       [1]   14  "vrf-id=cisco"
*Aug 23 04:06:20.135: RADIUS:  Vendor, Cisco       [26]  35
```

```
*Aug 23 04:06:20.135: RADIUS:   Cisco AVpair       [1]   29  "isakmp-initator-ip=11.1.2.2"
*Aug 23 04:06:20.135: RADIUS:  Vendor, Cisco       [26]  36
*Aug 23 04:06:20.135: RADIUS:   Cisco AVpair       [1]   30  "connect-progress=No
Progress"
*Aug 23 04:06:20.135: RADIUS:  User-Name           [1]   13  "joe@cclient"
*Aug 23 04:06:20.135: RADIUS:  Acct-Status-Type    [40]  6   Start                     [1]
*Aug 23 04:06:20.135: RADIUS:  Vendor, Cisco       [26]  25
*Aug 23 04:06:20.135: RADIUS:   cisco-nas-port     [2]   19  "FastEthernet0/0.1"
*Aug 23 04:06:20.135: RADIUS:  NAS-Port            [5]   6   0
*Aug 23 04:06:20.135: RADIUS:  NAS-IP-Address      [4]   6   10.1.1.147
*Aug 23 04:06:20.135: RADIUS:  Acct-Delay-Time     [41]  6   0
*Aug 23 04:06:20.139: RADIUS: Received from id 21645/4 10.1.1.4:1646, Accounting-response,
len 20
*Aug 23 04:06:20.139: RADIUS:  authenticator B7 E3 D0 F5 61 9A 89 D8 - 99 A6 8A 8A 98 79
9D 5D
```

Accounting Stop

An accounting stop packet is generated when there are no more flows (IPSec SA pairs) with the remote peer.

The accounting stop records contain the following information:

- Packets out
- Packets in
- Octets out
- Gigawords in
- Gigawords out

Below is an account start record that was generated on a router. The account start record is to be sent to the AAA server that is defined.

```
*Aug 23 04:20:16.519: RADIUS(00000003): Using existing nas_port 0
*Aug 23 04:20:16.519: RADIUS(00000003): Config NAS IP: 100.1.1.147
*Aug 23 04:20:16.519: RADIUS(00000003): sending
*Aug 23 04:20:16.519: RADIUS(00000003): Send Accounting-Request to 100.1.1.4:1646 id 19,
len 238
*Aug 23 04:20:16.519: RADIUS:  authenticator 82 65 5B 42 F0 3F 17 C3 - 23 F3 4C 35 A2 8A
3E E6
*Aug 23 04:20:16.519: RADIUS:  Acct-Session-Id     [44]  10  "00000002"
*Aug 23 04:20:16.519: RADIUS:  Vendor, Cisco       [26]  20
*Aug 23 04:20:16.519: RADIUS:   Cisco AVpair       [1]   14  "vrf-id=cisco"
*Aug 23 04:20:16.519: RADIUS:  Vendor, Cisco       [26]  35
*Aug 23 04:20:16.519: RADIUS:   Cisco AVpair       [1]   29  "isakmp-initator-ip=11.1.1.2"
*Aug 23 04:20:16.519: RADIUS:  Vendor, Cisco       [26]  36
*Aug 23 04:20:16.519: RADIUS:   Cisco AVpair       [1]   30  "connect-progress=No
Progress"
*Aug 23 04:20:16.519: RADIUS:  Acct-Session-Time   [46]  6   709
*Aug 23 04:20:16.519: RADIUS:  Acct-Input-Octets   [42]  6   152608
*Aug 23 04:20:16.519: RADIUS:  Acct-Output-Octets  [43]  6   152608
*Aug 23 04:20:16.519: RADIUS:  Acct-Input-Packets  [47]  6   1004
*Aug 23 04:20:16.519: RADIUS:  Acct-Output-Packets [48]  6   1004
*Apr 23 04:20:16.519: RADIUS:  Acct-Input-Giga-Word[52]  6   0
*Apr 23 04:20:16.519: RADIUS:  Acct-Output-Giga-Wor[53]  6   0
*Aug 23 04:20:16.519: RADIUS:  Acct-Terminate-Cause[49]  6   none                      [0]
*Aug 23 04:20:16.519: RADIUS:  Vendor, Cisco       [26]  32
*Aug 23 04:20:16.519: RADIUS:   Cisco AVpair       [1]   26  "disc-cause-ext=No Reason"
*Aug 23 04:20:16.519: RADIUS:  Acct-Status-Type    [40]  6   Stop                      [2]
*Aug 23 04:20:16.519: RADIUS:  Vendor, Cisco       [26]  25
*Aug 23 04:20:16.519: RADIUS:   cisco-nas-port     [2]   19  "FastEthernet0/0.1"
```

```
*Aug 23 04:20:16.519: RADIUS:  NAS-Port            [5]   6   0
*Aug 23 04:20:16.519: RADIUS:  NAS-IP-Address      [4]   6   100.1.1.147
*Aug 23 04:20:16.519: RADIUS:  Acct-Delay-Time     [41]  6   0
*Aug 23 04:20:16.523: RADIUS: Received from id 21645/19 100.1.1.4:1646,
Accounting-response, len 20
*Aug 23 04:20:16.523: RADIUS:  authenticator F1 CA C1 28 CE A0 26 C9 - 3E 22 C9 DA EA B8
22 A0
```

Accounting Updates

If accounting updates are enabled, accounting updates are sent while a session is "up." The update interval is configurable. To enable the accounting updates, use the **aaa accounting update** command.

The following is an accounting update record that is being sent from the router:

```
Router#
*Aug 23 21:46:05.263: RADIUS(00000004): Using existing nas_port 0
*Aug 23 21:46:05.263: RADIUS(00000004): Config NAS IP: 100.1.1.147
*Aug 23 21:46:05.263: RADIUS(00000004): sending
*Aug 23 21:46:05.263: RADIUS(00000004): Send Accounting-Request to 100.1.1.4:1646 id 22,
len 200
*Aug 23 21:46:05.263: RADIUS:  authenticator 30 FA 48 86 8E 43 8E 4B - F9 09 71 04 4A F1
52 25
*Aug 23 21:46:05.263: RADIUS:  Acct-Session-Id     [44]  10  "00000003"
*Aug 23 21:46:05.263: RADIUS:  Vendor, Cisco       [26]  20
*Aug 23 21:46:05.263: RADIUS:   Cisco AVpair       [1]   14  "vrf-id=cisco"
*Aug 23 21:46:05.263: RADIUS:  Vendor, Cisco       [26]  35
*Aug 23 21:46:05.263: RADIUS:   Cisco AVpair       [1]   29  "isakmp-initator-ip=11.1.1.2"
*Aug 23 21:46:05.263: RADIUS:  Vendor, Cisco       [26]  36
*Aug 23 21:46:05.263: RADIUS:   Cisco AVpair       [1]   30  "connect-progress=No
Progress"
*Aug 23 21:46:05.263: RADIUS:  Acct-Session-Time   [46]  6   109
*Aug 23 21:46:05.263: RADIUS:  Acct-Input-Octets   [42]  6   608
*Aug 23 21:46:05.263: RADIUS:  Acct-Output-Octets  [43]  6   608
*Aug 23 21:46:05.263: RADIUS:  Acct-Input-Packets  [47]  6   4
*Aug 23 21:46:05.263: RADIUS:  Acct-Output-Packets [48]  6   4
*Aug 23 21:46:05.263: RADIUS:  Acct-Status-Type    [40]  6   Watchdog                  [3]
*Aug 23 21:46:05.263: RADIUS:  Vendor, Cisco       [26]  25
*Aug 23 21:46:05.263: RADIUS:   cisco-nas-port     [2]   19  "FastEthernet0/0.1"
*Aug 23 21:46:05.263: RADIUS:  NAS-Port            [5]   6   0
*Aug 23 21:46:05.263: RADIUS:  NAS-IP-Address      [4]   6   100.1.1.147
*Aug 23 21:46:05.263: RADIUS:  Acct-Delay-Time     [41]  6   0
*Aug 23 21:46:05.267: RADIUS: Received from id 21645/22 100.1.1.4:1646,
Accounting-response, len 20
*Aug 23 21:46:05.267: RADIUS:  authenticator 51 6B BB 27 A4 F5 D7 61 - A7 03 73 D3 0A AC
1C
```

How to Configure IPSec VPN Accounting

This section contains the following procedures:

- Configuring IPSec VPN Accounting, page 1223
- Configuring Accounting Updates, page 1226
- Troubleshooting for IPSec VPN Accounting, page 1227

Configuring IPSec VPN Accounting

To enable IPSec VPN Accounting, you need to perform the following required task:

Prerequisites

Before configuring IPSec VPN accounting, you must first configure IPSec. To learn about configuring IPSec, refer to the following documents:

- The chapter "Configuring IPSec Network Security" in the *Cisco IOS Security Configuration Guide*, Release 12.2
- Other IPSec documentation at the Cisco.com website

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication login** *list-name method*
5. **aaa authorization network** *list-name method*
6. **aaa accounting network** *list-name* **start-stop** [**broadcast**] **group** *group-name*
7. **aaa session-id common**
8. **crypto isakmp profile** *profile-name*
9. **vrf** *ivrf*
10. **match identity group** *group-name*
11. **client authentication list** *list-name*
12. **isakmp authorization list** *list-name*
13. **client configuration address** [**initiate** | **respond**]
14. **accounting** *list-name*
15. **exit**
16. **crypto dynamic-map** *dynamic-map-name dynamic-seq-num*
17. **set transform-set** *transform-set-name*
18. **set isakmp-profile** *profile-name*
19. **reverse-route** [**remote-peer**]
20. **exit**
21. **crypto map** *map-name* **ipsec-isakmp dynamic** *dynamic-template-name*
22. **radius-server host** *ip-address* [**auth-port** *port-number*] [**acct-port** *port-number*]
23. **radius-server key** *string*
24. **interface** *interface-id*
25. **crypto map** *map-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router (config)# aaa new-model`	Enables periodic interim accounting records to be sent to the accounting server.
Step 4	**aaa authentication login** *list-name method* **Example:** `Router (config)# aaa authentication login cisco-client group radius`	Enforces authentication, authorization, and accounting (AAA) authentication for extended authorization (XAUTH) via RADIUS or local.
Step 5	**aaa authorization network** *list-name method* **Example:** `Router (config)# aaa authorization network cisco-client group radius`	Sets AAA authorization parameters on the remote client from RADIUS or local.
Step 6	**aaa accounting network** list-name **start-stop** [**broadcast**] **group** *group-name* **Example:** `Router (config)# aaa accounting network acc start-stop broadcast group radius`	Enables AAA accounting of requested services for billing or security purposes when you use RADIUS or TACACS+.
Step 7	**aaa session-id common** **Example:** `Router (config)# aaa session-id common`	Specifies whether the same session ID will be used for each AAA accounting service type within a call or whether a different session ID will be assigned to each accounting service type.
Step 8	**crypto isakmp profile** *profile-name* **Example:** `Route (config)# crypto isakmp profile cisco`	Audits IP security (IPSec) user sessions and enters isakmp-profile submode.
Step 9	**vrf** *ivrf* **Example:** `Router (conf-isa-prof)# vrf cisco`	Associates the on-demand address pool with a Virtual Private Network (VPN) routing and forwarding (VRF) instance name.

	Command or Action	Purpose
Step 10	**match identity group** *group-name* **Example:** `Router(conf-isa-prof)# match identity group cisco`	Matches an identity from a peer in an ISAKMP profile.
Step 11	**client authentication list** *list-name* **Example:** `Router(conf-isa-prof)# client authentication list cisco`	Configures Internet Key Exchange (IKE) extended authentication (XAUTH) in an Internet Security Association and Key Management Protocol (ISAKMP) profile.
Step 12	**isakmp authorization list** *list-name* **Example:** `Router(conf-isa-prof)# isakmp authorization list cisco-client`	Configures an IKE shared secret and other parameters using the AAA server in an ISAKMP profile. The shared secret and other parameters are generally pushed to the remote peer via mode configuration (MODECFG).
Step 13	**client configuration address** [**initiate** | **respond**] **Example:** `Router(conf-isa-prof)# client configuration address respond`	Configures IKE mode configuration (MODECFG) in the ISAKMP profile.
Step 14	**accounting** *list-name* **Example:** `Router(conf-isa-prof)# accounting acc`	Enables AAA accounting services for all peers that connect via this ISAKMP profile.
Step 15	**exit** **Example:** `Router(conf-isa-prof)# exit`	Exits isakmp-profile submode.
Step 16	**crypto dynamic-map** *dynamic-map-name dynamic-seq-num* **Example:** `Router(config)# crypto dynamic-map mymap 10 ipsec-isakmp`	Creates a dynamic crypto map template and enters the crypto map configuration command mode.
Step 17	**set transform-set** *transform-set-name* **Example:** `Router(config-crypto-map)# set transform-set aswan`	Specifies which transform sets can be used with the crypto map template.
Step 18	**set isakmp-profile** *profile-name* **Example:** `Router(config-crypto-map)# set isakmp-profile cisco`	Sets the ISAKMP profile name.

	Command or Action	Purpose
Step 19	**reverse-route** [**remote-peer**] **Example:** `Router(config-crypto-map)# reverse-route`	Allows routes (ip addresses) to be injected for destinations behind the VPN remote tunnel endpoint and may include a route to the tunnel endpoint itself (using the **remote-peer** keyword for the crypto map.
Step 20	**exit** **Example:** `Router(config-crypto-map)# exit`	Exits dynamic crypto map configuration mode.
Step 21	**crypto map** *map-name* **ipsec-isakmp dynamic** *dynamic-template-name* **Example:** `Router(config)# crypto map mymap ipsec-isakmp dynamic dmap`	Enters crypto map configuration mode
Step 22	**radius-server host** *ip-address* [**auth-port** *port-number*] [**acct-port** *port-number*] **Example:** `Router(config)# radius-server host 172.16.1.4`	Specifies a RADIUS server host.
Step 23	**radius-server key** *string* **Example:** `Router(config)# radius-server key nsite`	Sets the authentication and encryption key for all RADIUS communications between the router and the RADIUS daemon.
Step 24	**interface** *type slot/port* **Example:** `Router(config)# interface FastEthernet 1/0`	Configures an interface type and enters interface configuration mode.
Step 25	**crypto map** *map-name* **Example:** `Router(config-if)# crypto map mymap`	Applies a previously defined crypto map set to an interface.

Configuring Accounting Updates

To send accounting updates while a session is “up,” perform the following optional task:

Prerequisites

Before you configure accounting updates, you must first configure IPSec VPN accounting. See the section “Configuring IPSec VPN Accounting.”

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa accounting update periodic** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa accounting update periodic** *number* **Example:** `Router (config)# aaa accounting update periodic 1-2147483647`	(Optional) Enables periodic interim accounting records to be sent to the accounting server.

Troubleshooting for IPSec VPN Accounting

To display messages about IPSec accounting events, perform the following optional task:

SUMMARY STEPS

1. **enable**
2. **debug crypto isakmp aaa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto isakmp aaa** **Example:** `Router# debug crypto isakmp aaa`	Displays messages about Internet Key Exchange (IKE) events. • The **aaa** keyword specifies accounting events.

Configuration Examples for IPSec VPN Accounting

- Accounting and ISAKMP-Profile Example, page 1228
- Accounting Without ISAKMP Profiles Example, page 1230

Accounting and ISAKMP-Profile Example

The following example shows a configuration for supporting remote access clients with accounting and ISAKMP profiles:

```
version 12.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname sheep
!
aaa new-model
!
!
aaa accounting network ipsecaaa start-stop group radius
aaa accounting update periodic 1
aaa session-id common
ip subnet-zero
ip cef
!
!
no ip domain lookup
ip domain name cisco.com
ip name-server 172.29.2.133
ip name-server 172.29.11.48
!
!
crypto isakmp policy 1
authentication pre-share
group 2
!
crypto isakmp policy 10
hash md5
authentication pre-share
lifetime 200
crypto isakmp key cisco address 172.31.100.2

crypto iakmp client configuration group cclient
 key jegjegjhrg
 pool addressA

crypto-isakmp profile groupA
 vrf cisco
 match identity group cclient
 client authentication list cisco-client
 isakmp authorization list cisco-client
 client configuration address respond
 accounting acc
!
!
crypto ipsec transform-set esp-des-md5 esp-des esp-md5-hmac
!
crypto dynamic-map remotes 1
set peer 172.31.100.2
```

```
set security-association lifetime seconds 120
set transform-set esp-des-md5
reverse-route

!
crypto map test 10 ipsec-isakmp dynamic remotes
!
voice call carrier capacity active
!
interface Loopback0
ip address 10.20.20.20 255.255.255.0
no ip route-cache
no ip mroute-cache
!
interface FastEthernet0/0
ip address 10.2.80.203 255.255.255.0
no ip mroute-cache
load-interval 30
duplex full
!
interface FastEthernet1/0
ip address 192.168.219.2 255.255.255.0
no ip mroute-cache
duplex auto
speed auto
!
interface FastEthernet1/1
ip address 172.28.100.1 255.255.255.0
no ip mroute-cache
duplex auto
speed auto
crypto map test
!
no fair-queue
ip default-gateway 10.2.80.1
ip classless
ip route 10.0.0.0 0.0.0.0 10.2.80.1
ip route 10.20.0.0 255.0.0.0 10.2.80.56
ip route 10.10.10.0 255.255.255.0 172.31.100.2
ip route 10.0.0.2 255.255.255.255 10.2.80.73

ip local pool addressA 192.168.1.1 192.168.1.253
no ip http server
ip pim bidir-enable
!
!
ip access-list extended encrypt
permit ip host 10.0.0.1 host 10.5.0.1
!
access-list 101 permit ip host 10.20.20.20 host 10.10.10.10
!
!
radius-server host 172.27.162.206 auth-port 1645 acct-port 1646 key cisco123
radius-server retransmit 3
radius-server authorization permit missing Service-Type
radius-server vsa send accounting
call rsvp-sync
!
!
mgcp profile default
!
dial-peer cor custom
!
!
```

```
gatekeeper
shutdown
!
!
line con 0
exec-timeout 0 0
exec prompt timestamp
line aux 0
line vty 5 15
 ntp server 172.31.150.52
end
```

Accounting Without ISAKMP Profiles Example

The following example shows a full Cisco IOS configuration that supports accounting remote access peers when ISAKMP profiles are not used:

```
version 12.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname sheep
!
aaa new-model
!
!
aaa accounting network ipsecaaa start-stop group radius
aaa accounting update periodic 1
aaa session-id common
ip subnet-zero
ip cef
!
!
no ip domain lookup
ip domain name cisco.com
ip name-server 172.29.2.133
ip name-server 172.29.11.48
!
!
crypto isakmp policy 1
 authentication pre-share
 group 2
!
crypto isakmp policy 10
 hash md5
 authentication pre-share
 lifetime 200
crypto isakmp key cisco address 172.31.100.2
!
!
crypto ipsec transform-set esp-des-md5 esp-des esp-md5-hmac
!
crypto map test client accounting list ipsecaaa
crypto map test 10 ipsec-isakmp
 set peer 172.31.100.2
 set security-association lifetime seconds 120
 set transform-set esp-des-md5
 match address 101
!
voice call carrier capacity active
!
```

```
interface Loopback0
 ip address 10.20.20.20 255.255.255.0
 no ip route-cache
 no ip mroute-cache
!
interface FastEthernet0/0
 ip address 10.2.80.203 255.255.255.0
 no ip mroute-cache
 load-interval 30
 duplex full
!
interface FastEthernet1/0
 ip address 192.168.219.2 255.255.255.0
 no ip mroute-cache
 duplex auto
 speed auto
!
interface FastEthernet1/1
 ip address 172.28.100.1 255.255.255.0
 no ip mroute-cache
 duplex auto
 speed auto
 crypto map test
!
no fair-queue
ip default-gateway 10.2.80.1
ip classless
ip route 10.0.0.0 0.0.0.0 10.2.80.1
ip route 10.30.0.0 255.0.0.0 10.2.80.56
ip route 10.10.10.0 255.255.255.0 172.31.100.2
ip route 10.0.0.2 255.255.255.255 10.2.80.73
no ip http server
ip pim bidir-enable
!
!
ip access-list extended encrypt
 permit ip host 10.0.0.1 host 10.5.0.1
!
access-list 101 permit ip host 10.20.20.20 host 10.10.10.10
!
!
radius-server host 172.27.162.206 auth-port 1645 acct-port 1646 key cisco123
radius-server retransmit 3
radius-server authorization permit missing Service-Type
radius-server vsa send accounting
call rsvp-sync
!
!
mgcp profile default
!
dial-peer cor custom
!
!
gatekeeper
 shutdown
!
!
line con 0
 exec-timeout 0 0
 exec prompt timestamp
line aux 0
line vty 5 15
!
exception core-file ioscrypto/core/sheep-core
```

```
exception dump 172.25.1.129
ntp clock-period 17208229
ntp server 172.71.150.52
!
end
```

Additional References

For additional information related to IPSec VPN accounting, refer to the following references:

Related Documents

Related Topic	Document Title
Configuring AAA accounting	• The chapter "Configuring Accounting" in the *Cisco IOS Security Configuration Guide*, Release 12.2
Configuring IPSec VPN accounting	• The chapter "Configuring IPSec Network Security" in the *Cisco IOS Security Configuration Guide*, Release 12.2
Configuring basic AAA RADIUS	• *Configuring Basic AAA RADIUS for Dial-In Clients* • *How Does RADIUS Work?* • The chapter "Configuring RADIUS" in the *Cisco IOS Security Configuration Guide*, Release 12.2 • The chapter "RADIUS Commands" in the *Security Command Reference*, Release 12.2 T
Configuring ISAKMP profiles	*VRF-Aware IPSec*, Cisco IOS Release 12.2(15)T feature module
Privilege levels with TACACS+ and RADIUS	*How to Assign Privilege Levels with TACACS+ and RADIUS*
IP security, RADIUS, and AAA commands	*Cisco IOS Security Command Reference,* Release 12.2 T

Standards

Standards	Title
None	

MIBs

MIBs	MIBs Link
None	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

RFCs	Title
None	

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

New Commands

- client authentication list
- client configuration address
- crypto isakmp profile
- isakmp authorization list
- match identity
- set isakmp-profile
- vrf

Modified Commands

- crypto map (global IPSec)
- debug crypto isakmp

Glossary

IKE—Internet Key Exchange. IKE establishes a shared security policy and authenticates keys for services (such as IP security [IPSec]) that require keys. Before any IPSec traffic can be passed, each router, firewall, and host must verify the identity of its peer. This can be done by manually entering preshared keys into both hosts or by a certification authority (CA) service.

IPSec—IP security. IPSec is A framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer. IPSec uses IKE to handle the negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPSec. IPSec can protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

ISAKMP—Internet Security Association and Key Management Protocol. ISAKMP is an Internet IPSec protocol (RFC 2408) that negotiates, establishes, modifies, and deletes security associations. It also exchanges key generation and authentication data (independent of the details of any specific key generation technique), key establishment protocol, encryption algorithm, or authentication mechanism.

L2TP session—Layer 2 Transport Protocol. L2TP are communications transactions between the L2TP access concentrator (LAC) and the L2TP network server (LNS) that support tunneling of a single PPP connection. There is a one-to-one relationship among the PPP connection, L2TP session, and L2TP call.

NAS—network access server. A NAS is a Cisco platform (or collection of platforms, such as an AccessPath system) that interfaces between the packet world (for example, the Internet) and the circuit world (for example, the public switched telephone network [PSTN]).

PFS—perfect forward secrecy. PFS is a cryptographic characteristic associated with a derived shared secret value. With PFS, if one key is compromised, previous and subsequent keys are not compromised because subsequent keys are not derived from previous keys.

QM—Queue Manager. The Cisco IP Queue Manager (IP QM) is an intelligent, IP-based, call-treatment and routing solution that provides powerful call-treatment options as part of the Cisco IP Contact Center (IPCC) solution.

RADIUS—Remote Authentication Dial-In User Service. RADIUS is a database for authenticating modem and ISDN connections and for tracking connection time.

RSA—Rivest, Shamir, and Adelman. Rivest, Shamir, and Adelman are the inventors of the Public-key cryptographic system that can be used for encryption and authentication.

SA—security association. A SA is an instance of security policy and keying material that is applied to a data flow.

TACACS+—Terminal Access Controller Access Control System Plus. TACACS+ is a security application that provides centralized validation of users attempting to gain access to a router or network access server.

TED—Tunnel Endpoint Discovery. TED is a Cisco IOS software feature that allows routers to discover IPSec endpoints.

VPN—Virtual Private Network. A VPN enables IP traffic to travel securely over a public TCP/IP network by encrypting all traffic from one network to another. A VPN uses "tunneling" to encrypt all information at the IP level.

VRF—A VPN routing/forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE router.

VSA—vendor-specific attribute. A VSA is an attribute that has been implemented by a particular vendor. It uses the attribute Vendor-Specific to encapsulate the resulting AV pair: essentially, Vendor-Specific = protocol:attribute = value.

XAUTH—Extended authentication. XAUTH is an optional exchange between IKE Phase 1 and IKE Phase 2, in which the router demands additional authentication information in an attempt to authenticate the actual user (as opposed to authenticating the peer).

Note Refer to the *Internetworking Terms and Acronyms* for terms not included in this glossary.

IPSec VPN High Availability Enhancements

Feature History

Release	Modification
12.1(9)E	This feature was introduced in Cisco IOS Release 12.1(9)E.
12.2(8)T	This feature was integrated into Cisco IOS Release 12.2(8)T.
12.2(11)T	This feature was integrated into Cisco IOS Release 12.2(11)T and support was added for the Cisco AS5300 and Cisco AS5800 platforms.
12.2(9)YE	This feature was integrated into Cisco IOS Release 12.2(9)YE.
12.2(14)S	This feature was integrated into Cisco IOS Release 12.2(14)S.

This feature module describes the IPSec VPN High Availability Enhancements. It includes the following sections:

- Feature Overview, page 1237
- Supported Platforms, page 1240
- Supported Standards, MIBs, and RFCs, page 1241
- Configuration Tasks, page 1242
- Configuration Examples, page 1244
- Command Reference, page 1246

Feature Overview

The IPSec VPN High Availability Enhancements feature consists of two new features—Reverse Route Injection (RRI) and Hot Standby Router Protocol and IPSec (HSRP)—that work together to provide users with a simplified network design for VPNs, and reduced configuration complexity on remote peers with respect to defining gateway lists. When used together, RRI and HSRP provide a more reliable network design for VPNs and reduce configuration complexity on remote peers.

Reverse Route Injection

Reverse Route Injection (RRI) is a feature designed to simplify network design for Virtual Private Networks (VPNs) in which there is a requirement for redundancy or load balancing. RRI works with both dynamic and static crypto maps.

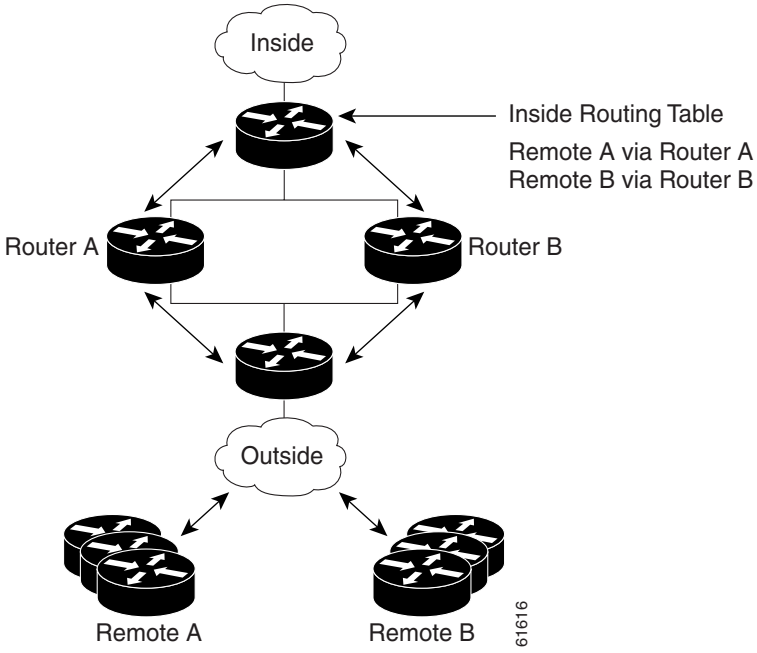
In the dynamic case, as remote peers establish IPSec security associations (SAs) with an RRI-enabled router, a static route is created for each subnet or host protected by that remote peer. For static crypto maps, a static route is created for each destination of an extended access list rule. When RRI is used on a static crypto map with an access control list (ACL), routes will always exist, even without the negotiation of IPsec SAs.

Note Use of any keyword in ACLs with RRI is not supported.

When routes are created, they are injected into any dynamic routing protocol and distributed to surrounding devices. This traffic flows, requiring IPSec to be directed to the appropriate RRI router for transport across the correct SAs to avoid IPSec policy mismatches and possible packet loss.

Figure 87 shows a RRI configuration functionality topology. Remote A is being serviced by Router A and Remote B connected to Router B, providing load balancing across VPN gateways at the central site. RRI on the central site devices will ensure that the other router on the inside of the network can automatically make the correct forwarding decision. RRI also eliminates the need to administer static routes on the inside router.

Figure 87 Topology Showing Reverse Route Injection Configuration Functionality



Hot Standby Router Protocol and IPSec

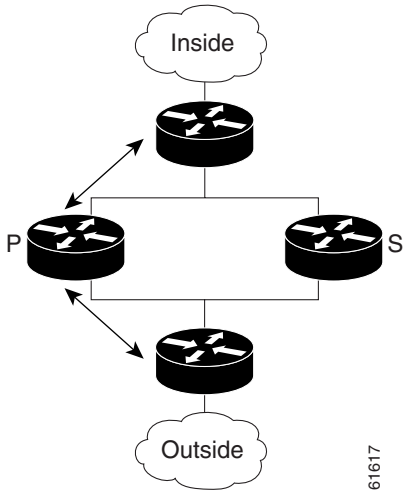
Hot Standby Router Protocol (HSRP) is designed to provide high network availability by routing IP traffic from hosts on Ethernet networks without relying on the availability of any single router. HSRP is particularly useful for hosts that do not support a router discovery protocol, such as ICMP Router Discovery Protocol (IRDP), and do not have the functionality to switch to a new router when their selected router reloads or loses power. Without this functionality, a router that loses its default gateway because of a router failure is unable to communicate with the network.

HSRP is configurable on LAN interfaces using standby command-line interface (CLI) commands. It is now possible to use the standby IP address from an interface as the local IPSec identity, or local tunnel endpoint.

By using the standby IP address as the tunnel endpoint, failover can be applied to VPN routers by using HSRP. Remote VPN gateways connect to the local VPN router via the standby address that belongs to the *active* device in the HSRP group. In the event of failover, the *standby* device takes over ownership of the standby IP address and begins to service remote VPN gateways.

Figure 88 shows the enhanced HSRP functionality topology. Traffic is serviced by the active Router P, the active device in the standby group. In the event of failover, traffic is diverted to Router S, the original standby device. Router S assumes the role of the new active router and takes ownership of the standby IP address.

Figure 88 Topology Showing Hot Standby Router Protocol Functionality



Note In case of a failover, HSRP does not facilitate IPSec state information transference between VPN routers. This means that without this state transference, SAs to remotes will be deleted requiring Internet Key Exchange (IKE) and IPSec SAs to be reestablished. To make IPSec failover more efficient, it is recommended that IKE keepalives be enabled on all routers.

Benefits

Reverse Route Injection

- Enables routing of IPSec traffic to a specific VPN headend device in environments that have multiple (redundant) VPN headend devices.
- Ensures predictable failover time of remote sessions between headend devices when using IKE keepalives, especially in environments in which remote device route flapping is common (not taking into consideration the effects of route convergence, which may vary depending on the routing protocol used and the size of the network).
- Eliminates the need for the administration of static routes on upstream devices as routes are dynamically learned by these devices.

Hot Standby Router Protocol with IPSec

Failover can be applied to VPN routers through the use of HSRP. Remote VPN gateways connect to the local VPN router through the standby address that belongs to the active device in the HSRP group. This functionality reduces configuration complexity on remote peers with respect to defining gateway lists because only the HSRP standby address needs to be defined.

Related Documents

- *IPSec Stateful Failover (VPN High Availability)*
- *Cisco IOS Security Configuration Guide,* Release 12.2
- *Cisco IOS IP Configuration Guide*, Release 12.2 (Configuring IP Services chapter)
- *VPN Acceleration Module Installation and Configuration Guide*
- *SA-VAM2 Installation and Configuration Guide*
- *Release Notes for the SA-VAM2*
- *Cisco 7100 Series VPN Router Installation and Configuration Guide*
- *Cisco 7200 VXR Installation and Configuration Guide*
- *Cisco 7401ASR Installation and Configuration Guide*

Supported Platforms

Cisco IOS Release 12.1(9)E and Cisco IOS Release 12.2(8)T

- Cisco 7100 series
- Cisco 7200VXR series

Cisco IOS Release 12.2(8)T Only

- Cisco 1710
- Cisco 1720
- Cisco 1750
- Cisco 1751
- Cisco 2600 series
- Cisco 3620
- Cisco 3631
- Cisco 3640
- Cisco 3660
- Cisco 3725
- Cisco 3745
- Cisco uBR7200
- Cisco uBR925

Cisco IOS Release 12.2(11)T Only

- Cisco AS5300 series
- Cisco AS5800 series

Cisco IOS Release 12.2(9)YE

- Cisco 7401ASR router

Cisco IOS Release 12.2(14)S

- Cisco 7200 series
- Cisco 7400 series

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Supported Standards, MIBs, and RFCs

Standards

- No new or modified standards are supported by this feature.

MIBs

- No new or modified standards are supported by this feature.

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

- No new or modified RFCs are supported by this feature.

Configuration Tasks

See the following sections for configuration tasks for the IPSec VPN High Availability Enhancements feature. Each task in the list is identified as either required or optional.

- Configuring Reverse Route Injection on a Dynamic Crypto Map (required)
- Configuring Reverse Route Injection on a Static Crypto Map (required)
- Configuring HSRP with IPSec (required)
- Verifying VPN IPSec Crypto Configuration (optional)

Configuring Reverse Route Injection on a Dynamic Crypto Map

Dynamic crypto map entries, like regular static crypto map entries, are grouped into sets. A set is a group of dynamic crypto map entries all with the same dynamic map name but each with a different dynamic sequence number. Each member of the set may be configured for RRI.

To create a dynamic crypto map entry and enable RRI, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router (config)# crypto dynamic map-name seq-num`	Creates a dynamic crypto map entry and enters crypto map configuration mode.
Step 2	`Router (config-crypto-m)# set transform-set`	Specifies which transform sets are allowed for the crypto map entry. Lists multiple transform sets in order of priority (highest priority first). This entry is the only configuration statement required in dynamic crypto map entries.
Step 3	`Router (config-crypto-m)# reverse-route`	Creates source proxy information.

Configuring Reverse Route Injection on a Static Crypto Map

Before configuring RRI on a static crypto map, please note the following items:

- Routes are not created based on access list 102 as reverse-route is not enabled on mymap 2. RRI is not enabled by default and is not displayed in the router configuration.
- Enable a routing protocol to distribute the VPN routes to upstream devices.
- If Cisco Express Forwarding (CEF) is run on a VPN router configured for RRI, adjacencies need to be formed for each RRI injected network through the next hop device. As the next hop is not explicitly defined in the routing table for these routes, proxy-ARP should be enabled on the next hop router which allows the CEF adjacency to be formed using the layer two addresses of that device. In cases where there are many RRI injected routes, adjacency tables may become quite large as an entry is created for each device from each of the subnets represented by the RRI route. This issue is to be resolved in a future release.

To add RRI to a static crypto map set, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router (config)# crypto map map-name seq-num ipsec-isakmp`	Adds a dynamic crypto map set to a static crypto map set and enters interface configuration mode.
Step 2	`Router (config-if)# set peer ip address`	Specifies an IPSec peer IP address in a crypto map entry.
Step 3	`Router (config-if)# reverse-route`	Creates dynamically static routes based on crypto access control lists (ACLs).
Step 4	`Router (config-if)# match address`	Specifies an extended access list for a crypto map entry.
Step 5	`Router (config-if)# set transform-set`	Specifies which transform sets are allowed for the crypto map entry. Lists multiple transform sets in order of priority (highest priority first).

Configuring HSRP with IPSec

When configuring HSRP with IPSec, the following conditions may apply:

- When HSRP is applied to a crypto map on an interface, the crypto map must be reapplied if the standby IP address or the standby name is changed on that interface.
- If HSRP is applied to a crypto map on an interface, and the user deletes the standby IP address or the standby name from that interface, the crypto tunnel endpoint is reinitialized to the actual IP address of that interface.
- If a user adds the standby IP address and the standby name to an interface with the requirement IPSec failover, the crypto map must be reapplied with the appropriate redundancy information.
- Standby priorities should be equal on active and standby routers. If they are not, the higher priority router takes over as the active router. When that occurs, the active router goes into a cycle where it continously goes down and comes back up.
- The IP addresses on the HSRP-tracked interfaces on the standby and active routers should both be either lower or higher on one router than the other. In the case of equal priorities (an HA requirement), HSRP will assign the active state-based IP address. If an addressing scheme exists so that the public IP address of router A is lower than the public IP address of router B, but the opposite is true for their private interfaces, an active/standby-stanby/active split conditon could exist which will break connectivity.

Note To configure HSRP without IPSec refer to the "Configuring IP Services" chapter of the *Cisco IOS IP Configuration Guide*, Release 12.2.

To apply a crypto map set to an interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router (config)# interface type slot/port`	Specifies an interface and enters interface configuration mode.
Step 2	`Router (config-if)# standby name group-name`	Specifies the standby group name (required).
Step 3	`Router (config-if)# standby ip ip-address`	Specifies the IP address of the standby groups (required for one device in the group).
Step 4	`Router (config-if)# crypto map map-name redundancy [standby-name]`	Specifies IP redundancy address as the tunnel endpoint for IPSec.

Verifying VPN IPSec Crypto Configuration

To verify your VPN IPSec crypto configuration, use the following EXEC commands:

Command	Purpose
`Router# show crypto ipsec transform-set`	Displays your transform set configuration.
`Router# show crypto map [interface interface | tag map-name]`	Displays your crypto map configuration.
`Router# show crypto ipsec sa [map map-name | address | identity] [detail]`	Displays information about IPSec SAs.
`Router# show crypto dynamic-map [tag map-name]`	Displays information about dynamic crypto maps.

Configuration Examples

This section provides the following configuration examples:

- Reverse Route Injection on a Dynamic Crypto Map Example
- Reverse Route Injection on a Static Crypto Map Example
- HSRP and IPSec Example

Reverse Route Injection on a Dynamic Crypto Map Example

In the following example, using the reverse route crypto map subcommand in the definition of the dynamic crypto map template ensures that routes are created for any remote proxies (subnets or hosts), protected by the connecting remote IPSec peers.

```
crypto dynamic mydynmap 1
    set transform-set esp-3des-sha
    reverse-route
```

This template is then associated with a "parent" crypto map statement and then applied to an interface.

```
crypto map mymap 3 ipsec-isakmp dynamic mydynmap

interface FastEthernet 0/0
crypto map mymap
```

Reverse Route Injection on a Static Crypto Map Example

RRI is a good solution for topologies that require encrypted traffic to be diverted to a VPN router and all other traffic to a different router.

In these scenarios, RRI eliminates the need to manually define static routes on devices.

RRI is not required if a single VPN router is used and all traffic passes through the VPN router during its path in and out of the network.

If the user chooses to manually define static routes on the VPN router for remote proxies, and have these routes permanently installed in the routing table, RRI should not be enabled on the crypto map instance that covers the same remote proxies. In this case, there is no possibility of user defined static routes being removed by RRI.

Routing convergence can affect the success of a failover based on the routing protocol used to advertise routes (link state versus periodic update). It is recommended that a link state routing protocol such as OSPF be used to help speed convergence time by ensuring that routing updates are sent as soon as a change in routing state is detected.

In the following example, RRI is enabled for mymap 1, but not for mymap 2. Upon the application of the crypto map to the interface, a route is created based on access-list 101 analogous to the following:

```
IP route 172.17.11.0 255.255.255.0 FastEthernet 0/0

crypto map mymap 1 ipsec-isakmp
   set peer 172.17.11.1
   reverse-route
   set transform-set esp-3des-sha
   match address 101
crypto map mymap 2 ipsec-isakmp
   set peer 10.1.1.1
   set transform-set esp-3des-sha
   match address 102

access-list 101 permit ip 192.168.1.0 0.0.0.255 172.17.11.0 0.0.0.255
access-list 102 permit ip 192.168.1.0 0.0.0.255 10.0.0.0 0.0.255.255

interface FastEthernet 0/0
    crypto map mymap
```

HSRP and IPSec Example

The following example shows how all remote VPN gateways connect to the router via 192.168.0.3. The crypto map on the interface binds this standby address as the local tunnel endpoint for all instances of *mymap* and at the same time ensures that HSRP failover is facilitated between an active and standby device belonging to the same standby group, group1.

Note that RRI is also enabled to provide the ability for only the *active* device in the HSRP group to be advertising itself to inside devices as the next hop VPN gateway to the remote proxies. If there is a failover, routes are deleted on the formerly active device and created on the newly active device.

```
crypto map mymap 1 ipsec-isakmp
    set peer 10.1.1.1
    reverse-route
    set transform-set esp-3des-sha
    match address 102

Interface FastEthernet 0/0
   ip address 192.168.0.2 255.255.255.0
   standby name group1
   standby ip 192.168.0.3
   crypto map mymap redundancy group1

access-list 102 permit ip 192.168.1.0 0.0.0.255 10.0.0.0 0.0.255.255
```

The standby name needs to be configured on all devices in the standby group and the standby address needs to configured on at least one member of the group. If the standby name is removed from the router, the IPSec SAs will be deleted. If the standby name is added again, regardless of whether the same name or a different name is used, the crypto map (using the redundancy option) will have to be reapplied to the interface.

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **crypto map (interface IPSec)**
- **reverse-route**

L2TP Security

Feature History for L2TP Security

Release	Modification
12.2(4)T	This feature was introduced.
12.2(4)T3	Support for the Cisco 7500 series routers was added.
12.2(11)T	This feature was integrated into Cisco IOS Release 12.2(11)T and support was added for the Cisco 1760, Cisco AS5300, Cisco AS5400, and Cisco AS5800 platforms.
12.2(27)SB	This feature was integrated into Cisco IOS Release 12.2(27)SB.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

This document describes the L2TP Security feature in Cisco IOS Release 12.2(11)T. It includes the following sections:

- Feature Overview, page 1247
- Supported Platforms, page 1248
- Supported Standards, MIBs, and RFCs, page 1249
- Prerequisites, page 1250
- Configuration Tasks, page 1250
- Configuration Examples, page 1259
- Command Reference, page 1262

Feature Overview

The L2TP Security feature provides enhanced security for tunneled PPP frames between the Layer 2 Transport Protocol (L2TP) access concentrator (LAC) and the L2TP network server (LNS). Previous releases of the Cisco IOS software provided only a one-time, optional mutual authentication during tunnel setup with no authentication of subsequent data packets or control messages. In situations where the L2TP is used to tunnel PPP sessions over an untrusted infrastructure such as the Internet, the security

attributes of L2TP and PPP are inadequate. PPP provides no protection of the L2TP tunnel, and current PPP encryption protocols provide inadequate key management and no authentication or integrity mechanisms. The L2TP Security feature allows the robust security features of IP Security (IPSec) to protect the L2TP tunnel and the PPP sessions within the tunnel. In addition, the L2TP Security feature provides built-in keepalives and standardized interfaces for user authentication and accounting to authentication, authorization, and accounting (AAA) servers.

The deployment of Microsoft Windows 2000 demands the integration of IPSec with L2TP because this is the default virtual private dialup network (VPDN) networking scenario. This integration of protocols is also used for LAN-to-LAN VPDN connections in Microsoft Windows 2000. The L2TP Security feature provides integration of IPSec with L2TP in a solution that is scalable to large networks with minimal configuration.

Benefits

The enhanced security provided by the L2TP Security feature increases the integrity and confidentiality of tunneled PPP sessions within a standardized, well deployed Layer 2 tunneling solution. The robust security features of IPSec and Internet Key Exchange (IKE) include confidentiality, integrity checking, replay protection, authentication and key management. Traditional routing protocols such as Routing Information Protocol (RIP), Open Shortest Path First (OSPF), and Interior Gateway Routing Protocol (IGRP) will run transparently because a real PPP interface is associated with the secure tunnel. Additional benefits include built in keepalives and standardized interfaces for user authentication and accounting to AAA servers, interface statistics, standardized MIBs, and multiprotocol support.

Related Features and Technologies

- L2TP Large-Scale Dial-Out
- Timer and Retry Enhancements for L2TP and L2F
- VPDN Group Session Limiting

Related Documents

- *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2
- *Cisco IOS Dial Technologies Command Reference*, Release 12.2
- *Cisco IOS Security Configuration Guide*, Release 12.2
- *Cisco IOS Security Command Reference*, Release 12.2

Supported Platforms

- Cisco 806
- Cisco 1600 series
- Cisco 1700 series
- Cisco 2691
- Cisco 3620

- Cisco 3640
- Cisco 3660
- Cisco 3700 series
- Cisco 7100 series
- Cisco 7200 series
- Cisco 7500 series
- Cisco AS5300
- Cisco AS5400
- Cisco AS5800
- Cisco IGX 8400 URM

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that are supported on specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

No new or modified MIBs are supported by this feature.

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

- RFC 1661, *The Point-to-Point Protocol (PPP)*
- RFC 2401, *Security Architecture for the Internet Protocol (IPSec)*
- RFC 2409, *The Internet Key Exchange (IKE)*
- RFC 2637, *Point to Point Tunneling Protocol (PPTP)*
- RFC 2661, *Layer Two Transport Protocol (L2TP)*
- RFC 3193, *Securing L2TP Using IPSec*

Prerequisites

The interface between the LAC and LNS must be configured for IP and must support IPSec.

To use the L2TP Security feature for client-initiated dial-in using compulsory tunneling, the interface between the client and the LAC must support PPP.

To use the L2TP Security feature for client-initiated dail-in using voluntary tunneling, the client software must support L2TP and IPSec. This is the default VPDN networking scenario in Microsoft Windows 2000.

Configuration Tasks

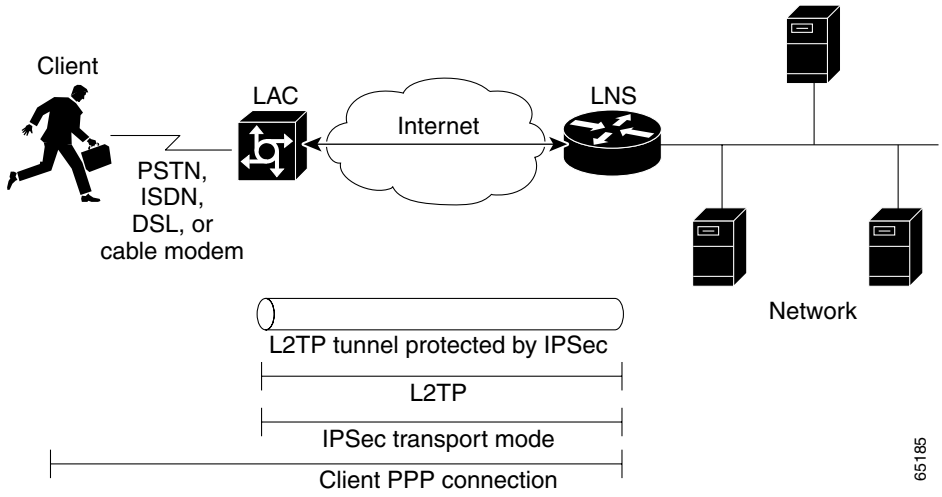
See the following sections for configuration tasks for the L2TP Security feature. Each task in the list is identified as either required or optional:

- Configuring NAS-Initiated VPDN Tunneling with L2TP Security (optional)
- Configuring Client-Initiated VPDN Tunneling with L2TP Security (optional)

Configuring NAS-Initiated VPDN Tunneling with L2TP Security

In the NAS-initiated (compulsory) tunneling scenario depicted in Figure 89, the client connects to the LAC through a media that supports PPP, such as a dialup modem, DSL, ISDN, or a cable modem. If the connection from the client to the LAC is considered secure such as a modem, ISDN or a DSL connection the client may choose not to provide additional security. The PPP session is securely tunneled from the LAC to the LNS without any required knowledge or interaction by the client.

Figure 89 NAS-Initiated Tunneling



To configure the L2TP Security feature for compulsory tunneling, perform the tasks described in the following sections to configure the client, LAC, and LNS:

- Configuring the Client (required)
- Configuring the LAC (required)
- Configuring the LNS (required)

Configuring the Client

To use the L2TP Security feature for NAS-initiated dial-in using compulsory tunneling, configure the interface between the client and the LAC for PPP. For more information on configuring PPP on the client, refer to the *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2.

Configuring the LAC

To configure the LAC to use the L2TP Security feature, perform the required tasks described in the following sections:

- Configuring the Interface Between the LAC and the LNS (required)
- Configuring IPSec Protection of a L2TP Tunnel at the LAC (required)
- Creating the Security Profile at the LAC (required)

Configuring the Interface Between the LAC and the LNS

To configure the interface between the LAC and the LNS, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# username username password secret`	Establishes a username-based authentication system for L2TP tunnel authentication at the LAC.
Step 2	`Router(config)# username username password secret`	Establishes a username-based authentication system for L2TP tunnel authentication at the LNS.
Step 3	`Router(config)# vpdn enable`	Enables VPDN on the router.
Step 4	`Router(config)# no vpdn logging`	Disables the logging of VPDN events.

Configuring IPSec Protection of a L2TP Tunnel at the LAC

To configure a VPDN group to tunnel PPP sessions with IPSec protection, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# vpdn-group name`	Enters VPDN group configuration mode and associates a VPDN group to a VPDN profile.
Step 2	`Router(config-vpdn)# request-dialin`	Enters VPDN request-dialin configuration submode, configures the LAC to request L2TP tunnels to the LNS, and specifies a VPDN subgroup.
Step 3	`Router(config-vpdn-req-in)# protocol l2tp`	Specifies L2TP as the tunneling protocol that the VPDN subgroup will use.
Step 4	`Router(config-vpdn-req-in)# domain name`	Requests that PPP calls from a specific domain name be tunneled. The request-dialin VPDN subgroup can be configured to tunnel calls from multiple Domain Name System numbers and domain names.
Step 5	`Router(config-vpdn-req-in)# exit`	Exits VPDN request-dialin configuration submode.
Step 6	`Router(config-vpdn)# initiate-to ip ip-address`	Specifies the IP address to which the LAC will tunnel.
Step 7	`Router(config-vpdn)# local name name`	Specifies a local host name that the tunnel will use to identify itself.

	Command	Purpose
Step 8	`Router(config-vpdn)# l2tp security crypto-profile profile-name [keep-sa]`	Enables the VPDN group to be protected by IPSec. • *profile-name*—The name of the crypto profile to be used for IPSec protection of tunneled PPP sessions. The *profile-name* must match that of a profile configured using the **crypto-map** command. • **keep-sa**—This keyword is used to control the destruction of IPSec security associations (SAs) upon tunnel teardown. By default, any IPSec phase 2 SAs and IKE phase 1 SAs are destroyed when the L2TP tunnel is torn down. Using the **keep-sa** keyword prevents the destruction of IKE phase 1 SAs.
Step 9	`Router(config-vpdn)# l2tp tunnel password password`	Sets the password that the router will use to authenticate the tunnel.

Creating the Security Profile at the LAC

To create an IKE policy and a crypto profile configuration associated with the VPDN group, you must first configure phase 1 ISAKMP policy and an IPSec transform set. For more information on configuring phase 1 ISAKMP policies and IPSec transform, sets refer to the *Cisco IOS Security Configuration Guide*, Release 12.2. Once the phase 1 ISAKMP policy and an IPSec transform set have been configured, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto map map-name seq-num ipsec-isakmp profile profile-name`	Enters crypto map configuration mode and creates a crypto profile to be used as a configuration template for dynamically created crypto maps. **Note** The **set peer** and **match address** commands are ignored by crypto profiles and should not be configured in the crypto map definition.
Step 2	`Router(config-crypto-map)# set transform-set transform-set-name [transform-set-name2...transform-set-name6]`	Specifies which transform sets can be used with the crypto map entry.
Step 3	`Router(config-crypto-map)# exit`	Returns to global configuration mode.
Step 4	`Router(config)# interface fastethernet slot/port`	Enters interface configuration mode and selects a particular Fast Ethernet interface for configuration.
Step 5	`Router(config-if)# ip address ip-address mask`	Sets a primary IP address for the interface.
Step 6	`Router(config-if)# crypto map map-name`	Associates the crypto map with the interface.

Configuring the LNS

To configure the LNS to use the L2TP Security feature, perform the required tasks described in the following sections:

- Configuring the Interface Between the LNS and the LAC (required)
- Configuring IPSec Protection of an L2TP Tunnel at the LNS (required)
- Creating the Security Profile at the LNS (required)

Configuring the Interface Between the LNS and the LAC

To configure the interface between the LNS and the LAC, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# username username password secret`	Establishes a username-based authentication system for Challenge Handshake Authentication Protocol (CHAP) authentication of the client.
Step 2	`Router(config)# username username password secret`	Establishes a username-based authentication system for L2TP tunnel authentication at the LAC.
Step 3	`Router(config)# username username password secret`	Establishes a username-based authentication system for L2TP tunnel authentication at the LNS.
Step 4	`Router(config)# ip address pool local`	Enables address pooling to supply IP addresses to the client.
Step 5	`Router(config)# vpdn enable`	Enables VPDN on the router.

Configuring IPSec Protection of an L2TP Tunnel at the LNS

To configure a VPDN group to tunnel PPP sessions with IPSec protection, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# vpdn-group name`	Enters VPDN group configuration mode and associates a VPDN group to a VPDN profile.
Step 2	`Router(config-vpdn)# accept dialin`	Enters VPDN accept-dialin configuration mode, configures the LNS to accept tunneled PPP connections from a LAC, and specifies a VPDN subgroup.
Step 3	`Router(config-vpdn-acc-in)# protocol l2tp`	Specifies L2TP as the tunneling protocol the VPDN subgroup will use.
Step 4	`Router(config-vpdn-acc-in)# virtual-template template-number`	Specifies which virtual template will be used to clone virtual access interfaces.
Step 5	`Router(config-vpdn-acc-in)# exit`	Returns to VPDN group configuration mode.
Step 6	`Router(config-vpdn)# terminate-from hostname host-name`	Specifies the host name of the remote LAC that is required to accept a VPDN tunnel.

	Command	Purpose
Step 7	`Router(config-vpdn)# lcp renegotiation {always | on-mismatch}`	Allows the LNS to renegotiate the Link Control Protocol (LCP).
Step 8	`Router(config-vpdn-acc-in)# l2tp security crypto-profile profile-name [keep-sa]`	Enables the VPDN group to be protected by IPSec. • *profile-name*—The name of the crypto profile to be used for IPSec protection of tunneled PPP sessions. The *profile-name* must match that of a profile configured using the **crypto-map** command. • **keep-sa**—This keyword is used to control the destruction of IPSec security associations (SAs) upon tunnel teardown. By default, any IPSec phase 2 SAs and IKE phase 1 SAs are destroyed when the L2TP tunnel is torn down. Using the **keep-sa** keyword prevents the destruction of IKE phase 1 SAs.

Creating the Security Profile at the LNS

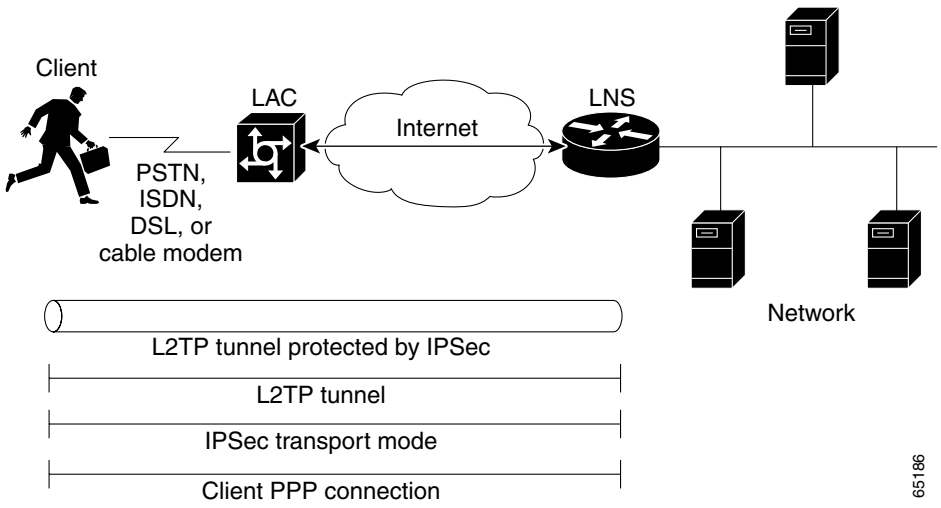
To create an IKE policy and a crypto profile configuration associated with the VPDN group, you must first configure phase 1 ISAKMP policy and an IPSec transform set. For more information on configuring phase 1 ISAKMP policies and IPSec transform, sets refer to the *Cisco IOS Security Configuration Guide*, Release 12.2. Once the phase 1 ISAKMP policy and an IPSec transform set have been configured, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto map map-name seq-num ipsec-isakmp profile profile-name`	Enters crypto map configuration mode and creates a crypto profile to be used as a configuration template for dynamically created crypto maps.
Step 2	`Router(config-crypto-map)# set transform-set transform-set-name [transform-set-name2...transform-set-name6]`	Specifies which transform sets can be used with the crypto map entry.
Step 3	`Router(config-crypto-map)# exit`	Returns to global configuration mode.
Step 4	`Router(config)# interface fastethernet slot/port`	Enters interface configuration mode and selects a particular Fast Ethernet interface for configuration.
Step 5	`Router(config-if)# ip address ip-address mask`	Sets a primary IP address for the interface.
Step 6	`Router(config-if)# speed {10 | 100 | auto}`	Configures the speed for a Fast Ethernet interface.
Step 7	`Router(config-if)# half-duplex`	Specifies half-duplex mode.
Step 8	`Router(config-if)# crypto map map-name`	Associates the crypto map with the interface.

Configuring Client-Initiated VPDN Tunneling with L2TP Security

In the client-initiated (voluntary) tunneling scenario depicted in Figure 90, the client initiates an L2TP tunnel to the LNS without the intermediate NAS participating in tunnel negotiation or establishment. The client must manage the software that initiates the tunnel. Microsoft Windows 2000 supports this VPDN scenario. In this scenario, extended services processor (ESP) with authentication must always be used.

Figure 90 Client-Initiated Tunneling



To configure the L2TP Security feature for voluntary tunneling, you must configure the LNS to interface with the LAC by performing the tasks described in the "Configuring the Interface Between the LNS and the LAC"section.

In addition, perform the tasks in the following sections, which are unique to configuring the NAS for client-initiated dial-in using voluntary tunneling:

- Configuring IPSec Protection of a VPDN Session at the LNS (required)
- Creating the Security Profile at the LNS (required)

Configuring IPSec Protection of a VPDN Session at the LNS

To configure a VPDN group to tunnel PPP sessions with IPSec protection, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# vpdn-group name`	Enters VPDN group configuration mode and associates a VPDN group to a VPDN profile.
Step 2	`Router(config-vpdn)# accept dialin`	Enters VPDN accept-dialin configurative mode, configures the LNS to accept tunneled PPP connections from a LAC, and specifies a VPDN subgroup.
Step 3	`Router(config-vpdn-acc-in)# protocol l2tp`	Specifies L2TP as the tunneling protocol the VPDN subgroup will use.

	Command	Purpose
Step 4	`Router(config-vpdn-acc-in)# virtual-template template-number`	Specifies which virtual template will be used to clone virtual access interfaces.
Step 5	`Router(config-vpdn-acc-in)# l2tp security crypto-profile profile-name [keep-sa]`	Enables the VPDN group to be protected by IPSec. • *profile-name*—The name of the crypto profile to be used for IPSec protection of tunneled PPP sessions. The *profile-name* must match that of a profile configured using the **crypto-map** command. • **keep-sa**—This keyword
Step 6	`Router(config-vpdn)# no l2tp tunnel authentication`	Disables L2TP tunnel authentication.
Step 7	`Router(config-vpdn)# lcp renegotiation {always | on-mismatch}`	Allows the LNS to renegotiate the LCP.
Step 8	`Router(config-vpdn)# ip pmtu`	Allows L2TP tunnels to participate in path maximum transmission unit (MTU) discovery.

Creating the Security Profile at the LNS

To create an IKE policy and a crypto profile configuration associated with the VPDN group, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto ipsec transform-set transform-set-name transform1 [transform2] [transform3]`	Enters crypto transform configuration mode and defines a transform set. There are complex rules defining which entries you can use for the *transform* arguments. For more information, refer to the **crypto ipsec transform-set** command description or the table of allowed transform combinations.
Step 2	`Router(config-crypto-trans)# crypto map map-name seq-num ipsec-isakmp profile profile-name`	Enters crypto map configuration mode and creates a crypto profile to be used as a configuration template for dynamically created crypto maps.
Step 3	`Router(config-crypto-map)# set transform-set transform-set-name [transform-set-name2...transform-set-name6]`	Specifies which transform sets can be used with the crypto map entry.
Step 4	`Router(config-crypto-map)# set security-association lifetime {seconds seconds | kilobytes kilobytes}`	Overrides the global lifetime value for a particular crypto map entry. The global lifetime value is used when negotiating IPSec security associations.
Step 5	`Router(config-crypto-map)# exit`	Returns to global configuration mode.
Step 6	`Router(config)# interface fastethernet slot/port`	Enters interface configuration mode and selects a particular Fast Ethernet interface for configuration.
Step 7	`Router(config-if)# ip address ip-address mask`	Sets a primary IP address for the interface.
Step 8	`Router(config-if)# speed {10 | 100 | auto}`	Configures the speed for a Fast Ethernet interface.

	Command	Purpose
Step 9	`Router(config-if)# half-duplex`	Specifies half-duplex mode.
Step 10	`Router(config-if)# crypto map map-name`	Associates the crypto map with the interface.

Verifying Session Establishment

To verify the establishment and security of an L2TP tunnel, perform the following steps:

Step 1 Enable the **debug crypto socket** and **debug vpdn l2x-events** commands. The **crypto socket messages** command allows you to view socket messages and verify that the socket is created and moved to the active state. The **vpdn l2x-events** command tracks incoming and outgoing L2TP packets.

```
router# debug crypto socket
router# debug vpdn l2x-events
*Mar  1 00:56:46.959:CRYPTO_SS(L2X Security):Passive open, socket info:local
10.0.0.13/1701, remote 10.0.0.12/1701, prot 17, ifc Fa0/0
*Mar  1 00:56:47.291:L2TP:I SCCRQ from ebooth02 tnl 5107
*Mar  1 00:56:47.295:L2X:Requested security for socket, UDP socket info:local
10.0.0.13(1701), remote 10.0.0.12(1701)
*Mar  1 00:56:47.295:Tnl 13582 L2TP:Got a challenge in SCCRQ, ebooth02
*Mar  1 00:56:47.295:Tnl 13582 L2TP:New tunnel created for remote ebooth02, address
10.0.0.12
*Mar  1 00:56:47.295:Tnl 13582 L2TP:O SCCRP  to ebooth02 tnlid 5107
*Mar  1 00:56:47.295:Tnl 13582 L2TP:Control channel retransmit delay set to 1 seconds
*Mar  1 00:56:47.299:Tnl 13582 L2TP:Tunnel state change from idle to wait-ctl-reply
*Mar  1 00:56:47.299:CRYPTO_SS(L2X Security):Completed binding of application to socket
```

Step 2 Use the **show crypto map tag** *crypto-map-name* command to verify that a crypto map was dynamically created for the L2TP tunnel.

```
router# show crypto map tag l2tpsec
Crypto Map "l2tpsec" 10 ipsec-isakmp
        No matching address list set.
        Current peer:0.0.0.0
        Security association lifetime:4608000 kilobytes/3600 seconds
        PFS (Y/N):N
        Transform sets={ esp, }

Crypto Map "l2tpsec" 20 ipsec-isakmp
        Peer = 10.0.0.13
        Extended IP access list
            access-list  permit udp host 10.0.0.12 port = 1701 host 10.0.0.13 port = 1701
        Current peer:10.0.0.13
        Security association lifetime:4608000 kilobytes/3600 seconds
        PFS (Y/N):N
        Transform sets={ esp, }
        Interfaces using crypto map l2tpsec:
                FastEthernet0/0
```

Step 3 To verify that packets are being encrypted/decrypted for the secure tunnel, use the **show crypto engine connections active** command.

```
router#show crypto engine connections active

  ID Interface       IP-Address      State  Algorithm           Encrypt  Decrypt
   1 FastEthernet0/0 10.0.0.13       set    HMAC_SHA+DES_56_CB        0        0
2000 FastEthernet0/0 10.0.0.13       set    HMAC_SHA+DES_56_CB        0       62
2001 FastEthernet0/0 10.0.0.13       set    HMAC_SHA+DES_56_CB       64        0
```

Configuration Examples

This section provides the following configuration examples:

- Configuring IPSec Protection of LAC-Initiated L2TP Tunnels Example
- Configuring IPSec Protection of Client-Initiated L2TP Tunnels Example

Configuring IPSec Protection of LAC-Initiated L2TP Tunnels Example

The following example configures L2TP Security on the client, LAC, and LNS for a compulsory tunneling scenario.

Client Configuration

```
! PPP configuration on the client.
interface Serial1/0
 ip address negotiated
 encapsulation ppp
 clockrate 128000
 no cdp enable
 ppp chap hostname userSerial10@cisco.com
 ppp chap password cisco
```

LAC Configuration

```
! Passwords for the L2TP tunnel authentication.
username LAC password 0 cisco
username LNS password 0 cisco
!
! VPDN configuration to tunnel users with the domain cisco.com
! to the LNS.  This configuration has l2tp tunnel authentication
! enabled.
!
vpdn enable
no vpdn logging
!
vpdn-group 1
 request-dialin
  protocol l2tp
  domain cisco.com
 initiate-to ip 10.0.0.13
 local name LAC
 l2tp security crypto-profile l2tp keep-sa
 l2tp tunnel password cisco
!
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco address 10.0.0.13
!
crypto ipsec transform-set esp-des-sha-transport esp-des esp-sha-hmac
 mode transport
!
! Crypto profile configuration which is bound to the vpdn-group shown above.
crypto map l2tpsec 10 ipsec-isakmp profile l2tp
 set transform-set esp-des-sha-transport
!
interface FastEthernet0/0
 ip address 10.0.0.12 255.255.255.0
 crypto map l2tpsec
```

LNS Configuration

```
! PPP client username and password needed for CHAP authentication.
username userSerial10@cisco.com password 0 cisco
!
! Passwords for the L2TP tunnel authentication.
username LAC password 0 cisco
username LNS password 0 cisco
!
! Using address pool to assign client an IP address.
ip address-pool local
!
! VPDN configuration.
vpdn enable
!
vpdn-group 1
 accept-dialin
  protocol any
  virtual-template 1
 terminate-from hostname LAC
 lcp renegotiation on-mismatch
 l2tp security crypto-profile l2tp keep-sa
!
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco address 10.0.0.12
!
crypto ipsec transform-set esp-des-sha-transport esp-des esp-sha-hmac
 mode transport
!
crypto map l2tpsec 10 ipsec-isakmp profile l2tp
 set transform-set esp-des-sha-transport
!
interface FastEthernet0/0
 ip address 10.0.0.13 255.255.255.0
 speed 10
 half-duplex
 crypto map l2tpsec
```

Configuring IPSec Protection of Client-Initiated L2TP Tunnels Example

The following example configures L2TP Security on the LNS for a voluntary tunneling scenario.

LNS Configuration

```
! PPP client username and password needed for CHAP authentication.
username userSerial10@cisco.com password 0 cisco
! Passwords for the L2TP tunnel authentication.
username LAC password 0 cisco
username LNS password 0 cisco
!
! Using address pool to assign client an IP address.
ip address-pool local
!
! VPDN configuration.
vpdn enable
!
vpdn-group dial-in
 accept-dialin
  protocol l2tp
  virtual-template 1
 l2tp security crypto-profile l2tp
 no l2tp tunnel authentication
ip pmtu
!
crypto ipsec transform-set esp-des-sha-transport esp-des esp-sha-hmac
 mode transport
!
crypto map l2tpsec 10 ipsec-isakmp profile l2tp
 set transform-set esp-des-sha-transport
 set security-association lifetime seconds 120
!
interface FastEthernet0/0
 ip address 10.0.0.13 255.255.255.0
 speed 10
 half-duplex
 crypto map l2tpsec
```

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

New Commands

- **ip pmtu**
- **l2tp security crypto-profile**

Modified Commands

- **crypto map (global IPSec)**

Low Latency Queueing (LLQ) for IPSec Encryption Engines

Feature History

Release	Modification
12.2(13)T	This feature was introduced.
12.2(14)S	This feature was integrated into Cisco IOS Release 12.2(14)S.

This feature module describes the Low Latency Queueing (LLQ) for IPSec encryption engines feature in Cisco IOS Release 12.2(13)T and 12.2(14)S. It includes the following sections:

- Feature Overview, page 1263
- Supported Platforms, page 1265
- Supported Standards, MIBs, and RFCs, page 1266
- Prerequisites, page 1266
- Configuration Tasks, page 1267
- Monitoring and Maintaining LLQ for IPSec Encryption Engines, page 1270
- Configuration Examples, page 1270
- Command Reference, page 1271
- Glossary, page 1271

Feature Overview

Low Latency Queueing (LLQ) for IPSec encryption engines helps reduce packet latency by introducing the concept of queueing before crypto engines. Prior to this, the crypto processing engine gave data traffic and voice traffic equal status. Administrators now designate voice traffic as priority. Data packets arriving at a router interface are directed into a data packet inbound queue for crypto engine processing. This queue is called the best effort queue. Voice packets arriving on a router interface are directed into a priority packet inbound queue for crypto engine processing. This queue is called the priority queue. The crypto engine undertakes packet processing in a favorable ratio for voice packets. Voice packets are guaranteed a minimum processing bandwidth on the crypto engine.

Benefits

The Low Latency Queueing (LLQ) for IPSec encryption engines feature guarantees a certain level of crypto engine processing time for priority designated traffic.

Note On the Cisco 2600 platform, with the exception of the Cisco 2691 router, the CPU utilization maximizes out before the crypto engine becomes congested, so latency is not improved.

Better Voice Performance

Voice packets can be identified as priority, allowing the crypto engine to guarantee a certain percentage of processing bandwidth. This feature impacts the end user experience by assuring voice quality if voice traffic is directed onto a congested network.

Improved Latency and Jitters

Predictability is a critical component of network performance. The Low Latency Queueing (LLQ) for IPSec encryption engines feature delivers network traffic predictability relating to VPN. With this feature disabled, an end user employing an IP phone over VPN might experience jitter or latency, both symptoms of overall network latency and congestion. With this feature enabled, these undesirable characteristics are dissipated.

Restrictions

- No per-tunnel QoS policy. An interface QoS policy represents all tunnels.
- Assume the same IP precedence/DSCP marking for inbound and outbound voice packets.
- Assume the IP precedence/DSCP marking for voice packets are done at the source.
- Limited match criteria for voice traffic in the interface QoS policy.
- Assume call admission control is enforced within the enterprise.
- No strict error checking when aggregate policy's bandwidth exceeds crypto engine bandwidth. Only a warning is displayed but configuration is allowed.
- Assume voice packets are either all encrypted or unencrypted.

Related Features and Technologies

- CBWFQ
- Priority Queueing
- Weighted Fair Queueing

Related Documents

- *Quality of Service Solutions Command Reference*, Cisco IOS Release 12.2
- *Class-Based Weighted Fair Queueing* feature module, Cisco IOS Release 12.1
- *IP RTP Priority* feature module, Cisco IOS Release 12.0

Supported Platforms

12.2(14)S and higher

The LLQ for IPSec encryption engines feature is supported on the following platform:

- Cisco 7200 series

12.2(13)T

The LLQ for IPSec encryption engines feature is supported on all platforms using Cisco IOS Release 12.2(13)T or later, including:

- Cisco 2600 series
- Cisco 3600 series
- Cisco 7100 series
- Cisco 7200 series

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that are supported on specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side-by-side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Supported Standards, MIBs, and RFCs

Standards

- No new or modified standards are supported by this feature.

MIBs

- No new or modified standards are supported by this feature.

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

- No new or modified RFCs are supported by this feature.

Prerequisites

To use this feature, you should be familiar with the following:

- Access control lists
- Bandwidth management
- CBWFQ

Configuration Tasks

To configure LLQ for IPSec encryption engines, perform the tasks described in the following section.

Note See the *Quality of Service Solutions Command Reference*, Cisco IOS Release 12.2, to learn more about configuring server policies on interfaces.

- Defining Class Maps (required)
- Configuring Class Policy in the Policy Map (required)
- Configuring Class Policy for a Priority Queue (required)
- Configuring Class Policy Using a Specified Bandwidth (optional)
- Configuring the Class-Default Class Policy (optional)
- Attaching the Service Policy (required)
- Verifying Configuration of Policy Maps and Their Classes (optional)

Defining Class Maps

To create a class map containing match criteria against which a packet is checked to determine if it belongs to a class, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# class-map class-map-name`	Specifies the name of the class map to be created.
Step 2	`Router(config-cmap)# match access-group {access-group | name access-group-name}` or `Router(config-cmap)# match input-interface interface-name` or `Router(config-cmap)# match protocol protocol`	Specifies the name of the access control list (ACL) against whose contents packets are checked to determine if they belong to the class. Specifies the name of the input interface used as a match criterion against which packets are checked to determine if they belong to the class. Specifies the name of the protocol used as a match criterion against which packets are checked to determine if they belong to the class.

Configuring Class Policy in the Policy Map

To configure a policy map and create class policies that make up the service policy, begin with the **policy-map** command to specify the policy map name. Then use one or more of the following commands to configure the policy for a standard class or the default class:

- **priority**
- **bandwidth**
- **queue-limit** or **random-detect**

- **fair-queue** (for class-default class only)

For each class that you define, you can use one or more of the commands listed to configure the class policy. For example, you might specify bandwidth for one class and both bandwidth and queue limit for another class.

The default class of the policy map (commonly known as the class-default class) is the class to which traffic is directed if that traffic does not satisfy the match criteria of the other classes defined in the policy map.

You can configure class policies for as many classes as are defined on the router, up to the maximum of 64. However, the total amount of bandwidth allocated for all classes in a policy map must not exceed the minimum committed information rate (CIR) configured for the virtual circuit (VC) minus any bandwidth reserved by the **frame-relay voice bandwidth** and **frame-relay ip rtp priority** commands. If the minimum CIR is not configured, the bandwidth defaults to one half of the CIR. If all of the bandwidth is not allocated, the remaining bandwidth is allocated proportionally among the classes on the basis of their configured bandwidth.

To configure class policies in a policy map, perform the tasks described in the following sections. The task in the first section is required; the tasks in the remaining sections are optional.

Configuring Class Policy for a Priority Queue

To configure a policy map and give priority to a class within the policy map, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map policy-map`	Specifies the name of the policy map to be created or modified.
Step 2	`Router(config-cmap)# class class-name`	Specifies the name of a class to be created and included in the service policy.
Step 3	`Router(config-pmap-c)# priority bandwidth-kbps`	Creates a strict priority class and specifies the amount of bandwidth, in kbps, to be assigned to the class.

Configuring Class Policy Using a Specified Bandwidth

To configure a policy map and create class policies that make up the service policy, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map policy-map`	Specifies the name of the policy map to be created or modified.
Step 2	`Router(config-cmap)# class class-name`	Specifies the name of a class to be created and included in the service policy.
Step 3	`Router(config-pmap-c)# bandwidth bandwidth-kbps`	Specifies the amount of bandwidth to be assigned to the class, in kbps, or as a percentage of the available bandwidth. Bandwidth must be specified in kbps or as a percentage consistently across classes. (Bandwidth of the priority queue must be specified in kbps.)

To configure more than one class in the same policy map, repeat Step 2 and Step 3.

Configuring the Class-Default Class Policy

The class-default class is used to classify traffic that does not fall into one of the defined classes. Even though the class-default class is predefined when you create the policy map, you still have to configure it. If a default class is not configured, then traffic that does not match any of the configured classes is given best-effort treatment, which means that the network will deliver the traffic if it can, without any assurance of reliability, delay prevention, or throughput.

To configure a policy map and the class-default class, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# policy-map policy-map`	Specifies the name of the policy map to be created or modified.
Step 2	`Router(config-cmap)# class class-default default-class-name`	Specifies the default class so that you can configure or modify its policy.
Step 3	`Router(config-pmap-c)# bandwidth bandwidth-kbps` `or` `Router(config-pmap-c)# fair-queue [number-of-dynamic-queues]`	Specifies the amount of bandwidth, in kbps, to be assigned to the class. Specifies the number of dynamic queues to be reserved for use by flow-based WFQ running on the default class. The number of dynamic queues is derived from the bandwidth of the interface.

Attaching the Service Policy

To attach a service policy to the output interface and enable LLQ for IPSec encryption engines, use the following command in map-class configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type number`	Specifies the interface using the LLQ for IPSec encryption engines.
Step 2	`Router(config-if)# service-policy output policy-map`	Attaches the specified service policy map to the output interface and enables LLQ for IPSec encryption engines.

Verifying Configuration of Policy Maps and Their Classes

To display the contents of a specific policy map or all policy maps configured on an interface, use the following commands in EXEC mode, as needed:

	Command	Purpose
Step 1	`Router# show frame-relay pvc dlci`	Displays statistics about the PVC and the configuration of classes for the policy map on the specified data-link connection identifier (DLCI).

	Command	Purpose
Step 2	`Router# show policy-map interface interface-name`	When LLQ is configured, displays the configuration of classes for all policy maps.
Step 3	`Router# show policy-map interface interface-name dlci dlci`	When LLQ is configured, displays the configuration of classes for the policy map on the specified DLCI.

Monitoring and Maintaining LLQ for IPSec Encryption Engines

To monitor and maintain LLQ for IPSec encryption engines, use the following command in EXEC mode:

	Command	Purpose
Step 1	`Router# show crypto eng qos`	Displays quality of service queueing statistics for LLQ for IPSec encryption engines.

For a more detailed list of commands that can be used to monitor LLQ for IPSec encryption engines, see the section "Verifying Configuration of Policy Maps and Their Classes"

Configuration Examples

This section provides the following configuration example:

- LLQ for IPSec Encryption Engines Example

LLQ for IPSec Encryption Engines Example

In the following example, a strict priority queue with a guaranteed allowed bandwidth of 50 kbps is reserved for traffic that is sent from the source address 10.10.10.10 to the destination address 10.10.10.20, in the range of ports 16384 through 20000 and 53000 through 56000.

First, the following commands configure access list 102 to match the desired voice traffic:

```
Router(config)# access-list 102 permit udp host 10.10.10.10 host 10.10.10.20 range 16384
20000
Router(config)# access-list 102 permit udp host 10.10.10.10 host 10.10.10.20 range 53000
56000
```

Next, the class map voice is defined, and the policy map called policy1 is created; a strict priority queue for the class voice is reserved, a bandwidth of 20 kbps is configured for the class bar, and the default class is configured for WFQ. The service-policy command then attaches the policy map to the fas0/0.

```
Router(config)# class-map voice
Router(config-cmap)# match access-group 102
Router(config)# policy-map policy1
Router(config-pmap)# class voice
Router(config-pmap-c)# priority 50
Router(config-pmap)# class bar
Router(config-pmap-c)# bandwidth 20
Router(config-pmap)# class class-default
Router(config-pmap-c)# fair-queue
Router(config)# interface fas0/0
```

```
Router(config-if)# service-policy output policy1
```

Command Reference

The following new command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **show crypto eng qos**

Glossary

IKE—Internet Key Exchange. IKE establishes a shared security policy and authenticates keys for services (such as IPSec). Before any IPSec traffic can be passed, each router/firewall/host must verify the identity of its peer. This can be done by manually entering preshared keys into both hosts or by a CA service.

IPSec—IP Security. A framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer. IPSec uses IKE to handle the negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPSec. IPSec can protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

L2TP—IPSec Support for NAT and PAT Windows Clients

The L2TP—IPSec Support for NAT and PAT Windows Clients feature allows more than one Windows client to connect to a Layer 2 Tunneling Protocol (L2TP) Network Server (LNS) at one time with IP Security (IPSec) enabled and a network address translation (NAT) or port address translation (PAT) server between the Windows client and LNS.

Currently, if one Windows client is connected to a Cisco IOS LNS router through a NAT or PAT server with IPSec enabled, and then another Windows client connects to the same Cisco IOS LNS router, the first client's connection is effectively terminated. Enabling L2TP—IPSec Support for NAT and PAT Windows Clients ensures that Windows client connections in this environment are established and maintained until the connection is closed.

History for the L2TP—IPSec Support for NAT and PAT Windows Clients Feature

Release	Modification
12.3(11)T4	This feature was introduced.
12.4(1)	This feature was integrated into Release 12.4(1).

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for L2TP—IPSec Support for NAT and PAT Windows Clients, page 1274
- Restrictions for L2TP—IPSec Support for NAT and PAT Windows Clients, page 1274
- Information About L2TP—IPSec Support for NAT and PAT Windows Clients, page 1274
- How to Enable L2TP—IPSec Support for NAT and PAT Windows Clients, page 1276
- Configuration Examples for L2TP—IPSec Support for NAT and PAT Windows Clients, page 1278
- Additional References, page 1280
- Command Reference, page 1282

Prerequisites for L2TP—IPSec Support for NAT and PAT Windows Clients

- You have an environment consisting of Windows clients and Cisco IOS LNS routers with IPSec enabled and a NAT or PAT server between the Windows client and LNS router.
- You must have a version of IPSec that contains the L2TP—IPSec Support for NAT and PAT Windows Clients feature.
- You must understand Windows 2000 concepts and configuration requirements.
- You must understand Cisco IOS LNS routers concepts and configuration requirements.
- You must understand NAT and PAT concepts and configuration requirements.
- You must understand IPSec concepts and configuration requirements.
- You must understand L2TP concepts and configuration requirements.

Restrictions for L2TP—IPSec Support for NAT and PAT Windows Clients

- Tested with Windows 2000 clients only.
- Port translation is not a standard default behavior. Port translation is incompatible with standard IPSec because it changes the LNS header port information.
- L2TP requires the client to have Microsoft DUN configured. L2TP is supported solely by Windows 2000 MS-DUN (L2TP is not supported by Windows 95, Windows 98, or Windows NT).

Information About L2TP—IPSec Support for NAT and PAT Windows Clients

To use the L2TP—IPSec Support for NAT and PAT Windows Clients feature, the following concept should be understood:

- How L2TP—IPSec Support for NAT and PAT Windows Clients Works, page 1274

How L2TP—IPSec Support for NAT and PAT Windows Clients Works

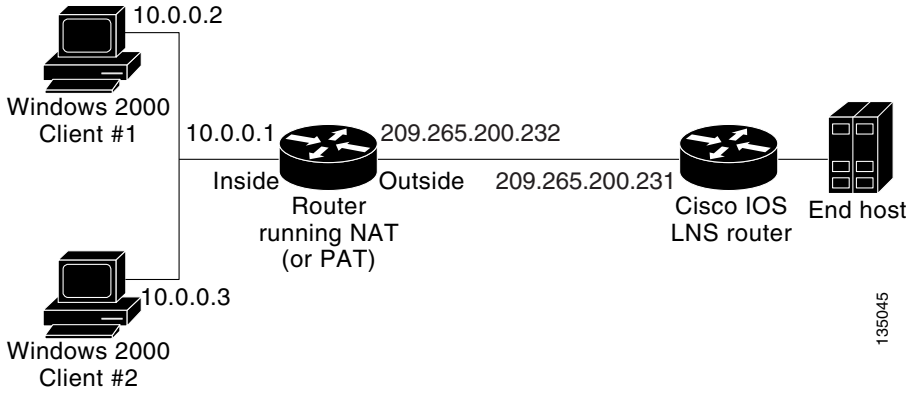
With the L2TP—IPSec Support for NAT and PAT Windows Clients feature not enabled, Windows clients lose connection with the Cisco IOS LNS router when another Windows client establishes an IPSec-protected L2TP tunnel to the Cisco IOS LNS router when IPSec is enabled and there is a NAT or PAT server between the Windows clients and the LNS.

Note If you do not have IPSec enabled, or you do not have a NAT or PAT server, you can have multiple Windows clients connect to a LNS without this command enabled.

Without L2TP—IPSec Support for NAT and PAT Windows Clients Feature Enabled

For example, Figure 91 shows two Windows 2000 clients that are trying to connect to the end host through the router running NAT or PAT and the same Cisco IOS LNS router. IPSec is enabled.

Figure 91 Multiple Windows 2000 Clients, NAT Router, and Cisco IOS LNS Router with IP Addresses



The Windows 2000 Client #1 establishes an IPSec-protected L2TP tunnel to the Cisco IOS LNS router. The Windows 2000 client and the Cisco IOS LNS router recognize that there is a router running NAT between them and IPSec and NAT-Traversal (NAT-T) are enabled. The Windows 2000 client attempts to establish an IPSec security association (SA) and requests transport mode (which it does by default) with proxies from 10.0.0.2, its local address, to 209.265.200.231, the Cisco IOS LNS router's address.

In transport mode NAT, running on the router, translates all outgoing connections (including 10.0.0.2) to its outside IP address (209.265.200.232),the address the traffic will come in on. However, NAT cannot modify the L2TP port designation (1701), which is protected by the IPSec encrypted area. So now, we have a local address of 209.265.200.231, a remote address of 209.265.200.232 and a remote port of 1701. All traffic is sent to the Windows 2000 Client #1 that matches the tunnel 209.265.200.231, port 1701.

Then Windows 2000 Client #2 establishes an IPSec-protected L2TP tunnel to the Cisco IOS LNS router, again in transport mode. And NAT, again, translates all outgoing connections to its outside IP address (209.265.200.232), but it cannot modify the L2TP port designation (1701). All traffic is now sent to Windows 2000 Client #2 that matches tunnel 209.265.200.231, port 1701. This second Windows client connection has effectively ended Windows Client #1's connection to the Cisco IOS LNS router since it is no longer receiving traffic.

With L2TP—IPSec Support for NAT and PAT Windows Clients Feature Enabled

With the L2TP—IPSec Support for NAT and PAT Windows Clients feature enabled, IPSec can translate the L2TP ports after decryption. This feature allows IPSec to map traffic from different hosts to different source ports. L2TP can now distinguish between traffic destined for multiple Windows 2000 clients.

So now, when an SA is created, a translated port will be assigned to it. This port is client-specific. The same port will be used for any new SA created by that client. When an encrypted request is received and decrypted, the source port is translated from the standard value, 1701, to a client specific value. The request with the translated port is then forwarded to L2TP.

As shown in Figure 91 with port translation enabled, the Windows 2000 Client #1 would have a translated port number of 1024 assigned and Windows 2000 Client #2 would have a translated port number of 1025 assigned.

When L2TP sends the reply packet, it uses the translated port number and creates a packet to that destination port. IPSec uses the destination port number to select the SA with which to encrypt the packet. Before encrypting the packet, IPSec translates the destination port back to the standard port number, 1701, which the Windows 2000 client expects. IPSec encrypts the packet, either with the SA to Windows 2000 Client #1 if the destination port was 1024 or with the SA to Windows 2000 Client #2 if the destination port was 1025. And now, all traffic is sent to the appropriate client and multiple Windows clients can be connected to a Cisco IOS LNS router through a NAT server at the same time.

The connection is maintained until one of the following actions occurs:

- The IPSec connection is closed.
- The NAT or PAT device ends the session.
- The LNS closes the session.
- The Windows client closes the session.

How to Enable L2TP—IPSec Support for NAT and PAT Windows Clients

This section contains the following procedure that allows you to enable NAT/PAT port translation:

- Enabling L2TP—IPSec Support, page 1276

Enabling L2TP—IPSec Support

Use the following task to enable L2TP—IPSec Support for NAT and PAT Windows Clients for environments that have IPSec enabled and include multiple windows clients, a NAT or PAT server, L2TP, and a Cisco IOS LNS router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* [**ipsec-isakmp**]

   or

   crypto dynamic-map *dynamic-map-name dynamic-seq-num*

4. **set nat demux**
5. **exit**
6. **exit**
7. **show crypto map** [**interface** *interface* | **tag** *map-name*]

   or

   show crypto dynamic-map [**tag** *map-name*]
8. **show crypto ipsec sa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* [**ipsec-isakmp**] **Example:** `Router(config)# crypto map STATIC_MAP 5` or **crypto dynamic-map** *dynamic-map-name dynamic-seq-num* Example: `Router(config)# crypto dynamic-map DYNAMIC_MAP 10`	Names the static crypto map entry to create (or modify) and enters crypto map configuration mode. or Names the dynamic crypto map entry to create (or modify) and enters crypto map configuration mode.
Step 4	**set nat demux** **Example:** `Router(config-crypto-map)# set nat demux`	Enables L2TP—IPSec support.
Step 5	**exit** **Example:** `Router(config-crypto-map)# exit`	Exits crypto map configuration mode and returns to global configuration mode.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 7	**show crypto map** [**interface** *interface* | **tag** *map-name*] **Example:** `Router# show crypto map` or **show crypto dynamic-map** [**tag** *map-name*] **Example:** `Router# show crypto dynamic-map`	(Optional) Displays information about crypto map configuration. or (Optional) Displays information about dynamic crypto map configuration.
Step 8	**show crypto ipsec sa** **Example:** `Router# show crypto ipsec sa`	(Optional) Displays the settings used by current SAs.

Configuration Examples for L2TP—IPSec Support for NAT and PAT Windows Clients

This section provides the following configuration example:

- Dynamic Map Configuration: Example, page 1278

Dynamic Map Configuration: Example

The following example shows how to enable the L2TP—IPSec Support for NAT and PAT Windows Clients feature for a dynamic crypto map:

```
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname 72_LNS
!
boot-start-marker
boot-end-marker
!
aaa new-model
!
!
aaa authentication ppp default local
aaa session-id common
ip subnet-zero
!
!
no ip cef
no ip domain lookup
ip domain name cisco.com
```

```
ip dhcp excluded-address 20.0.0.8
ip dhcp excluded-address 20.0.0.10
!
!
ip vrf VPN
 rd 1:1
!
!Enable virtual private networking.
vpdn enable
vpdn ip udp ignore checksum
!
! Default L2TP VPDN group
vpdn-group L2TP
!
!Enables the LNS to accept dial in requests; specifies L2TP as the tunneling
!protocol; specifies the number of the virtual templates used to clone
!virtual-access interfaces
 accept-dialin
  protocol l2tp
  virtual-template 1

!Disables L2TP tunnel authentication.
no l2tp tunnel authentication
!
!
crypto keyring L2TP
  pre-shared-key address 0.0.0.0 0.0.0.0 key *****
!
!Defines an Internet Key Exchange (IKE) policy and assigns priority 1.
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
 lifetime 3600
!
crypto isakmp key cisco hostname w2k01
crypto isakmp keepalive 3600
!
crypto ipsec security-association lifetime seconds 600
!
!Defines a transform set.
crypto ipsec transform-set TS1 esp-3des esp-sha-hmac
 mode transport
!
!Names the dynamic crypto map entry and enters crypto map configuration mode; Enables
!L2TP—IPSec support; Specifies which transform sets can be used with the crypto map
!entry
crypto dynamic-map DYN_MAP 10
 set nat demux
 set transform-set TS1!
!
crypto map CRYP_MAP 6000 ipsec-isakmp dynamic DYN_MAP
!
interface Loopback0
 ip address 12.0.0.8 255.255.255.255
!
interface FastEthernet0/0
 ip address 11.0.0.8 255.255.255.0
 no ip route-cache
 duplex full
 speed 100
 crypto map CRYP_MAP
!
```

```
interface FastEthernet0/1
 ip address 20.0.0.8 255.255.255.0
 duplex full
 speed 100
!
interface FastEthernet2/0
 ip address 172.19.192.138 255.255.255.0
 duplex full
!
interface Virtual-Template1
 ip unnumbered Loopback0
 peer default ip address pool POOL
 ppp mtu adaptive
 ppp authentication chap ms-chap
!
router ospf 1
 log-adjacency-changes
 redistribute static subnets
 network 11.0.0.0 0.0.0.255 area 0
!
ip local pool POOL 20.0.0.100 20.0.0.110
ip classless
ip route 171.0.0.0 255.0.0.0 172.19.192.1
!
no ip http server
no ip http secure-server
!
!
control-plane
!
gatekeeper
 shutdown!
!
line con 0
 exec-timeout 0 0
 logging synchronous
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
!
end
```

Additional References

The following sections provide references related to L2TP—IPSec Support for NAT and PAT Windows Clients.

Related Documents

Related Topic	Document Title
IP Security and Encryption Overview	*Cisco IOS Security Configuration Guide*, Release 12.3
Configuring IPSec Network Security	*Cisco IOS Security Configuration Guide*, Release 12.3

Standards

Standard	Title
None	–

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	–

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Command Reference

The following new command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **set nat demux**

Pre-Fragmentation for IPSec VPNs

Feature History

Release	Modification
12.1(11b)E	This feature was introduced.
12.2(13)T	This feature was integrated into Cisco IOS Release 12.2(13)T.
12.2(14)S	This feature was integrated into Cisco IOS Release 12.2(14)S.

This feature module describes the Pre-fragmentation for IPSec VPNs feature in Cisco IOS Release 12.2(13)T and 12.2(14)S. It includes the following sections:

- Feature Overview, page 1283
- Supported Platforms, page 1285
- Configuration Tasks, page 1287
- Configuration Tasks, page 1287
- Configuration Examples, page 1289
- Command Reference, page 1290

Feature Overview

When a packet is nearly the size of the maximum transmission unit (MTU) of the outbound link of the encrypting router, and it is encapsulated with IPSec headers, it is likely to exceed the MTU of the outbound link. This causes packet fragmentation after encryption, which makes the decrypting router reassemble in the process path. Pre-fragmentation for IPSec VPNs increases the decrypting router's performance by enabling it to operate in the high performance CEF path instead of the process path.

This feature allows an encrypting router to predetermine the encapsulated packet size from information available in transform sets, which are configured as part of the IPSec security association (SA). If it is predetermined that the packet will exceed the MTU of the output interface, the packet is fragmented before encryption. This function avoids process level reassembly before decryption and helps improve decryption performance and overall IPSec traffic throughput.

Note The pre-fragmentation feature is turned off by default for tunnel interfaces. To receive pre-fragmentation performance benefits, turn pre-fragmentation on after insuring that the tunnel interfaces have the same MTU on both ends.

Benefits

Increased Performance

Delivers encryption throughput at maximum encryption hardware accelerator speeds. This performance increase is for near MTU-sized packets.

Uniform Fragmentation

Packets are fragmented into equally sized units to prevent further downstream fragmentation.

Interoperability

This feature is interoperable with all Cisco IOS platforms and a number of Cisco VPN clients.

Restrictions

Take the following information into consideration before this feature is configured:

- Pre-fragmentation for IPSec VPNs operates in IPSec tunnel mode and IPSec tunnel mode with GRE, but not with IPSec transport mode.
- Pre-fragmentation for IPSec VPNs configured on the decrypting router in a unidirectional traffic scenario does not improve the performance or change the behavior of either of the peers.
- Pre-fragmentation for IPSec VPNs occurs before the transform is applied if compression is turned on for outgoing packets.
- Pre-fragmentation for IPSec VPNs functionality depends on the egress interface **crypto ipsec df-bit** configuration and the incoming packet "do not fragment" (DF) bit state. See Table 54.

Table 54 Pre-Fragmentation for IPSec VPNs Dependencies

Pre-Fragmentation for IPSec VPNs Feature State (Enabled/Disabled)	Egress Interface "crypto ipsec df-bit" Configuration	Incoming Packet DF Bit State	Result
Enabled	**crypto ipsec df-bit clear**	0	Fragmentation occurs before encryption.
Enabled	**crypto ipsec df-bit clear**	1	Fragmentation occurs before encryption.
Disabled	**crypto ipsec df-bit clear**	0	Fragmentation occurs after encryption and packets are reassembled before decryption.
Disabled	**crypto ipsec df-bit clear**	1	Fragmentation occurs after encryption and packets are reassembled before decryption.
Enabled	**crypto ipsec df-bit set**	0	Fragmentation occurs before encryption.

Table 54 Pre-Fragmentation for IPSec VPNs Dependencies (continued)

Pre-Fragmentation for IPSec VPNs Feature State (Enabled/Disabled)	Egress Interface "crypto ipsec df-bit" Configuration	Incoming Packet DF Bit State	Result
Enabled	**crypto ipsec df-bit set**	1	Packets are dropped.
Disabled	**crypto ipsec df-bit set**	0	Fragmentation occurs after encryption and packets are reassembled before decryption.
Disabled	**crypto ipsec df-bit set**	1	Packets are dropped.
Enabled	**crypto ipsec df-bit copy**	0	Fragmentation occurs before encryption.
Enabled	**crypto ipsec df-bit copy**	1	Packets are dropped.
Disabled	**crypto ipsec df-bit copy**	0	Fragmentation occurs after encryption and packets are reassembled before decryption.
Disabled	**crypto ipsec df-bit copy**	1	Packets are dropped.

Supported Platforms

12.2(14)S and higher

The Pre-fragmentation for IPSec VPN feature is supported on the following platforms:

- Cisco 7200 series
- Cisco 7400 series

12.2(13)T

The Pre-fragmentation for IPSec VPN feature is supported on all platforms using Cisco IOS Release 12.2(13)T or higher, including:

- Cisco 1710
- Cisco 1720
- Cisco 1721
- Cisco 1751
- Cisco 1760
- Cisco 2600
- Cisco 2691
- Cisco 3620
- Cisco 3640
- Cisco 3660

- Cisco 3725
- Cisco 3745
- Cisco 7100 series
- Cisco 7200 series
- Cisco 7400 series

12.1(11b)E

The Pre-fragmentation for IPSec VPN feature is supported on all platforms using Cisco IOS Release 12.1(11b)E or higher, including:

- Cisco 7100 series

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that are supported on specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Supported Standards, MIBs, and RFCs

Standards

- No new or modified standards are supported by this feature.

MIBs

- No new or modified standards are supported by this feature.

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

- No new or modified RFCs are supported by this feature.

Configuration Tasks

See the following sections for configuration tasks for the Pre-fragmentation for IPSec VPNs feature. Each task in the list is identified as either required or optional.

- Configuring Pre-Fragmentation For IPSec VPNs (required)
- Verifying Pre-Fragmentation For IPSec VPNs (optional)

Configuring Pre-Fragmentation For IPSec VPNs

Pre-fragmentation for IPSec VPNs is globally enabled by default. To enable or disable pre-fragmentation for IPSec VPNs while in interface configuration mode, enter the commands in the following table. Use the **no** form of the commands to revert back to the default configuration, or use the commands themselves to enable configuration of the pre-fragmentation IPSec VPNs.

Note Manually enabling or disabling this feature will override the global configuration.

Command	Purpose
`Router(config-if)# crypto ipsec fragmentation before-encryption`	Enables pre-fragmentation for IPSec VPNs on the interface.
`Router(config-if)# crypto ipsec fragmentation after-encryption`	Disables pre-fragmentation for IPSec VPNs on the interface.
`Router(config)# crypto ipsec fragmentation before-encryption`	Enables pre-fragmentation for IPSec VPNs globally.
`Router(config)# crypto ipsec fragmentation after-encryption`	Disables pre-fragmentation for IPSec VPNs globally.

Verifying Pre-Fragmentation For IPSec VPNs

To verify that this feature is enabled, consult the interface statistics on the encrypting router and the decrypting router. If fragmentation occurs on the encrypting router, and no reassembly occurs on the decrypting router, fragmentation is happening before encryption, and thus the packets are not being reassembled before decryption. This means that the feature is enabled.

Note This method of verification does not apply to packets destined for the decrypting router.

Step 1 Enter the **show running-configuration** command on the encrypting router. If the feature is enabled, you will observe output similar to the following:

```
Router# show running-configuration
crypto isakmp policy 10
 authentication pre-share
crypto isakmp key abcd123 address 25.0.0.7
!
!
crypto ipsec transform-set fooprime esp-3des esp-sha-hmac
!
crypto map bar 10 ipsec-isakmp
 set peer 25.0.0.7
 set transform-set fooprime
 match address 102
```

If the feature has been disabled, you will observe output similar to the following:

```
Router# show running-configuration
crypto isakmp policy 10
 authentication pre-share
crypto isakmp key abcd123 address 25.0.0.7
!
!
crypto ipsec transform-set fooprime esp-3des esp-sha-hmac
crypto ipsec fragmentation after-encryption
!
crypto map bar 10 ipsec-isakmp
 set peer 25.0.0.7
 set transform-set fooprime
 match address 102
```

Step 2 Enter the **show running-configuration interface** *type number* command to display statistics for the encrypting router egress interface. If the feature is enabled, you will observe output similar to the following:

```
Router# show running-configuration interface fastethernet 0/0
interface FastEthernet0/0
 ip address 25.0.0.6 255.0.0.0
 no ip mroute-cache
 load-interval 30
 duplex full
 speed 100
 crypto map bar
```

If the feature has been disabled, you will observe output similar to the following:

```
Router# show running-configuration interface fastethernet 0/0

interface FastEthernet0/0
 ip address 25.0.0.6 255.0.0.0
 no ip mroute-cache
 load-interval 30
 duplex full
 speed 100
 crypto map bar
 crypto ipsec fragmentation after-encryption
```

Configuration Examples

This section provides the following configuration example:

- Enabling Pre-Fragmentation For IPSec VPNs Example

Enabling Pre-Fragmentation For IPSec VPNs Example

The following configuration example shows how to configure the Pre-Fragmentation for IPSec VPNs feature:

Note This feature does not show up in the running configuration in this example because the default global pre-fragmentation for IPSec VPNs feature is enabled. Pre-fragmentation for IPSec VPNs shows in the running configuration only when you explicitly enable the feature on the interface.

```
crypto isakmp policy 10
 authentication pre-share
crypto isakmp key abcd123 address 25.0.0.7
!
!
crypto ipsec transform-set fooprime esp-3des esp-sha-hmac
!
crypto map bar 10 ipsec-isakmp
 set peer 25.0.0.7
 set transform-set fooprime
 match address 102
```

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **crypto ipsec fragmentation**
- **crypto ipsec fragmentation (interface configuration)**

Real-Time Resolution for IPSec Tunnel Peer

After a user specifies a host name (instead of an IP address) for remote IP Security (IPSec) peer, the Real-Time Resolution for IPSec Tunnel Peer feature allows the host name to be domain name server (DNS) resolved before the router establishes the IPSec tunnel. Thus, the router can immediately discover whether the IP address of the peer has changed.

Feature History for Real-Time Resolution for IPSec Tunnel Peer

Release	Modification
12.3(4)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Restrictions for Real-Time Resolution for IPSec Tunnel Peer, page 1291
- Information About Real-Time Resolution for IPSec Tunnel Peer, page 1292
- How to Configure Real-Time Resolution, page 1292
- Configuration Examples for Real-Time Resolution, page 1294
- Additional References, page 1295
- Command Reference, page 1296

Restrictions for Real-Time Resolution for IPSec Tunnel Peer

Secure DNS Requirement

It is recommended that you use this feature only with secure DNS and when the DNS responses can be authenticated. Otherwise, an attacker can spoof or forge DNS responses and have access to Internet Key Exchange (IKE) authentication data, such as a certificate. If an attacker has a certificate that is trusted

by the initiating host, the attacker can successfully establish Phase 1 IKE security association (SA), or the attacker can try to guess the preshared key that is shared between the initiator and the actual responder.

DNS Initiator

DNS names resolution for remote IPSec peers will work only if they are used as an initiator. The first packet that is to be encrypted will trigger a DNS lookup; after the DNS lookup is complete, subsequent packets will trigger IKE.

Information About Real-Time Resolution for IPSec Tunnel Peer

To configure real-time resolution for your IPSec peer, you should understand the following concept:

- Benefits of Real-Time Resolution Via Secure DNS, page 1292

Benefits of Real-Time Resolution Via Secure DNS

When specifying the host name of a remote IPSec peer via the **set peer** command, you can also issue the **dynamic** keyword, which defers DNS resolution of the host name until right before the IPSec tunnel has been established. Deferring resolution enables the Cisco IOS software to detect whether the IP address of the remote IPSec peer has changed. Thus, the software can contact the peer at the new IP address.

If the **dynamic** keyword is not issued, the host name is resolved immediately after it is specified. So, the Cisco IOS software cannot detect an IP address change and, therefore, attempts to connect to the IP address that it previously resolved.

DNS resolution assures users that their established IPSec tunnel is secure and authenticated.

How to Configure Real-Time Resolution

This section contains the following procedure:

- Configuring Real-Time Resolution for IPSec Peers, page 1292

Configuring Real-Time Resolution for IPSec Peers

Use this task to configure a router to perform real-time DNS resolution with a remote IPSec peer; that is, the host name of peer is resolved via a DNS lookup right before the router establishes a connection (an IPSec tunnel) with the peer.

Prerequisites

Before creating a crypto map, you should perform the following tasks:

- Define Internet Security Association Key Management Protocol (ISAKMP) policies.
- Define IPSec transform sets.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* **ipsec-isakmp**
4. **match address** *access-list-id*
5. **set peer** {*host-name* [**dynamic**] | *ip-address*}
6. **set transform-set** *transform-set-name1* [*transform-set-name2...transform-set-name6*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* **ipsec-isakmp** **Example:** `Router(config)# crypto map secure_b 10 ipsec-isakmp`	Specifies the crypto map entry to create (or modify) and enters crypto map configuration mode.
Step 4	**match address** *access-list-id* **Example:** `Router(config-crypto-m)# match address 140`	Names an extended access list. This access list determines which traffic should be protected by IPSec and which traffic should not be protected by IPSec in the context of this crypto map entry.
Step 5	**set peer** {*host-name* [**dynamic**]	*ip-address*} **Example:** `Router(config-crypto-m)# set peer b.cisco.com dynamic`	Specifies a remote IPSec peer. This is the peer to which IPSec-protected traffic can be forwarded. • **dynamic**—Allows the host name to be resolved via a DNS lookup just before the router establishes the IPSec tunnel with the remote peer. If this keyword is not specified, the host name will be resolved immediately after the host name is specified. Repeat for multiple remote peers.
Step 6	**set transform-set** *transform-set-name1* [*transform-set-name2...transform-set-name6*] **Example:** `Router(config-crypto-m)# set transform-set myset`	Specifies which transform sets are allowed for this crypto map entry. List multiple transform sets in order of priority (highest priority first).

Troubleshooting Tips

To display crypto map configuration information, use the **show crypto map** command.

What to Do Next

You need to apply a crypto map set to each interface through which IPSec traffic will flow. Applying the crypto map set to an interface instructs the router to evaluate all the interface's traffic against the crypto map set and to use the specified policy during connection or security association (SA) negotiation on behalf of traffic to be protected by crypto.

Configuration Examples for Real-Time Resolution

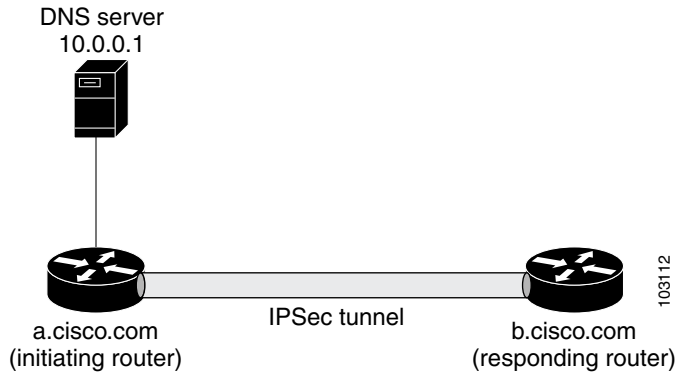
This section provides the following configuration example:

- Configuring Real-Time Resolution for an IPSec Peer: Example, page 1294

Configuring Real-Time Resolution for an IPSec Peer: Example

Figure 92 and the following example illustrate how to create a crypto map that configures the host name of a remote IPSec peer to DNS resolved via a DNS lookup right before the Cisco IOS software attempts to establish a connection with that peer.

Figure 92 Real-Time Resolution Sample Topology




```
! Configure the initiating router.
hostname a.cisco.com
ip domain name cisco.com
ip name server 10.0.0.1
!
crypto map secure_b 10 ipsec-isakmp
  match address 140
  set peer b.cisco.com dynamic
  set transform-set xset
interface serial1
  ip address 30.0.0.1
  crypto map secure_b
access-list 140 permit ...
!
```

```
! Configure the responding router (the remote IPSec peer).
hostname b.cisco.com
!
crypto map secure_a 10 ipsec-isakmp
  match address 150
  set peer 30.0.0.1
  set transform-set
interface serial0/1
  ip address 40.0.0.1
  crypto map secure_a
access-list 150 ...

! DNS server configuration

b.cisco.com    40.0.0.1       # the address of serial0/1 of b.cisco.com
```

Additional References

The following sections provide references related to Real-Time Resolution for IPSec Tunnel Peer.

Related Documents

Related Topic	Document Title
Crypto maps	The chapter "Configuring IPSec Network Security" in the *Cisco IOS Security Configuration Guide*
ISAKMP policies	The chapter "Configuring Internet Key Exchange Security Protocol" in the *Cisco IOS Security Configuration Guide*
IPSec and IKE configuration commands	*Cisco IOS Security Command Reference*, Release 12.3 T

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **set peer (IPSec)**

Reverse Route Injection

Reverse route injection (RRI) is the ability for static routes to be automatically inserted into the routing process for those networks and hosts protected by a remote tunnel endpoint. These protected hosts and networks are known as remote proxy identities.

Each route is created on the basis of the remote proxy network and mask, with the next hop to this network being the remote tunnel endpoint. By using the remote Virtual Private Network (VPN) router as the next hop, the traffic is forced through the crypto process to be encrypted.

Enhancements to the default behavior of RRI, the addition of a route tag value, and enhancements to how RRI is configured have been added to the Reverse Route Injection feature in Cisco IOS Release 12.3(14)T.

Feature History for Reverse Route Injection

Release	Modification
12.1(9)E	The Reverse Route Injection feature was introduced.
12.2(8)T	This feature was integrated into Cisco IOS Release 12.2(8)T.
12.2(9)YE	This feature was integrated into Cisco IOS Release 12.2(9)YE.
12.2(13)T	Reverse route remote peer options were added in Cisco IOS Release 12.2(13)T.
12.2(14)S	This feature was integrated into Cisco IOS Release 12.2(14)S.
12.3(14)T	The following enhancements were added to the Reverse Route Injection feature: • The default behavior of static crypto maps will be the same as that of dynamic crypto maps unless the **reverse-route** command and **static** keyword are used. • A route tag value was added for any routes that are created using RRI. • RRI can be configured on the same crypto map that is applied to multiple router interfaces. RRI configured with the **reverse-route remote-peer** {*ip-address*} command, keyword, and argument will create one route instead of two.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for Reverse Route Injection, page 1298
- Restrictions for Reverse Route Injection, page 1298
- Information About Reverse Route Injection, page 1298
- How to Configure Reverse Route Injection, page 1300
- Configuration Examples for Reverse Route Injection, page 1304
- Additional References, page 1307
- Command Reference, page 1308

Prerequisites for Reverse Route Injection

- IP routing should be enabled and static routes should be redistributed if dynamic routing protocols are to be used to propagate RRI-generated static routes.

Restrictions for Reverse Route Injection

- If RRI is applied to a crypto map, that map must be unique to one interface on the router. In other words, the same crypto map cannot be applied to multiple interfaces. If more than one crypto map is applied to multiple interfaces, routes may not be cleaned up correctly. If multiple interfaces require a crypto map, each must use a uniquely defined map. This restriction applies only to RRI prior to Cisco IOS Release 12.3(14)T.
- In the case of the **reverse route** command and **remote peer** keyword options, two routes are created for each unique IP Security (IPSec) security association (SA) flow pair. The two-route creation may lead to large numbers of routes being injected into routing tables. This restriction applies only to RRI prior to 12.3(14)T.
- For static crypto maps, routes are always present if RRI is configured on an applied crypto map. In Cisco IOS Release 12.3(14)T, the default behavior—of routes always being present for a static map—will not apply unless the **static** keyword is added to the **reverse-route** command.

Information About Reverse Route Injection

To configure the Reverse Route Injection enhancements, you should understand the following concepts:

- Reverse Route Injection, page 1298
- Reverse Route with Remote Peer Option, page 1299
- Enhancements to Reverse Route Injection, page 1299

Reverse Route Injection

RRI is the ability for static routes to be automatically inserted into the routing process for those networks and hosts that are protected by a remote tunnel endpoint. These protected hosts and networks are known as remote proxy identities.

Each route is created on the basis of the remote proxy network and mask, with the next hop to this network being the remote tunnel endpoint. By using the remote VPN router as the next hop, the traffic is forced through the crypto process to be encrypted.

After the static route is created on the VPN router, this information is propagated to upstream devices, allowing them to determine the appropriate VPN router to which to send returning traffic in order to maintain IPSec state flows. Being able to determine the appropriate VPN router is particularly useful if multiple VPN routers are used at a site to provide load balancing or failover or if the remote VPN devices are not accessible via a default route. Routes are created in either the global routing table or the appropriate virtual route forwarding (VRF) table.

RRI is applied on a per-crypto map basis, whether this is via a static crypto map or a dynamic crypto map template. The default behavior for the two map types is as follows:

- In the case of a dynamic crypto map, routes are created upon the successful establishment of IPSec SAs for those remote proxies. The next hop back to those remote proxies is via the remote VPN router whose address is learned and applied during the creation of the dynamic crypto map template. The routes are deleted after the SAs are deleted.
- For static crypto maps, routes are created on the basis of the destination information defined in the crypto access list. The next hop is taken from the first set peer statement that is attached to the crypto map. If at any time, RRI, the peer, or the access list is removed from the crypto map, routes will be deleted. This behavior changes with the addition of the RRI enhancements, as explained in the sections below.

Reverse Route with Remote Peer Option

In Cisco IOS Release 12.2(13)T, an enhancement was added to RRI to allow you to specify an interface or address as the explicit next hop to the remote VPN device. This functionality allows the overriding of a default route to properly direct outgoing encrypted packets.

Enhancements to Reverse Route Injection

The following enhancements have been added to the Reverse Route Injection feature for Cisco IOS Release 12.3(14)T:

- A change in the default behavior of static crypto maps that have crypto ACLs.

  For both dynamic and static maps, routes are created only at the time of IPSec SA creation. Routes are removed when the SAs are deleted. The **static** keyword has been added to the **reverse-route** command and is used if the behavior on static crypto maps is required, that is, if routes are created on the basis of the content of the crypto ACLs that are attached to the static crypto map.
- Ability to add a route tag value to any routes that are created using RRI.

  This route tag allows redistribution of groups of routes using route maps, allowing you to be selective about which routes enter your global routing table.
- Ability to allow RRI to be configured on the same crypto map that is applied to multiple router interfaces.

  Configuring RRI in this way is important for devices that support VLANs.
- The RRI **reverse-route remote-peer** {*ip-address*} command, keyword, and argument now creates one route instead of two:

In the current case of the **reverse-route remote-peer** {*next-hop*} command, keyword, and argument, two routes are created. One route is the standard remote proxy ID and the next hop is the remote VPN client tunnel address. The second route is the actual route to that remote tunnel endpoint in the case in which a recursive lookup should be forced to impose that the remote endpoint is reachable via "next-hop." Creation of the second route for the actual next hop has been very important in the VRF case in which a default route must be overridden by a more explicit route.

To reduce the number of routes created and support some platforms that do not readily facilitate route recursion, the **reverse-route remote-peer** {*ip-address*} command, keyword, and argument produce a modified result in which only one route is created of the following form:

ip route proxy-net proxy-mask via next-hop.

- For virtual IPSec interfaces, the reverse route option will create routes that list the virtual access interface as the next hop.
- For devices using a VPN Services Module (VPNSM), reverse route specifies the next hop to be the interface or subinterface/virtual LAN (VLAN) with the crypto map applied to it.

How to Configure Reverse Route Injection

The following sections show how to configure reverse route injection for Cisco IOS software before Release 12.3(14)T and for Release 12.3(14)T and later releases.

- Configuring RRI for Cisco IOS Releases Before 12.3(14)T, page 1300
- Configuring RRI with Enhancements, page 1302

Configuring RRI for Cisco IOS Releases Before 12.3(14)T

This section includes the following tasks:

- Configuring RRI Under a Static Crypto Map, page 1300
- Configuring RRI Under a Dynamic Map Template, page 1301

Configuring RRI Under a Static Crypto Map

To configure RRI under a static crypto map for Cisco IOS software prior to Release 12.3(14)T, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** {*map-name*} {*seq-name*} **ipsec-isakmp**
4. **reverse-route** [**remote-peer** | **remote-peer** {***ip-address***}]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** {*map-name*} {*seq-name*} **ipsec-isakmp** **Example:** `Router (config)# crypto map mymap 1 ipsec-isakmp`	Creates or modifies a crypto map entry and enters crypto map configuration mode.
Step 4	**reverse-route** [**remote-peer** | **remote-peer** {*ip-address*}] **Example:** `Router (config-crypto-map)# reverse-route remote peer 10.1.1.1`	Creates source proxy information for a crypto map entry.

Configuring RRI Under a Dynamic Map Template

To configure RRI under a dynamic map template for Cisco IOS software prior to Release 12.3(14)T, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto dynamic-map** {*dynamic-map-name*} {*dynamic-seq-name*}
4. **reverse-route** [**remote-peer** | **remote-peer** {*ip-address*}]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto dynamic-map** {dynamic-*map-name*} {dynamic-*seq-name*} **Example:** `Router (config)# crypto dynamic-map mymap 1`	Creates a dynamic crypto map entry and enters the crypto map configuration command mode.
Step 4	**reverse-route** [**remote-peer** | **remote-peer** {*ip-address*}] **Example:** `Router (config-crypto-map)# reverse-route remote peer 10.1.1.1`	Creates source proxy information for a crypto map entry.

Configuring RRI with Enhancements

The following tasks show how to configure RRI with the enhancements that were added in Cisco IOS Release 12.3(14)T:

- Configuring RRI with Enhancements Under a Static Crypto Map, page 1302
- Configuring RRI with Enhancements Under a Dynamic Map Template, page 1303

Configuring RRI with Enhancements Under a Static Crypto Map

To configure RRI with enhancements under a static crypto map (for Cisco IOS Release 12.3(14)T and later), perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** {*map-name*} {*seq-name*} **ipsec-isakmp**
4. **reverse-route** [[**static**] | **tag** {*tag-id*} [**static**] | **remote-peer** [**static**] | **remote-peer** {*ip-address*} [**static**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** {*map-name*} {*seq-name*} **ipsec-isakmp** **Example:** `Router (config)# crypto map mymap 1 ipsec-isakmp`	Creates or modifies a crypto map entry and enters crypto map configuration mode.
Step 4	**reverse-route** [[**static**] | **tag** {*tag-id*} [**static**] | **remote-peer** [**static**] | **remote-peer** {*ip-address*} [**static**]] **Example:** `Router (config-crypto-map)# reverse-route tag 5`	Creates source proxy information for a crypto map entry.

Configuring RRI with Enhancements Under a Dynamic Map Template

To configure RRI with enhancements under a dynamic map template (for Cisco IOS Release 12.3(14)T and later), perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto dynamic-map** {*dynamic-map-name*} {*dynamic-seq-name*}
4. **reverse-route** [**tag** {*tag-id*} | **remote-peer** | **remote-peer** {*ip-address*}]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto dynamic-map** {dynamic-*map-name*} {dynamic-*seq-name*} **Example:** `Router (config)# crypto dynamic-map mymap 1`	Creates a dynamic crypto map entry and enters the crypto map configuration command mode.
Step 4	**reverse-route** [**tag** {*tag-id*} | **remote-peer** | **remote-peer** {*ip-address*}] **Example:** `Router (config-crypto-map)# reverse-route remote peer 10.1.1.1`	Creates source proxy information for a crypto map entry.

Troubleshooting Tips

To observe the behavior of RRI and its relationship to the creation and deletion of an IPSec SA, you can use the **debug crypto ipsec** command.

Configuration Examples for Reverse Route Injection

This section contains the following sections:

- Configuring RRI Prior to Cisco IOS Release 12.3(14)T: Examples, page 1304
- Configuring RRI with the Enhancements Added in Cisco IOS Release 12.3(14)T: Examples, page 1305

Configuring RRI Prior to Cisco IOS Release 12.3(14)T: Examples

The following are examples of RRI configurations and output prior to Cisco IOS Release 12.3(14)T:

- Configuring RRI When Crypto ACLs Exist: Example, page 1305
- Configuring RRI When Two Routes Are Created, One for the Remote Endpoint and One for Route Recursion: Example, page 1305

Configuring RRI When Crypto ACLs Exist: Example

The following example shows that all remote VPN gateways connect to the router via 192.168.0.3. RRI is added on the static crypto map, which creates routes on the basis of the source network and source netmask that are defined in the crypto ACL:

```
crypto map mymap 1 ipsec-isakmp
 set peer 10.1.1.1
 reverse-route
 set transform-set esp-3des-sha
 match address 102

Interface FastEthernet 0/0
 ip address 192.168.0.2 255.255.255.0
 standby name group1
 standby ip 192.168.0.3
 crypto map mymap redundancy group1

access-list 102 permit ip 192.168.1.0 0.0.0.255 10.0.0.0 0.0.255.255
```

In Cisco IOS Release 12.3(14)T and later, for the static map to retain this same behavior of creating routes on the basis of crypto ACL content, the **static** keyword is required, that is, **reverse-route static**.

Note The **reverse-route** command in this situation creates routes that are analogous to the following static route command-line interface (CLI) commands (**ip route**):

Remote Tunnel Endpoint

```
ip route 10.1.1.1 255.255.255.255 192.168.1.1
```

VPNSM

```
ip route 10.1.1.1 255.255.255.255 vlan0.1
```

Configuring RRI When Two Routes Are Created, One for the Remote Endpoint and One for Route Recursion: Example

In the following example, two routes are created, one for the remote endpoint and one for route recursion to the remote endpoint via the interface on which the crypto map is configured.

```
reverse-route remote-peer
```

Configuring RRI with the Enhancements Added in Cisco IOS Release 12.3(14)T: Examples

The following are examples of configurations and output for the RRI enhancements that were added in Cisco IOS Release 12.3(14)T.

- Configuring RRI When Crypto ACLs Exist: Example, page 1306
- Configuring RRI with Route Tags: Example, page 1306
- Configuring RRI for One Route to the Remote Proxy Via a User-Defined Next Hop: Example, page 1306

Configuring RRI When Crypto ACLs Exist: Example

The following example shows that RRI has been configured for a situation in which there are existing ACLs.

```
crypto map mymap 1 ipsec-isakmp
   set peer 172.17.11.1
   reverse-route static
   set transform-set esp-3des-sha
   match address 101

access-list 101 permit ip 192.168.1.0 0.0.0.255 172.17.11.0 0.0.0.255
```

Configuring RRI with Route Tags: Example

The following example shows how RRI-created routes can be tagged with a tag number and then used by a routing process to redistribute those tagged routes via a route map.

```
crypto dynamic-map ospf-clients 1
 reverse-route tag 5

router ospf 109
 redistribute rip route-map rip-to-ospf

route-map rip-to-ospf permit
 match tag 5
 set metric 5
 set metric-type type1
```

`show ip ospf topology`

```
P 10.81.7.48/29, 1 successors, FD is 2588160, tag is 5
     via 192.168.82.25 (2588160/2585600), FastEthernet0/1
```

Configuring RRI for One Route to the Remote Proxy Via a User-Defined Next Hop: Example

The following example shows that one route has been created to the remote proxy via a user-defined next hop. This next hop should not require a recursive route lookup unless it will recurse to a default route.

```
reverse-route remote-peer 10.4.4.4
```

The above example yields the following prior to Cisco IOS Release 12.3(14)T:

```
10.0.0.0/24 via 10.1.1.1 (in the VRF table if VRFs are configured)
10.1.1.1/32 via 10.4.4.4 (in the global route table)
```

And this result occurs with RRI enhancements:

```
10.0.0.0/24 via 10.4.4.4 (in the VRF table if VRFs are configured, otherwise in the global
table)
```

Additional References

The following sections provide references related to Reverse Route Injection enhancements.

Related Documents

Related Topic	Document Title
Cisco IOS Security commands	*Cisco IOS Security Command Reference*, Release 12.3T
Other Cisco IOS commands	*Cisco IOS Command Reference*, Release 12.3T

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following modified command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **reverse-route**

SafeNet IPSec VPN Client Support

The SafeNet IPSec VPN Client Support feature allows you to limit the scope of an Internet Security Association and Key Management Protocol (ISAKMP) profile or ISAKMP keyring configuration to a local termination address or interface. The benefit of this feature is that different customers can use the same peer identities and ISAKMP keys by using different local termination addresses.

History for the SafeNet IPSec VPN Client Support Feature

Release	Modification
12.3(14)T	This feature was introduced.
12.2(18)SXE	This feature was integrated into Cisco IOS Release 12.2(18)SXE.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for SafeNet IPSec VPN Client Support, page 1309
- Restrictions for SafeNet IPSec VPN Client Support, page 1310
- Information About SafeNet IPSec VPN Client Support, page 1310
- How to Configure SafeNet IPSec VPN Client Support, page 1311
- Configuration Examples for SafeNet IPSec VPN Client Support, page 1315
- Additional References, page 1316
- Command Reference, page 1317

Prerequisites for SafeNet IPSec VPN Client Support

- You must understand how to configure ISAKMP profiles and ISAKMP keyrings.

Restrictions for SafeNet IPSec VPN Client Support

- The local address option works only for the primary address of an interface.
- If an IP address is provided, the administrator has to ensure that the connection of the peer terminates to the address that is provided.
- If the IP address does not exist on the device, or if the interface does not have an IP address, the ISAKMP profile or ISAKMP keyring will be effectively disabled.

Information About SafeNet IPSec VPN Client Support

Before configuring SafeNet IPSec VPN Client Support, you should understand the following concepts:

- ISAKMP Profile and ISAKMP Keyring Configurations: Background, page 1310
- Local Termination Address or Interface, page 1310

ISAKMP Profile and ISAKMP Keyring Configurations: Background

Prior to Cisco IOS Release 12.3(14)T, ISAKMP-profile and ISAKMP-keyring configurations could be only global, meaning that the scope of these configurations could not be limited by any locally defined parameters (VRF instances were an exception). For example, if an ISAKMP keyring contained a preshared key for address 10.11.12.13, the same key would be used if the peer had the address 10.11.12.13, irrespective of the interface or local address to which the peer was connected. There are situations, however, in which users prefer that associate keyrings be bound not only with virtual route forwarding (VRF) instances but also to a particular interface. For example, if instead of VRF instances, there are virtual LANS, and the Internet Key Exchange (IKE) is negotiated with a group of peers using one fixed virtual LAN (VLAN) interface. Such a group of peers uses a single preshared key, so if keyrings could be bound to an interface, it would be easy to define a wildcard key without risking that the keys would also be used for other customers.

Sometimes the identities of the peer are not in the control of the administrator, and even if the same peer negotiates for different customers, the local termination address is the only way to distinguish the peer. After such a distinction is made, if the traffic is sent to different VRF instances, configuring an ISAKMP profile is the only way to distinguish the peer. Unfortunately, when the peer uses an identical identity for all such situations, the ISAKMP profile cannot distinguish among the negotiations. For such scenarios, it would be beneficial to bind ISAKMP profiles to a local termination address. If a local termination address could be assigned, identical identities from the peer would not be a problem.

Local Termination Address or Interface

Effective with Cisco IOS Release 12.3(14)T, the SafeNet IPSec VPN Client Support feature allows you to limit the scope of ISAKMP profiles and ISAKMP keyrings to a local termination address or interface.

Benefit of SafeNet IPSec VPN Client Support

The benefit of this feature is that different customers can use the same peer identities and ISAKMP keys by using different local termination addresses.

How to Configure SafeNet IPSec VPN Client Support

This section contains the following procedures. The first two configurations are independent of each other.

- Limiting an ISAKMP Profile to a Local Termination Address or Interface, page 1311 (required)
- Limiting a Keyring to a Local Termination Address or Interface, page 1312 (required)
- Monitoring and Maintaining SafeNet IPSec VPN Client Support, page 1313 (optional)
- Examples, page 1314 (optional)

Limiting an ISAKMP Profile to a Local Termination Address or Interface

To configure an ISAKMP profile and limit it to a local termination address or interface, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp profile** *profile-name*
4. **keyring** *keyring-name*
5. **match identity address** *address*
6. **local-address** {*interface-name* | *ip-address* [*vrf-tag*]}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp profile** *profile-name* **Example:** `Router (config)# crypto isakmp profile profile1`	Defines an ISAKMP profile and enters ISAKMP profile configuration mode.
Step 4	**keyring** *keyring-name* **Example:** `Router (conf-isa-profile)# keyring keyring1`	(Optional) Configures a keyring with an ISAKMP profile. • A keyring is not needed inside an ISAKMP profile for local termination to work. Local termination works even if Rivest, Shamir, and Adelman (RSA) certificates are used.

	Command or Action	Purpose
Step 5	**match identity address** *address* **Example:** `Router (conf-isa-profile)# match identity address 10.0.0.0 255.0.0.0`	Matches an identity from a peer in an ISAKMP profile.
Step 6	**local-address** {*interface-name* | *ip-address* [*vrf-tag*]} **Example:** `Router (conf-isa-profile)# local-address serial2/0`	Limits the scope of an ISAKMP profile or an ISAKMP keyring configuration to a local termination address or interface.

Limiting a Keyring to a Local Termination Address or Interface

To configure an ISAKMP keyring and limit its scope to a local termination address or interface, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto keyring** *keyring-name*
4. **local-address** {*interface-name* | *ip-address* [*vrf-tag*]}
5. **pre-shared-key address** *address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto keyring** *keyring-name* **Example:** `Router (config)# crypto keyring keyring1`	Defines a crypto keyring to be used during IKE authentication and enters keyring configuration mode.

	Command or Action	Purpose
Step 4	**local-address** {*interface-name* | *ip-address* [*vrf-tag*]} **Example:** `Router (conf-keyring)# local-address serial2/0`	Limits the scope of an ISAKMP profile or an ISAKMP keyring configuration to a local termination address or interface.
Step 5	**pre-shared-key address** *address* **Example:** `Router (conf-keyring)# pre-shared-key address 10.0.0.1`	Defines a preshared key to be used for IKE authentication.

Monitoring and Maintaining SafeNet IPSec VPN Client Support

The following **debug** and **show** commands may be used to monitor and maintain the configuration in which you limited the scope of an ISAKMP profile or ISAKMP keyring to a local termination address or interface.

SUMMARY STEPS

1. **enable**
2. **debug crypto isakmp**
3. **show crypto isakmp profile**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto isakmp** **Example:** `Router# debug crypto isakmp`	Displays messages about IKE events.
Step 3	**show crypto isakmp profile** **Example:** `Router# show crypto isakmp profile`	Lists all the ISAKMP profiles that are defined on a router.

Examples

debug crypto isakmp Command Output for an ISAKMP Keyring That Is Bound to Local Termination Addresses: Example

You have an ISAKMP configuration as follows (the address of serial2/0 is 10.0.0.1, and the address of serial2/1 is 10.0.0.2),

```
crypto keyring keyring1
! Scope of the keyring is limited to interface serial2/0.
  local-address serial2/0
  ! The following is the key string used by the peer.
  pre-shared-key address 10.0.0.3 key somerandomkeystring
crypto keyring keyring2
  local-address serial2/1
  ! The following is the keystring used by the peer coming into serial2/1.
   pre-shared-key address 10.0.0.3 key someotherkeystring
```

and if the connection is coming into serial2/0, keyring1 is chosen as the source of the preshared key (and keyring2 is ignored because it is bound to serial2/1), you would see the following output:

```
Router# debug crypto isakmp

*Feb 11 15:01:29.595: ISAKMP:(0:0:N/A:0):Keyring keyring2 is bound to
  10.0.0.0, skipping
*Feb 11 15:01:29.595: ISAKMP:(0:0:N/A:0):Looking for a matching key for
  10.0.0.3 in keyring1
*Feb 11 15:01:29.595: ISAKMP:(0:0:N/A:0): : success
*Feb 11 15:01:29.595: ISAKMP:(0:0:N/A:0):found peer pre-shared key
  matching 10.0.0.3
*Feb 11 15:01:29.595: ISAKMP:(0:0:N/A:0): local preshared key found
```

debug crypto isakmp Command Output for an ISAKMP ProfileThat Is Bound to a Local Termination Address: Example

If you have the following configuration,

```
crypto isakmp profile profile1
  keyring keyring1
  match identity address 10.0.0.0 255.0.0.0
  local-address serial2/0
crypto isakmp profile profile2
  keyring keyring1
  keyring keyring2
  self-identity fqdn
  match identity address 10.0.0.1 255.255.255.255
  local-address serial2/1
```

and the connection is coming through the local terminal address serial2/0, you will see the following output:

```
Router# debug crypto isakmp

*Feb 11 15:01:29.935: ISAKMP:(0:0:N/A:0):

Profile profile2 bound to 10.0.0.0 skipped

*Feb 11 15:01:29.935: ISAKMP:(0:1:SW:1):: peer matches profile1 profile
```

show crypto isakmp profile Command Output: Example

The following is an example of typical **show** command output for an ISAKMP profile that is bound to serial2/0:

```
Router# show crypto isakmp profile

ISAKMP PROFILE profile1
  Identities matched are:
    ip-address 10.0.0.0 255.0.0.0
  Certificate maps matched are:
  keyring(s): keyring1
  trustpoint(s): <all>
  Interface binding: serial2/0 (10.20.0.1:global)
```

Troubleshooting SafeNet IPSec VPN Client Support

If an ISAKMP profile or ISAKMP keyring fails to be selected, you should double-check the local-address binding in the ISAKMP profile or ISAKMP keyring configuration and follow the output of the IKE debugs to determine whether the peer is correctly terminating on the address. You may remove the local-address binding (to make the scope of the profile or keyring global) and check to determine whether the profile or keyring is selected to confirm the situation.

Configuration Examples for SafeNet IPSec VPN Client Support

This section contains the following configuration, **debug** command, and **show** command examples.

- ISAKMP Profile Bound to a Local Interface: Example, page 1315
- ISAKMP Keyring Bound to a Local Interface: Example, page 1315
- ISAKMP Keyring Bound to a Local IP Address: Example, page 1316
- ISAKMP Keyring Bound to an IP Address and Limited to a VRF: Example, page 1316

ISAKMP Profile Bound to a Local Interface: Example

The following example shows that the ISAKMP profile is bound to a local interface:

```
crypto isakmp profile profile1
  keyring keyring1
  match identity address 10.0.0.0 255.0.0.0
  local-address serial2/0
```

ISAKMP Keyring Bound to a Local Interface: Example

The following example shows that the ISAKMP keyring is bound only to interface serial2/0:

```
crypto keyring
  local-address serial2/0
  pre-shared-key address 10.0.0.1
```

ISAKMP Keyring Bound to a Local IP Address: Example

The following example shows that the ISAKMP keyring is bound only to IP address 10.0.0.2:

```
crypto keyring keyring1
  local-address 10.0.0.2
  pre-shared-key address 10.0.0.2 key
```

ISAKMP Keyring Bound to an IP Address and Limited to a VRF: Example

The following example shows that an ISAKMP keyring is bound to IP address 10.34.35.36 and that the scope is limited to VRF examplevrf1:

```
ip vrf examplevrf1
  rd 12:3456
crypto keyring ring1
  local-address 10.34.35.36 examplevrf1
interface ethernet2/0
  ip vrf forwarding examplevrf1
  ip address 10.34.35.36 255.255.0.0
```

Additional References

The following sections provide references related to SafeNet IPSec VPN Client Support.

Related Documents

Related Topic	Document Title
Configuring ISAKMP profiles and ISAKMP keyrings	*VRF-Aware IPSec*
Security commands	*Cisco IOS Security Command Reference*, Release 12.3T

Standards

Standard	Title
No new or modified standards are supported by this feature.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature.	—

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Command Reference

The following new command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **local-address**

Stateful Failover for IPSec

Stateful failover for IP Security (IPSec) enables a router to continue processing and forwarding IPSec packets after a planned or unplanned outage occurs. Customers employ a backup (secondary) router that automatically takes over the tasks of the active (primary) router if the active router loses connectivity for any reason. This process is transparent to the user and does not require adjustment or reconfiguration of any remote peer.

Stateful failover for IPSec is designed to work in conjunction with stateful switchover (SSO) and Hot Standby Routing Protocol (HSRP). HSRP provides network redundancy for IP networks, ensuring that user traffic immediately and transparently recovers from failures in network edge devices or access circuits. That is, HSRP monitors both the inside and outside interfaces so that if either interface goes down, the whole router is deemed to be down and ownership of Internet Key Exchange (IKE) and IPSec security associations (SAs) is passed to the standby router (which transitions to the HSRP active state). SSO allows the active and standby routers to share IKE and IPSec state information so that each router has enough information to become the active router at any time. To configure stateful failover for IPSec, a network administrator should enable HSRP, assign a virtual IP address, and enable the SSO protocol.

Feature History for Stateful Failover for IPSec

Release	Modification
12.3(11)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for Stateful Failover for IPSec, page 2
- Restrictions for Stateful Failover for IPSec, page 2
- Information About Stateful Failover for IPSec, page 3
- How to Use Stateful Failover for IPSec, page 6
- Configuration Examples for Stateful Failover, page 27
- Additional References, page 36

- Command Reference, page 37

Prerequisites for Stateful Failover for IPSec

Complete, Duplicate IPSec and IKE Configuration on the Active and Standby Devices

This document assumes that you have a complete IKE and IPSec configuration. (This document describes only how to add stateful failover to a working IPSec configuration.)

The IKE and IPSec configuration that is set up on the active device must be duplicated on the standby device. That is, the crypto configuration must be identical with respect to Internet Security Association and Key Management Protocol (ISAKMP) policy, ISAKMP keys (preshared), IPSec profiles, IPSec transform sets, all crypto map sets that are used for stateful failover, all access control lists (ACLs) that are used in match address statements on the crypto map sets, all AAA configurations used for crypto, client configuration groups, ip local pools used for crypto, and ISAKMP profiles.

Note None of the configuration information between the active and standby device is automatically transferred; the user is responsible for ensuring that the crypto configurations match on both devices. If the crypto configurations on both devices do not match, failover from the active device to the standby device will not be successful.

Device Requirements

- Stateful failover for IPSec requires that your network contains two identical routers that are available to be either the primary or secondary device. Both routers should be the same type of device, have the same CPU and memory, and have either no encryption accelerator or identical encryption accelerators.
- This feature is currently supported only on a limited number of platforms. To check the latest platform support, go to Cisco Feature Navigator at http://www.cisco.com/go/fn.

Restrictions for Stateful Failover for IPSec

When configuring redundancy for a virtual private network (VPN), the following restrictions exist:

- Both the active and standby devices must run the identical version of the Cisco IOS software, and both the active and standby devices must be connected via hub or switch.
- Only the VPN Acceleration Module (VAM), VAM2, and AIM-VPN/HPII hardware encryption accelerators are supported.
- Only “box-to-box” failover is supported; that is, intrachassis failover is currently not supported.
- WAN interfaces between the active (primary) router and the standby (secondary) router are not supported. (HSRP requires inside interfaces and outside interfaces to be connected via LANs.)
- Load balancing is not supported; that is, no more than one device in a redundancy group can be active at any given time.
- Stateful failover of IPSec with Layer 2 Tunneling Protocol (L2TP) is not supported.
- Public key infrastructure (PKI) is not supported when used with stateful failover. (Only preshared keys for IKE are supported.)

- IKE keepalives are not supported. (Enabling this functionality will cause the connection to be torn down after the standby router assumes ownership control.) However, dead peer detection (DPD) and periodic DPD are supported.
- IPSec idle timers are not supported when used with stateful failover.
- A stateful failover crypto map applied to an interface in a virtual route forwarding (VRF) instance is not supported. However, VRF-aware IPSec features are supported when a stateful failover crypto map is applied to an interface in the global VRF.
- Stateful failover is not compatible or interoperable with the State Synchronization Protocol (SSP) version of stateful failover (which is available in Cisco IOS Release 12.2YX1 and Cisco IOS Release 12.2SU).

Information About Stateful Failover for IPSec

To configure stateful failover for VPNs, you should understand the following concepts:

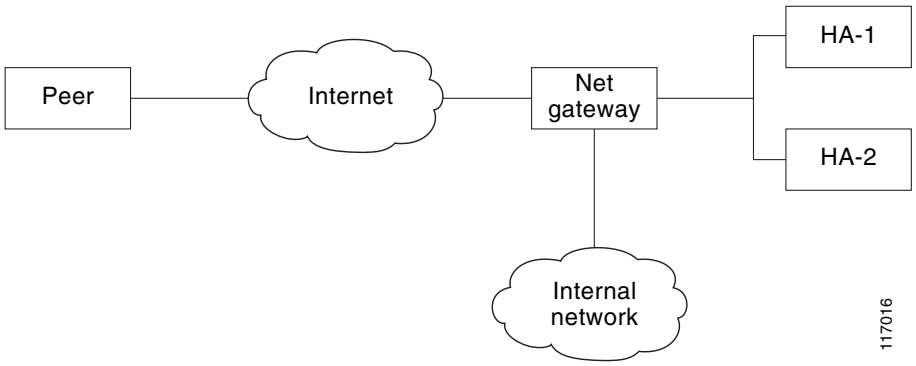
- Supported Deployment Scenarios: Stateful Failover for IPSec, page 3
- IPSec Stateful Failover for Remote Access Connections, page 5

Supported Deployment Scenarios: Stateful Failover for IPSec

It is recommended that you implement IPSec stateful failover in one of the following recommended deployment scenarios—a single interface scenario or a dual interface scenario.

In a single interface scenario, the VPN gateways use one LAN connection for both encrypted traffic arriving from remote peers and decrypted traffic flowing to inside hosts (see Figure 1). The single interface design allows customers to save money on router ports and subnets. This design is typically used if all traffic flowing in and out of the organization does not traverse the VPN routers.

Figure 93 Single Interface Network Topology



In a dual interface scenario, a VPN gateway has more than one interface, enabling traffic to flow in and out of the router via separate interfaces (see Figure 2). This scenario is typically used if traffic flowing in and out of a site must traverse the routers, so the VPN routers will provide the default route out of the network.

Figure 94 Dual Interface Network Topology

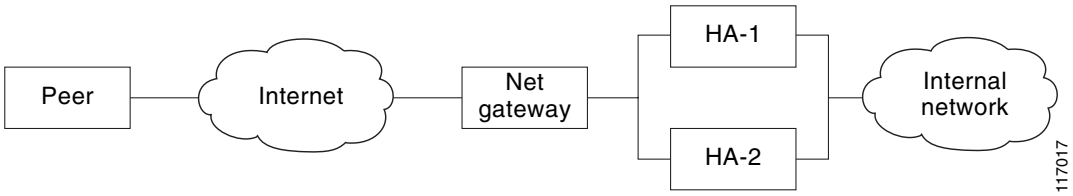


Table 1 lists the functionality available in both a single interface scenario and a dual interfaces scenario.

Table 55 IPSec StateFul Failover: Single and Dual Interface Functionality Overview

Single Interface	Dual Interface
Route Injection	
Routes must be injected to provide the devices that are behind the VPN gateways with a next hop for traffic that requires encryption. Stateful failover for IPSec typically requires routes to be injected for this network topology.	If the VPN gateways are not the logical next hop for devices inside the network, the routes must be created and injected into the routing process. Thus, traffic that is returning from inside the network can be sent back to the VPN routers for IPSec services before it is sent out. A virtual IP (VIP) address cannot be used as the advertiser of routing updates, so flows must be synchronized via the injected routes. If the VPN gateways are the next hop (default route) for all devices inside the network, the VIP address that is used on the inside interfaces can be used as the next hop. Thus, injection of the VPN routes is not required. However, static routes on inside hosts must be used to direct the routes to the next hop VIP address.
HSRP Configuration	
The role of HSRP is simplified in a single interface design because if the only interface is disabled, the entire device is deemed unavailable. This functionality helps to avoid some of the routing considerations to be discussed in the next scenario.	Because each interface pair functions independently, you should configure HSRP so that multiple pairs of interfaces can be tracked. (That is, HSRP should not be configured on only one pair of interfaces or on both pairs of interfaces without each pair mutually tracking each other.) Mutual tracking means that if the outside interface does fail, the inside interface on the same router will also be deemed down, allowing for complete router failover to the secondary router.
Secure State Information	
If secured-state information is passed between routers, the information is passed over the same interface as all other traffic.	The router has a separate inside and outside interface; thus, the inside interface can be used as a more secure channel for the exchange of state information.

Table 55 IPSec StateFul Failover: Single and Dual Interface Functionality Overview (continued)

Single Interface	Dual Interface
Firewall Configuration	
The VPN gateways can sit in front of the firewall or behind the firewall.	VPN gateways may sit behind or in front of a firewall, a firewall can be installed in parallel to the VPN gateways.

IPSec Stateful Failover for Remote Access Connections

The main difference between a remote access and a LAN-to-LAN connection is the use of Xauth and mode-config. IKE Xauth is often used to authenticate the user. IKE mode-config is often used to push security policy from the hub (concentrator) router to the user's IPSec implementation. Mode-config is also typically used to assign an internal company network IP address to a user.

In addition to the differences between a remote access configuration and a LAN-to-LAN configuration, you should note the following remote-access-server-specific functions:

- Assigned IP address—The IP address can be assigned to the client via one of the following options:
  - Local IP pools. For local IP pools, the administrator must first configure identical local IP address pools on each router in the high availability (HA) pair (via the **ip local pool** *client-address-pool* command). This pool name can be applied in one of two places—in a group policy via the **crypto isakmp client configuration group** *group-name* (and the submode command **pool** *pool-name*) or in a client configuration via the **crypto isakmp client configuration address-pool local** *local-pool* command.
  - RADIUS-assigned address. If you are using RADIUS authentication and the RADIUS server returns the Framed-IP-Address attribute, the concentrator will always assign that address to the client. It is recommended that you refer to your RADIUS server vendor's documentation, especially for vendors that allow you to configure address pools on the RADIUS server. Typically those servers require crypto accounting to work properly.

    To enable accounting on the HA pair, you should issue the following commands on both Active and Standby devices: **aaa accounting network radius-accounting start-stop group radius** then apply radius-accounting either to the crypto isakmp profile or the crypto map set.
- RADIUS NAS-IP address—The HA pair should appear as a single device to the RADIUS server. Thus, both HA routers must communicate with the RADIUS server using the same IP address. However, when communicating with the RADIUS server, the router must use a physical IP address, not a virtual IP (VIP) address as the NAS-IP address of the router. To configure the RADIUS NAS-IP address for the HA pair, you must configure the same loopback address in the HA pair via **interface loopback ip address** command; thereafter, you must issue the **ip radius source-interface loopback** command in the HA pair. Finally, add the new loopback IP address to the RADIUS servers configuration so the RADIUS server can process requests from the HA pair.

For additional information on how to configure IPSec stateful failover for a remote access connection, see the section "Configuring IPSec Stateful Failover for an Easy VPN Server: Example" in this document.

How to Use Stateful Failover for IPSec

This section contains the following the procedures:

- Enabling HSRP: IP Redundancy and a Virtual IP Address, page 6 (required)
- Enabling SSO, page 9 (required)
- Configuring Reverse Route Injection on a Crypto Map, page 13 (required)
- Enabling Stateful Failover for IKE and IPSec, page 15 (required)
- Protecting SSO Traffic, page 18 (optional)
- Managing and Verifying High Availability Information, page 20 (optional)

Enabling HSRP: IP Redundancy and a Virtual IP Address

HSRP provides two services—IP redundancy and a VIP address. Each HSRP group may provide either or both of these services. IPSec stateful failover uses the IP redundancy services from only one HSRP standby group. It can use the VIP address from one or more HSRP groups. Use the following task to configure HSRP on the outside and inside interfaces of the router.

Note Perform this task on both routers (active and standby) and of both interfaces on each router.

Prerequisites for Spanning Tree Protocol and HSRP Stability

If a switch connects the active and standby routers, you must perform one of the following steps to ensure that the correct settings are configured on that switch:

- Enable the **spanning-tree portfast** command on every switch port that connects to a HSRP-enabled router interface.
- Disable the Spanning Tree Protocol (STP) on the switch only if your switch does not connect to other switches. Disabling spanning tree in a multi-switch environment may cause network instability.
- Enable the **standby delay minimum** [*min-delay*] **reload** [*reload-delay*] command if you do not have access to the switch. The *reload-delay* argument should be set to a value of at least 120 seconds. This command must be applied to all HSRP interfaces on both routers.

For more information on HSRP instability, see the document *Avoiding HSRP Instability in a Switching Environment with Various Router Platforms*.

Note You must perform at least one of these steps for correct HSRP operation.

Restrictions

- Both the inside (private) interface and the outside (public) interface must belong to separate HSRP groups, but the HSRP group number can be the same.
- The state of the inside interface and the outside interface must be the same—both interfaces must be in the active state or standby state; otherwise, the packets will not have a route out of the private network.

- Standby priorities should be equal on both active and standby routers. If the priorities are not equal, the higher priority router will unnecessarily take over as the active router, negatively affecting uptime.
- The IP addresses on the HSRP-tracked interfaces of the standby and active routers should both be either lower or higher on one router than the other. In the case of equal priorities (an HA requirement), HSRP will assign the active state on the basis of the IP address. If an addressing scheme exists so that the public IP address of Router A is lower than the public IP address of Router B, but the opposite is true for their private interfaces, an active/standby-standby/active split condition could exist which will break connectivity.

Note Each time an active device relinquishes control to become the standby device, the active device will reload. This functionality ensures that the state of the new standby device synchronizes correctly with the new active device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **standby** *standby-group-number* **name** *standby-group-name*
5. **standby** *standby-group-number* **ip** *ip-address*
6. **standby** *standby-group-number* **track** *interface-name*
7. **standby** [*group-number*] **preempt**
8. **standby** [*group-number*] **timers** [**msec**] *hellotime* [**msec**] *holdtime*
9. **standby delay minimum** [*min-delay*] **reload** [*reload-delay*]
10. Repeat.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet 0/0`	Configures an interface type for the router and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**standby** *standby-group-number* **name** *standby-group-name* **Example:** `Router(config-if)# standby 1 name HA-out`	Assigns a user-defined group name to the HSRP redundancy group. **Note** The *standby-group-number* argument should be the same for both routers that are on directly connected interfaces. However, the *standby-group-name* argument should be different between two (or more) groups on the same router. The *standby-group-number* argument can be the same on the other pair of interfaces as well.
Step 5	**standby** *standby-group-number* **ip** *ip-address* **Example:** `Router(config-if)# standby 1 ip 209.165.201.1`	Assigns an IP address that is to be "shared" among the members of the HSRP group and owned by the primary IP address. **Note** The virtual IP address must be configured identically on both routers (active and standby) that are on directly connected interfaces.
Step 6	**standby** *standby-group-number* **track** *interface-name* **Example:** `Router(config-if)# standby 1 track Ethernet1/0`	Configures HSRP to monitor the second interface so that if either of the two interfaces goes down, HSRP causes failover to the standby device. **Note** Although this command is not required, it is recommended for dual interface configurations.
Step 7	**standby** [*group-number*] **preempt** **Example:** `Router(config-if)# standby 1 preempt`	Enables the active device to relinquish control because of an interface tracking event.
Step 8	**standby** [*group-number*] **timers** [**msec**] *hellotime* [**msec**] *holdtime* **Example:** `Router(config-if)# standby 1 timers 1 5`	(Optional) Configures the time between hello packets and the time before other routers declare the active Hot Standby or standby router to be down. • *holdtime*—Amount of time the routers take to detect types of failure. A larger hold time means that failure detection will take longer. For the best stability, it is recommended that you set the hold time between 5 and 10 times the hello interval time; otherwise, a failover could falsely occur when no actual failure has happened.
Step 9	**standby delay minimum** [*min-delay*] **reload** [*reload-delay*] **Example:** `Router(config-if)# standby delay minimum reload 120`	Configures the delay period before the initialization of HSRP groups. **Note** It is suggested that you enter 120 as the value for the *reload-delay* argument and leave the *min-delay* argument at the preconfigured default value.
Step 10	`Repeat.`	Repeat this task on both routers (active and standby) and on both interfaces of each router.

Troubleshooting Tips

To help troubleshoot possible HSRP-related configuration problems, issue any of the following HSRP-related debug commands—**debug standby errors**, **debug standby events**, and **debug standby packets** [**terse**].

Examples

The following example shows how to configure HSRP on a router:

```
interface Ethernet0/0
 ip address 209.165.201.1 255.255.255.224
 standby 1 ip 209.165.201.3
 standby 1 preempt
 standby 1 name HA-out
 standby 1 track Ethernet1/0
 standby delay reload 120
```

What to Do Next

After you have successfully configured HSRP on both the inside and outside interfaces, you should enable SSO as described the in the section "Enabling SSO."

Enabling SSO

Use this task to enable SSO, which is used to transfer IKE and IPSec state information between two routers.

SSO: Interacting with IPSec and IKE

SSO is a method of providing redundancy and synchronization for many Cisco IOS applications and features. SSO is necessary for IPSec and IKE to learn about the redundancy state of the network and to synchronize their internal application state with their redundant peers.

Prerequisites

- You should configure HSRP before enabling SSO.
- To avoid losing SCTP communication between peers, be sure to include the following commands to the local address section of the SCTP section of the IPC configuration:
  - **retransmit-timeout** *retran-min* [*msec*] *retra-max* [*msec*]
  - **path-retransmit** *max-path-retries*
  - **association-retransmit** *max-association-retries*

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy inter-device**

4. **scheme standby** *standby-group-name*
5. **exit**
6. **ipc zone default**
7. **association 1**
8. **protocol sctp**
9. **local-port** *local-port-number*
10. **local-ip** *device-real-ip-address* [*device-real-ip-address2*]
11. **retransmit-timeout** *retran-min* [*msec*] *retra-max* [*msec*]
12. **path-retransmit** *max-path-retries*
13. **association-retransmit** *max-association-retries*
14. **exit**
15. **remote-port** *remote-port-number*
16. **remote-ip** *peer-real-ip-address* [*peer-real-ip-address2*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**redundancy inter-device** **Example:** `Router(config)# redundancy inter-device`	Configures redundancy and enters inter-device configuration mode. To exit inter-device configuration mode, use the **exit** command. To remove all inter-device configuration, use the **no** form of the command.
Step 4	**scheme standby** *standby-group-name* **Example:** `Router(config-red-interdevice)# scheme standby HA-in`	Defines the redundancy scheme that is to be used. Currently, "standby" is the only supported scheme. • *standby-group-name*—Must match the standby name specified in the **standby name** interface configuration command. Also, the standby name should be the same on both routers. **Note** Only the active or standby state of the standby group is used for SSO. The VIP address of the standby group is not required or used by SSO. Also, the standby group does not have to be part of any crypto map configuration.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(config-red-interdevice)# exit`	Exits inter-device configuration mode.
Step 6	**ipc zone default** **Example:** `Router(config)# ipc zone default`	Configures the inter-device communication protocol, Inter-Process Communication (IPC), and enters IPC zone configuration mode. Use this command to initiate the communication link between the active router and standby router.
Step 7	**association 1** **Example:** `Router(config-ipczone)# association 1`	Configures an association between the two devices and enters IPC association configuration mode.
Step 8	**protocol sctp** **Example:** `Router(config-ipczone-assoc)# protocol sctp`	Configures Stream Control Transmission Protocol (SCTP) as the transport protocol and enters SCTP protocol configuration mode.
Step 9	**local-port** *local-port-number* **Example:** `Router(config-ipc-protocol-sctp)# local-port 5000`	Defines the local SCTP port number that is used to communicate with the redundant peer and puts you in IPC transport - SCTP local configuration mode. • *local-port-number*—There is not a default value. This argument must be configured for the local port to enable inter-device redundancy. Valid port values: 1 to 65535. The local port number should be the same as the remote port number on the peer router.
Step 10	**local-ip** *device-real-ip-address* [*device-real-ip-address2*] **Example:** `Router(config-ipc-local-sctp)# local-ip 10.0.0.1`	Defines at least one local IP address that is used to communicate with the redundant peer. The local IP addresses must match the remote IP addresses on the peer router. There can be either one or two IP addresses, which must be in the global VRF. A virtual IP address cannot be used.
Step 11	**retransmit-timeout** *retran-min* [*msec*] *retra-max* [*msec*] **Example:** `Router(config-ipc-local-sctp)# retransmit-timeout 300 10000`	Configures the maximum amount of time, in milliseconds, that SCTP will wait before retransmitting data. • *retran-min*: 300 to 60000; default: 300 • *retran-max*: 300 to 60000; default: 600
Step 12	**path-retransmit** *max-path-retries* **Example:** `Router(config-ipc-local-sctp)#  path-retransmit 10`	Configures the number of consecutive retransmissions SCTP will perform before failing a path within an association. • *max-path-retries*: 2 to 10; default: 4 retries

	Command or Action	Purpose
Step 13	**association-retransmit** *max-association-retries* **Example:** `Router(config-ipc-local-sctp)# association-retransmit 20`	Configures the number of consecutive retransmissions SCTP will perform before failing an association. • *max-association-retries*: 2 to 10; defaul: 4 retries
Step 14	**exit** **Example:** `Router(config-ipc-local-sctp)# exit`	Exits IPC transport - SCTP local configuration mode.
Step 15	**remote-port** *remote-port-number* **Example:** `Router(config-ipc-protocol-sctp)# remote-port 5000`	Defines the remote SCTP port number that is used to communicate with the redundant peer and puts you in IPC transport - SCTP remote configuration mode. **Note** *remote-port-number*—There is not a default value. This argument must be configured for the remote port to enable inter-device redundancy. Valid port values: 1 to 65535. The remote port number should be the same as the local port number on the peer router.
Step 16	**remote-ip** *peer-real-ip-address* [*peer-real-ip-address2*] **Example:** `Router(config-ipc-remote-sctp)# remote-ip 10.0.0.2`	Defines at least one remote IP address of the redundant peer that is used to communicate with the local device. All remote IP addresses must refer to the same device. A virtual IP address cannot be used.

Troubleshooting Tips

To help troubleshoot possible SSO-related configuration problems, issue the **debug redundancy** command.

Examples

The following example shows how to enable SSO:

```
!
redundancy inter-device
 scheme standby HA-in
!
!
ipc zone default
 association 1
  no shutdown
  protocol sctp
   local-port 5000
    local-ip 10.0.0.1
    retransmit-timeout 300 10000
    path-retransmit 10
    assoc-retransmit 20
   remote-port 5000
    remote-ip 10.0.0.2
!
```

What to Do Next

After you have enabled SSO, you should configure reverse route injection (RRI) on a crypto map as shown in the following section.

Configuring Reverse Route Injection on a Crypto Map

You should configure RRI on all existing crypto maps that you want to use with stateful failover. RRI is used with stateful failover so routers on the inside network can learn about the correct path to the current active device. When failover occurs, the new active device injects the RRI routes into its IP routing table and sends out routing updates to its routing peers.

Use one of the following tasks to configure RRI on a dynamic or static crypto map.

- Configuring RRI on Dynamic Crypto Map, page 13
- Configuring RRI on a Static Crypto Map, page 14

Configuring RRI on Dynamic Crypto Map

Dynamic crypto map entries, like regular static crypto map entries, are grouped into sets. A set is a group of dynamic crypto map entries all with the same dynamic map name but each with a different dynamic sequence number. Each member of the set may be configured for RRI.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto dynamic-map** *map-name seq-num*
4. **reverse-route**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto dynamic-map** *map-name seq-num* **Example:** `Router(config)# crypto dynamic-map mymap 10`	Creates a dynamic crypto map entry and enters crypto map configuration mode.
Step 4	**reverse-route** **Example:** `Router(config-crypto-map)# reverse-route`	Enables RRI for a dynamic crypto map.

Configuring RRI on a Static Crypto Map

Static crypto map entries are grouped into sets. A set is a group of static crypto map entries all with the same static map name but each with a different sequence number. Each static crypto map in the map set can be configured for RRI. Use this task to configure RRI on a static crypto map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* **ipsec-isakmp**
4. **reverse-route**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* **ipsec-isakmp** **Example:** `Router(config)# crypto map to-peer-outside 10 ipsec-isakmp`	Enters crypto map configuration mode and creates or modifies a crypto map entry.
Step 4	**reverse-route** **Example:** `Router(config-crypto-map)# reverse-route`	Dynamically creates static routes based on crypto ACLs.

Examples

The following example shows how to configure RRI on the static crypto map "to-peer-outside":

```
crypto map to-peer-outside redundancy replay-interval inbound 1000 outbound 10000
crypto map to-peer-outside 10 ipsec-isakmp
 set peer 209.165.200.225
 set transform-set trans1
 match address peer-outside
 reverse-route
```

What to Do Next

After you have configured RRI, you can enable stateful failover for IPSec and IKE.

Enabling Stateful Failover for IKE and IPSec

Use the following tasks to configure stateful failover for IPSec, IKE, and tunnel protection:

- Enabling Stateful Failover for IKE, page 15
- Enabling Stateful Failover for IPSec, page 15
- Enabling Stateful Failover for Tunnel Protection, page 17

Enabling Stateful Failover for IKE

There is no specific command-line interface (CLI) necessary to enable stateful failover for IKE. It is enabled for a particular VIP address when a stateful failover crypto map is applied to an interface.

Enabling Stateful Failover for IPSec

Use this task to enable stateful failover for IPSec. All IPSec state information is transferred from the active router to the standby router via the SSO redundancy channel that was specified in the task "Enabling SSO."

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **crypto map** *map-name* [**redundancy** *standby-group-name* [**stateful**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet 0/0`	Defines an interface that has already been configured for redundancy and enters interface configuration mode.
Step 4	**crypto map** *map-name* [**redundancy** *standby-group-name* [**stateful**]] **Example:** `Router(config-if)# crypto map to-peer-outside redundancy HA-out stateful`	Binds the crypto map on the specified interface to the redundancy group. **Note** Although the standby group does not have to be the same group that was used when enabling SSO, it does have to be the same group that was used with the **standby ip** command on this interface. This crypto map will use the same VIP address for both IKE and IPSec to communicate with peers.

Troubleshooting Tips

To help troubleshoot possible IPSec HA-related problems, issue the **debug crypto ipsec ha** [**detail**] [**update**] command.

Examples

The following example shows how to configure IPSec stateful failover on the crypto map "to-peer-outside":

```
interface Ethernet0/0
 ip address 209.165.201.1 255.255.255.224
 standby 1 ip 209.165.201.3
 standby 1 preempt
 standby 1 name HA-out
 standby 1 track Ethernet1/0
 crypto map to-peer-outside redundancy HA-out stateful
```

Enabling Stateful Failover for Tunnel Protection

Use an existing IPSec profile to configure stateful failover for tunnels using IPSec. (You do not configure the tunnel interface as you would with a crypto map configuration.)

Restrictions

The tunnel source address must be a VIP address, and it must not be an interface name.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec profile** *name*
4. **redundancy** *standby-group-name* **stateful**
5. **exit**
6. **interface tunnel** *number*
7. **tunnel protection ipsec profile** *name*
8. **tunnel source** *virtual-ip-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec profile** *name* **Example:** `Router(config)# crypto ipsec profile peer-profile`	Defines the IPSec parameters that are to be used for IPSec encryption between two routers and enters crypto map configuration mode.
Step 4	**redundancy** *standby-group-name* **stateful** **Example:** `Router(config-crypto-map)# redundancy HA-out stateful`	Configures stateful failover for tunnels using IPSec.
Step 5	**exit** **Example:** `Router(config-crypto-map)# exit`	Exits crypto map configuration mode.

	Command or Action	Purpose
Step 6	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 5`	Configures a tunnel interface and enters interface configuration mode • *number*—Specifies the number of the interface that you want to create or configure. There is no limit on the number of tunnel interfaces you can create.
Step 7	**tunnel protection ipsec profile** *name* **Example:** `Router(config-if)# tunnel protection ipsec profile catprofile`	Associates a tunnel interface with an IPSec profile. *name*—Specifies the name of the IPSec profile; this value must match the name specified in the **crypto ipsec profile name** command.
Step 8	**tunnel source** *virtual-ip-address* **Example:** `Router(config-if)# tunnel source 10.1.1.1`	Sets source address for a tunnel interface. • *virtual-ip-address*—Must be a VIP address. **Note** Do not use the interface name as the tunnel source.

Examples

The following example shows how to configure stateful failover for tunnel protection:

```
crypto ipsec profile peer-profile
  redundancy HA-out stateful

interface Tunnel1
 ip unnumbered Loopback0
 tunnel source 209.165.201.3
 tunnel destination 10.0.0.5
 tunnel protection ipsec profile peer-profile
!
interface Ethernet0/0
 ip address 209.165.201.1 255.255.255.224
 standby 1 ip 209.165.201.3
 standby 1 name HA-out
```

What to Do Next

After you have configured stateful failover, you can use the CLI to protect, verify, and manage your configurations. For more information on completing these tasks, see the sections "Protecting SSO Traffic" and "Managing and Verifying High Availability Information."

Protecting SSO Traffic

Use this task to secure a redundancy group via an IPSec profile. To configure SSO traffic protection, the active and standby devices must be directly connected to each other via Ethernet networks.

The crypto maps that are automatically generated when protecting SSO traffic are applied to each interface, which corresponds to an IP address that was specified via the **local-ip** command. Traffic that is destined for an IP address that was specified via the **remote-ip** command is forced out of the crypto-map-configured interface via an automatically created static host route.

Note If you are certain that the SSO traffic between the redundancy group runs on a physically secure interface, you do not have to configure SSO traffic protection.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp key** *keystring* **address** *peer-address*
4. **crypto ipsec transform-set** *transform-set-name transform-set-list*
5. **crypto ipsec profile** *profile-name*
6. **set transform-set** *transform-set-name*
7. **exit**
8. **redundancy inter-device**
9. **security ipsec** *profile-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp key** *keystring* **address** *peer-address* **Example:** `Router(config)# crypto isakmp key cisco123 address 0.0.0.0 0.0.0.0`	Configures a preshared authentication key. • *peer-address*—The SCTP remote IP address.
Step 4	**crypto ipsec transform-set** *transform-set-name transform-set-list* **Example:** `Router(config)# crypto ipsec transform-set trans2 ah-md5-hmac esp-aes`	Configures a transform set that defines the packet format and cryptographic algorithms used for IPSec.
Step 5	**crypto ipsec profile** *profile-name* **Example:** `Router(config)# crypto ipsec profile sso-secure`	Defines an IPSec profile that describes how the traffic will be protected.

	Command or Action	Purpose
Step 6	**set transform-set** *transform-set-name* **Example:** `Router(config-crypto-map)# set transform-set trans2`	Specifies which transform sets can be used with the IPSec profile.
Step 7	**exit** **Example:** `Router(config-crypto-map)# exit`	Exits crypto map configuration mode.
Step 8	**redundancy inter-device** **Example:** `Router(config)# redundancy inter-device`	Configures redundancy and enters inter-device configuration mode.
Step 9	**security ipsec** *profile-name* **Example:** `Router(config-red-interdevice)# security ipsec sso-secure`	Applies the IPSec profile to the redundancy group communications, protecting all SSO traffic that is passed between the active and standby device.

Examples

The following example shows how to configure SSO traffic protection:

```
crypto isakmp key cisco123 address 0.0.0.0 0.0.0.0 no-xauth
!
crypto ipsec transform-set trans2 ah-md5-hmac esp-aes
!
crypto ipsec profile sso-secure
 set transform-set trans2
!
redundancy inter-device
 scheme standby HA-in
 security ipsec sso-secure
```

Managing and Verifying High Availability Information

Use any of the following optional tasks to secure and manage your high availability configurations:

- Managing Anti-Replay Intervals, page 21
- Managing and Verifying HA Configurations, page 22

Managing Anti-Replay Intervals

Use this optional task to modify the interval in which an IP redundancy-enabled crypto map forwards anti-replay updates from the active router to the standby router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name* **redundancy replay-interval inbound** *in-value* **outbound** *out-value*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name* **redundancy replay-interval inbound** *in-value* **outbound** *out-value* **Example:** `Router(config)# crypto map to-peer-outside redundancy replay-interval inbound 1000 outbound 10000`	Modifies the interval at which inbound and outbound replay counters are passed from an active device to a standby device. • **inbound** *in-value*—Number of inbound packets that are processed before an anti-replay update is sent from the active router to the standby router. Default value: one update every 1,000 packets. • **outbound** *out-value*—Number of outbound packets that are processed before an anti-replay update is sent from the active router to the standby router. Default value: one update every 100,000 packets.

Examples

The following example shows how to modify replay counter intervals between the active and standby devices on the crypto map “to-peer-outside”:

```
crypto map to-peer-outside redundancy replay-interval inbound 1000 outbound 10000
crypto map to-peer-outside 10 ipsec-isakmp
 set peer 209.165.200.225
 set transform-set trans1
 match address peer-outside
```

Managing and Verifying HA Configurations

Use any of the steps within this optional task to display and verify the high availability configurations.

SUMMARY STEPS

1. **enable**
2. **show redundancy** [**states** | **inter-device**]
3. **show crypto isakmp sa** [**active** | **standby**]
4. **show crypto ipsec sa** [**active** | **standby**]
5. **show crypto session** [**active** | **standby**]
6. **show crypto ha**
7. **clear crypto isakmp** [**active** | **standby**]
8. **clear crypto sa** [**active** | **standby**]
9. **clear crypto session** [**active** | **standby**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show redundancy** [**states**	**inter-device**] **Example:** `Router# show redundancy states`	Displays the current state of SSO on the configured device. After the two devices have negotiated with each other, one device should show an "ACTIVE" state and the other device should show a "STANDBY HOT" state.
Step 3	**show crypto isakmp sa** [**active**	**standby**] **Example:** `Router# show crypto isakmp sa active`	Displays IKE SAs present on the device. An "ACTIVE" or "STDBY" state is shown for each SA. • The **active** keyword displays only ACTIVE, HA-enabled SAs; The **standby** keyword displays only STDBY, HA-enabled SAs.
Step 4	**show crypto ipsec sa** [**active**	**standby**] **Example:** `Router# show crypto ipsec sa active`	Displays IPSec SAs present on the device. An "ACTIVE" or "STDBY" state is shown for each SA. • The **active** keyword displays only ACTIVE, HA-enabled SAs; The **standby** keyword displays only STDBY, HA-enabled SAs.
Step 5	**show crypto session** [**active**	**standby**] **Example:** `Router# show crypto session active`	Displays crypto sessions that are currently present on the device. An "ACTIVE" or "STANDBY" state is shown as part of the state of each session, such as "UP-STANDBY." Only HA-enabled SAs are shown.

	Command or Action	Purpose
Step 6	**show crypto ha** **Example:** `Router# show crypto ha`	Displays all virtual IP addresses that are currently in use by IPSec and IKE.
Step 7	**clear crypto isakmp** [**active** | **standby**] **Example:** `Router# clear crypto isakmp active`	Clears IKE SAs. When this command is issued on the standby device, all standby IKE SAs are resynchronized from the active device. • The **active** keyword clears only IKE HA-enabled SAs in the active state; the **standby** keyword clears only IKE HA-enabled SAs in the standby state.
Step 8	**clear crypto sa** [**active** | **standby**] **Example:** `Router# clear crypto sa active`	Clears IPSec SAs. When this command is issued on the standby device, all standby IPSec SAs are resynchronized from the active device. • The **active** keyword clears only IPSec HA-enabled SAs in the active state; the **standby** keyword clears only IPSec HA-enabled SAs in the standby state.
Step 9	**clear crypto session** [**active** | **standby**] **Example:** `Router# clear crypto session active`	Clears both IKE and IPSec SAs. Any standby SAs will resynchronize from the active device after they are cleared on the standby. Only HA-enabled SAs are cleared from the device.

Examples

Verifying the Active Device:Examples

```
Router# show redundancy states

       my state = 13 -ACTIVE
     peer state = 8  -STANDBY HOT
           Mode = Duplex
        Unit ID = 0

     Split Mode = Disabled
   Manual Swact = Enabled
 Communications = Up

   client count = 7
 client_notification_TMR = 30000 milliseconds
          keep_alive TMR = 4000 milliseconds
        keep_alive count = 0
    keep_alive threshold = 7
           RF debug mask = 0x0

Router# show crypto isakmp sa active

dst             src             state          conn-id slot status
209.165.201.3   209.165.200.225 QM_IDLE              5    0 ACTIVE
```

```
Router# show crypto ipsec sa active

interface:Ethernet0/0
    Crypto map tag:to-peer-outside, local addr 209.165.201.3

   protected vrf:(none)
   local  ident (addr/mask/prot/port):(192.168.0.1/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port):(172.16.0.1/255.255.255.255/0/0)
   current_peer 209.165.200.225 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps:3, #pkts encrypt:3, #pkts digest:3
    #pkts decaps:4, #pkts decrypt:4, #pkts verify:4
    #pkts compressed:0, #pkts decompressed:0
    #pkts not compressed:0, #pkts compr. failed:0
    #pkts not decompressed:0, #pkts decompress failed:0
    #send errors 0, #recv errors 0

     local crypto endpt.:209.165.201.3, remote crypto endpt.:209.165.200.225
     path mtu 1500, media mtu 1500
     current outbound spi:0xD42904F0(3559458032)

     inbound esp sas:
      spi:0xD3E9ABD0(3555306448)
        transform:esp-3des ,
        in use settings ={Tunnel, }
        conn id:2006, flow_id:6, crypto map:to-peer-outside
        sa timing:remaining key lifetime (k/sec):(4586265/3542)
             HA last key lifetime sent(k):(4586267)
        ike_cookies:9263635C CA4B4E99 C14E908E 8EE2D79C
        IV size:8 bytes
        replay detection support:Y
        Status:ACTIVE
inbound ah sas:
      spi: 0xF3EE3620(4092474912)
        transform: ah-md5-hmac ,
        in use settings ={Tunnel, }
        conn id: 2006, flow_id: 6, crypto map: to-peer-outside
        sa timing: remaining key lifetime (k/sec): (4586265/3542)
             HA last key lifetime sent(k): (4586267)
        ike_cookies: 9263635C CA4B4E99 C14E908E 8EE2D79C
        replay detection support: Y
        Status: ACTIVE

     inbound pcp sas:

     outbound esp sas:
      spi: 0xD42904F0(3559458032)
        transform: esp-3des ,
        in use settings ={Tunnel, }
        conn id: 2009, flow_id: 9, crypto map: to-peer-outside
        sa timing: remaining key lifetime (k/sec): (4586266/3542)
             HA last key lifetime sent(k): (4586267)
        ike_cookies: 9263635C CA4B4E99 C14E908E 8EE2D79C
        IV size: 8 bytes
        replay detection support: Y
        Status: ACTIVE

     outbound ah sas:
      spi: 0x75251086(1965363334)
        transform: ah-md5-hmac ,
        in use settings ={Tunnel, }
        conn id: 2009, flow_id: 9, crypto map: to-peer-outside
        sa timing: remaining key lifetime (k/sec): (4586266/3542)
             HA last key lifetime sent(k): (4586267)
```

```
        ike_cookies: 9263635C CA4B4E99 C14E908E 8EE2D79C
        replay detection support: Y
        Status: ACTIVE

     outbound pcp sas:


Router# show crypto session active
Crypto session current status

Interface: Ethernet0/0
Session status: UP-ACTIVE
Peer: 209.165.200.225 port 500
  IKE SA: local 209.165.201.3/500 remote 209.165.200.225/500 Active
  IKE SA: local 209.165.201.3/500 remote 209.165.200.225/500 Active
  IPSEC FLOW: permit ip host 192.168.0.1 host 172.16.0.1
        Active SAs: 4, origin: crypto map


Router# show crypto ha
IKE VIP: 209.165.201.3
  stamp: 74 BA 70 27 9C 4F 7F 81 3A 70 13 C9 65 22 E7 76
IPSec VIP: 209.165.201.3
IPSec VIP: 255.255.255.253
IPSec VIP: 255.255.255.254
```

Verifying the Standby Device: Examples

```
Router# show redundancy states
       my state = 8  -STANDBY HOT
     peer state = 13 -ACTIVE
           Mode = Duplex
        Unit ID = 0
     Split Mode = Disabled
   Manual Swact = Enabled
 Communications = Up
   client count = 7
 client_notification_TMR = 30000 milliseconds
          keep_alive TMR = 4000 milliseconds
        keep_alive count = 1
    keep_alive threshold = 7
           RF debug mask = 0x0

Router# show crypto isakmp sa standby
dst             src             state          conn-id slot status
209.165.201.3   209.165.200.225 QM_IDLE              5    0 STDBY


Router# show crypto ipsec sa standby
interface:Ethernet0/0
    Crypto map tag:to-peer-outside, local addr 209.165.201.3
   protected vrf:(none)
   local  ident (addr/mask/prot/port):(192.168.0.1/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port):(172.16.0.1/255.255.255.255/0/0)
   current_peer 209.165.200.225 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps:0, #pkts encrypt:0, #pkts digest:0
    #pkts decaps:0, #pkts decrypt:0, #pkts verify:0
    #pkts compressed:0, #pkts decompressed:0
    #pkts not compressed:0, #pkts compr. failed:0
    #pkts not decompressed:0, #pkts decompress failed:0
    #send errors 0, #recv errors 0
     local crypto endpt.:209.165.201.3, remote crypto endpt.:209.165.200.225
     path mtu 1500, media mtu 1500
```

```
      current outbound spi:0xD42904F0(3559458032)
      inbound esp sas:
       spi:0xD3E9ABD0(3555306448)
         transform:esp-3des ,
         in use settings ={Tunnel, }
         conn id:2012, flow_id:12, crypto map:to-peer-outside
         sa timing:remaining key lifetime (k/sec):(4441561/3486)
              HA last key lifetime sent(k):(4441561)
         ike_cookies:00000000 00000000 00000000 00000000
         IV size:8 bytes
         replay detection support:Y
         Status:STANDBY
      inbound ah sas:
       spi:0xF3EE3620(4092474912)
         transform:ah-md5-hmac ,
         in use settings ={Tunnel, }
         conn id:2012, flow_id:12, crypto map:to-peer-outside
         sa timing:remaining key lifetime (k/sec):(4441561/3486)
              HA last key lifetime sent(k):(4441561)
         ike_cookies:00000000 00000000 00000000 00000000
         replay detection support:Y
         Status:STANDBY
      inbound pcp sas:
      outbound esp sas:
       spi:0xD42904F0(3559458032)
         transform:esp-3des ,
         in use settings ={Tunnel, }
         conn id:2011, flow_id:11, crypto map:to-peer-outside
         sa timing:remaining key lifetime (k/sec):(4441561/3485)
              HA last key lifetime sent(k):(4441561)
         ike_cookies:00000000 00000000 00000000 00000000
         IV size:8 bytes
         replay detection support:Y
         Status:STANDBY
      outbound ah sas:
       spi:0x75251086(1965363334)
         transform:ah-md5-hmac ,
         in use settings ={Tunnel, }
         conn id:2011, flow_id:11, crypto map:to-peer-outside
         sa timing:remaining key lifetime (k/sec):(4441561/3485)
              HA last key lifetime sent(k):(4441561)
         ike_cookies:00000000 00000000 00000000 00000000
         replay detection support:Y
         Status:STANDBY
      outbound pcp sas:

Router# show crypto session standby
Crypto session current status
Interface:Ethernet0/0
Session status:UP-STANDBY
Peer:209.165.200.225 port 500
  IKE SA:local 209.165.201.3/500 remote 209.165.200.225/500 Active
  IPSEC FLOW:permit ip host 192.168.0.1 host 172.16.0.1
        Active SAs:4, origin:crypto map

Router# show crypto ha
IKE VIP:209.165.201.3
  stamp:74 BA 70 27 9C 4F 7F 81 3A 70 13 C9 65 22 E7 76

IPSec VIP:209.165.201.3
IPSec VIP:255.255.255.253
IPSec VIP:255.255.255.254
ha-R2#
```

Verifying the Active and Standby SAs: Example

The following sample output shows SAs of both the active and standby devices:

```
Router# show crypto isakmp sa
dst             src             state          conn-id slot status
209.165.201.3   209.165.200.225 QM_IDLE              2    0 STDBY
10.0.0.1        10.0.0.2        QM_IDLE              1    0 ACTIVE
```

Configuration Examples for Stateful Failover

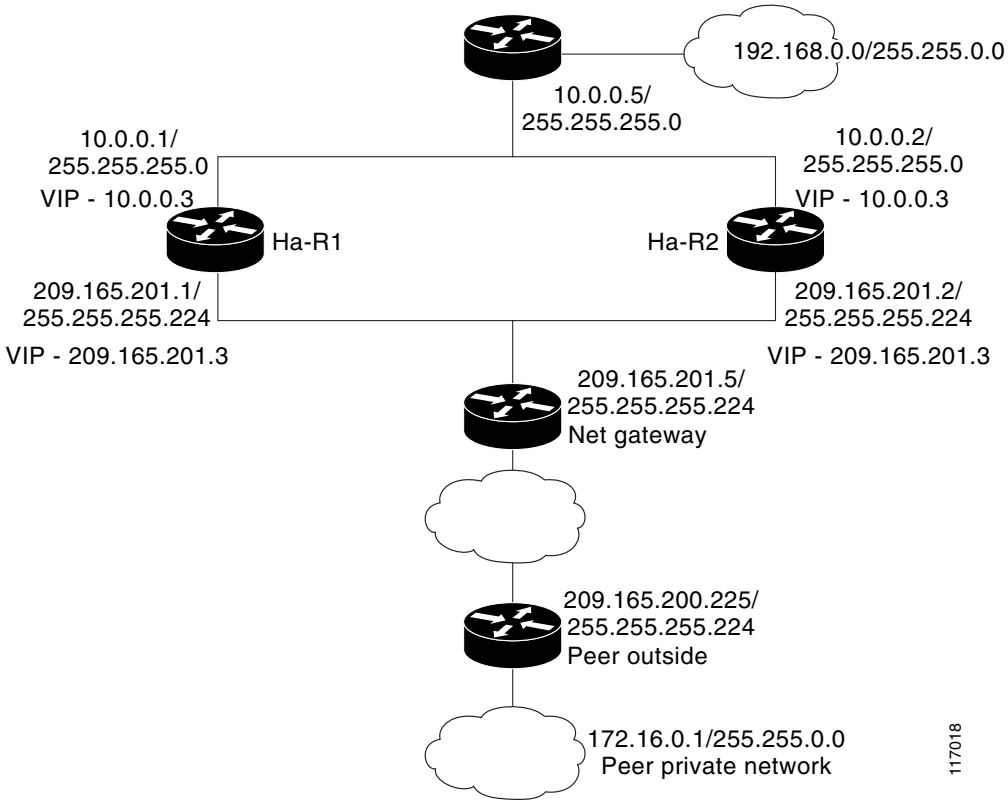
This section contains the following comprehensive IPSec stateful failover configuration examples:

- Configuring IPSec Stateful Failover: Example, page 27
- Configuring IPSec Stateful Failover for an Easy VPN Server: Example, page 31

Configuring IPSec Stateful Failover: Example

Figure 3 and the following sample outputs from the show running-config command illustrate how to configure stateful failover on two devices—Ha-R1 and Ha-R2.

Figure 95 IPSec Stateful Failover Sample Topology



Stateful Failover Configuration on Ha-R1

```
Ha-R1#show running-config
Building configuration...

Current configuration :2086 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname ha-R1
!
boot-start-marker
boot-end-marker
!
!
redundancy inter-device
 scheme standby HA-in
 security ipsec sso-secure
!
logging buffered 10000000 debugging
logging rate-limit console 10000
!
!
ipc zone default
 association 1
  no shutdown
  protocol sctp
   local-port 5000
    local-ip 10.0.0.1
   remote-port 5000
    remote-ip 10.0.0.2
!
clock timezone PST 0
no aaa new-model
ip subnet-zero
!
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco123 address 0.0.0.0 0.0.0.0 no-xauth
!
!
crypto ipsec transform-set trans1 ah-md5-hmac esp-3des
crypto ipsec transform-set trans2 ah-md5-hmac esp-aes
!
crypto ipsec profile sso-secure
 set transform-set trans2
!
!
crypto map to-peer-outside redundancy replay-interval inbound 1000 outbound 10000
crypto map to-peer-outside 10 ipsec-isakmp
 set peer 209.165.200.225
 set transform-set trans1
 match address peer-outside
!
!
!
interface Ethernet0/0
 ip address 209.165.201.1 255.255.255.224
 standby 1 ip 209.165.201.3
 standby 1 preempt
 standby 1 name HA-out
```

```
 standby 1 track Ethernet1/0
 standby delay reload 120
 crypto map to-peer-outside redundancy HA-out stateful
!
interface Ethernet1/0
 ip address 10.0.0.1 255.255.255.0
 standby 2 ip 10.0.0.3
 standby 2 preempt
 standby 2 name HA-in
 standby delay reload 120
 standby 2 track Ethernet0/0
!
interface Serial2/0
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial3/0
 no ip address
 shutdown
 serial restart-delay 0
!
ip classless
ip route 0.0.0.0 0.0.0.0 209.165.201.5
ip route 192.168.0.0 255.255.0.0
no ip http server
no ip http secure-server
!
!
!
ip access-list extended peer-outside
 permit ip host 192.168.0.1 host 172.16.0.1
!
!
control-plane
!
!
line con 0
 exec-timeout 0 0
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
 login
 transport preferred all
 transport input all
 transport output all
!
end
```

Stateful Failover Configuration on Ha-R2

```
Ha-R2#show running-config
Building configuration...

Current configuration :2100 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
```

```
hostname ha-R2
!
boot-start-marker
boot-end-marker
!
!
redundancy inter-device
 scheme standby HA-in
 security ipsec sso-secure
!
logging buffered 10000000 debugging
logging rate-limit console 10000
!
!
ipc zone default
 association 1
  no shutdown
  protocol sctp
   local-port 5000
    local-ip 10.0.0.2
   remote-port 5000
    remote-ip 10.0.0.1
!
clock timezone PST 0
no aaa new-model
ip subnet-zero
!
!
crypto isakmp policy 1
 authentication pre-share
 lifetime 120
crypto isakmp key cisco123 address 0.0.0.0 0.0.0.0 no-xauth
!
!
crypto ipsec transform-set trans1 ah-md5-hmac esp-3des
crypto ipsec transform-set trans2 ah-md5-hmac esp-aes
!
crypto ipsec profile sso-secure
 set transform-set trans2
!
!
crypto map to-peer-outside redundancy replay-interval inbound 1000 outbound 10000
crypto map to-peer-outside 10 ipsec-isakmp
 set peer 209.165.200.225
 set transform-set trans1
 match address peer-outside
!
!
!
interface Ethernet0/0
 ip address 209.165.201.2 255.255.255.224
 standby 1 ip 209.165.201.3
 standby 1 preempt
 standby 1 name HA-out
 standby 1 track Ethernet1/0
 standby delay reload 120
 crypto map to-peer-outside redundancy HA-out stateful
!
interface Ethernet1/0
 ip address 10.0.0.2 255.255.255.0
 standby 2 ip 10.0.0.3
 standby 2 preempt
 standby 2 name HA-in
 standby delay reload 120
```

```
 standby 2 track Ethernet0/0
!
interface Serial2/0
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial3/0
 no ip address
 shutdown
 serial restart-delay 0
!
ip classless
ip route 0.0.0.0 0.0.0.0 209.165.201.5
ip route 192.168.0.0 255.255.0.0
no ip http server
no ip http secure-server
!
!
!
ip access-list extended peer-outside
 permit ip host 192.168.0.1 host 172.16.0.1
!
!
control-plane
!
!
line con 0
 exec-timeout 0 0
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
 login
 transport preferred all
 transport input all
 transport output all
!
end

Ha-R2#
```

Configuring IPSec Stateful Failover for an Easy VPN Server: Example

The following sample outputs from the **show running-config** command show how to configure stateful failover for a remote access connection via an Easy VPN server:

Stateful Failover for an Easy VPN Server Configuration on RAHA-R1

```
RAHA-R1# show running-config
Building configuration...

Current configuration :3829 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
```

```
hostname RAHA-R1
!
boot-start-marker
boot-end-marker
!
redundancy inter-device
 scheme standby HA-in
!
username remote_user password 0 letmein
!
ipc zone default
 association 1
  no shutdown
  protocol sctp
   local-port 5000
    local-ip 10.0.0.1
   remote-port 5000
    remote-ip 10.0.0.2
!
aaa new-model
!
!
! Enter the following command if you are doing Xauth locally.
aaa authentication login local_xauth local
!
! Enter the following command if you are doing Xauth remotely via RADIUS.
!aaa authentication login radius_xauth group radius
!
! Enter the following command if you are not doing Xauth
!aaa authentication login no_xauth none
!
! Enter the following command if you are doing local group authentication.
aaa authorization network local_auth local
!
! Enter the following command if you are doing group authentication remotely via RADIUS.
!aaa authorization network radius_auth group radius
!
!
! Enter the following command if you are doing Xauth remotely via RADIUS.
!
aaa accounting network radius_accounting start-stop group radius
aaa session-id common
ip subnet-zero
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
 group 2
!
!
! Enter the following command if you are doing group authentication locally.
crypto isakmp client configuration group unity
 key cisco123
 domain cisco.com
 pool client-address-pool
!
!
crypto ipsec transform-set trans1 esp-3des esp-sha-hmac
!
crypto dynamic-map to-remote-client 10
 set transform-set trans1
 reverse-route remote-peer
!
```

```
! Use this map if you want to do local group authentication and Xauth.
crypto map to_peer_outside_local_xauth client authentication list local_xauth
crypto map to_peer_outside_local_xauth isakmp authorization list local_auth
crypto map to_peer_outside_local_xauth client configuration address respond
crypto map to_peer_outside_local_xauth 10 ipsec-isakmp dynamic to-remote-client
!
! Use this map if you want to use Radius for group authentication and Xauth.
!crypto map to_peer_outside_radius_xauth isakmp client authentication list radius_xauth
!crypto map to_peer_outside_radius_xauth client accounting list radius_accounting
!crypto map to_peer_outside_radius_xauth isakmp authorization list radius_auth
!crypto map to_peer_outside_radius_xauth isakmp client configuration address respond
!crypto map to_peer_outside_radius_xauth isakmp 10 ipsec-isakmp dynamic to-remote-client
!
! Use this map if you want to do local group authentication and no Xauth
!crypto map to_peer_outside_no_xauth isakmp authorization list local_auth
!crypto map to_peer_outside_no_xauth configuration address respond
!crypto map to_peer_outside_no_xauth 10 ipsec-isakmp dynamic to-remote-client
!
interface Ethernet0/0
 ip address 209.165.201.1 255.255.255.224
 standby 1 ip 209.165.201.3
 standby 1 preempt
 standby 1 name HA-out
 standby 1 track Ethernet1/0
 standby delay reload 120
 crypto map to_peer_outside_local_xauth redundancy HA-out stateful
!
interface Ethernet1/0
 ip address 10.0.0.1 255.255.255.0
 standby 2 ip 10.0.0.3
 standby 2 preempt
 standby 2 name HA-in
 standby 2 track Ethernet0/0
 standby delay reload 120
!
! Enable loopback0 if you are using radius for Xauth, group auth, or accounting with
! crypto HA
!interface loopback0
! ip address 192.168.100.1 255.255.255.255
!
! Enable this command if you are using radius for Xauth, group auth, or accounting with
! crypto HA
!ip radius source-interface loopback0
!
ip local pool client-address-pool 50.0.0.1 50.0.0.254
ip classless
ip route 0.0.0.0 0.0.0.0 209.165.201.5
ip route 192.168.0.0 255.255.255.0 10.0.0.5
!
radius-server host 192.168.0.0 255.255.0.0 auth-port 1845 acct-port 1846
radius-server key radius123
!
control-plane
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
!
end
```

Stateful Failover for an Easy VPN Server Configuration on RAHA-R2

```
RAHA-R2# show running-config
Building configuration...

Current configuration :3829 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname RAHA-R2
!
boot-start-marker
boot-end-marker
!
redundancy inter-device
 scheme standby HA-in
!
username remote_user password 0 letmein
!
ipc zone default
 association 1
  no shutdown
  protocol sctp
   local-port 5000
    local-ip 10.0.0.2
   remote-port 5000
    remote-ip 10.0.0.1
!
aaa new-model
!
!
! Enter the following command if you are doing Xauth locally.
aaa authentication login local_xauth local
!
! Enter the following command if you are doing Xauth remotely via RADIUS.
!aaa authentication login radius_xauth group radius
!
! Enter the following command if you are not doing Xauth.
!aaa authentication login no_xauth none
!
! Enter the following command if you are doing local group authentication.
aaa authorization network local_auth local
!
! Enter the following command if you are doing group authentication remotely via RADIUS.
!aaa authorization network radius_auth group radius
!
!
! Enter the following command if you are doing Xauth remotely via RADIUS.
!aaa accounting network radius_accounting start-stop group radius
aaa session-id common
ip subnet-zero
!
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
 group 2
!
!
! Enter the following commands if you are doing group authentication locally.
crypto isakmp client configuration group unity
```

```
 key cisco123
 domain cisco.com
 pool client-address-pool
!
!
crypto ipsec transform-set trans1 esp-3des esp-sha-hmac
!
crypto dynamic-map to-remote-client 10
 set transform-set trans1
 reverse-route remote-peer
!
!
! Use this map if you want to dolocal group authentication and Xauth.
crypto map to_peer_outside_local_xauth client authentication list local_xauth
crypto map to_peer_outside_local_xauth isakmp authorization list local_auth
crypto map to_peer_outside_local_xauth client configuration address respond
crypto map to_peer_outside_local_xauth 10 ipsec-isakmp dynamic to-remote-client
!
! Use this map if you want to use Radius for group authentication and Xauth.
!crypto map to_peer_outside_radius_xauth isakmp client authentication list radius_xauth
!crypto map to_peer_outside_radius_xauth client accounting list radius_accounting
!crypto map to_peer_outside_radius_xauth isakmp authorization list radius_auth
!crypto map to_peer_outside_radius_xauth isakmp client configuration address respond
!crypto map to_peer_outside_radius_xauth isakmp 10 ipsec-isakmp dynamic to-remote-client
!
!
! Use this map if you want to do local authentication and no Xauth.
!crypto map to_peer_outside_no_xauth isakmp authorization list local_auth
!crypto map to_peer_outside_no_xauth configuration address respond
!crypto map to_peer_outside_no_xauth 10 ipsec-isakmp dynamic to-remote-client
!
interface Ethernet0/0
 ip address 209.165.201.2 255.255.255.224
 standby 1 ip 209.165.201.3
 standby 1 preempt
 standby 1 name HA-out
 standby 1 track Ethernet1/0
 standby delay reload
 crypto map to_peer_outside_local_xauth redundancy HA-out stateful
!
interface Ethernet1/0
 ip address 10.0.0.2 255.255.255.0
 standby 2 ip 10.0.0.3
 standby 2 preempt
 standby 2 name HA-in
 standby 2 track Ethernet0/0
 standby delay reload
!
! Enable loopback0 if you are using radius for Xauth, group auth, or accounting with
! crypto HA
!interface loopback0
! ip address 192.168.100.1 255.255.255.255
!
! Enable this command if you are using radius for Xauth, group auth, or accounting with
! crypto HA
!ip radius source-interface loopback0
!
ip local pool client-address-pool 50.0.0.1 50.0.0.254
ip classless
ip route 0.0.0.0 0.0.0.0 209.165.201.5
ip route 192.168.0.0 255.255.0.0
!
radius-server host 192.168.0.200 auth-port 1845 acct-port 1846
radius-server key radius123
```

```
!
control-plane
!
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
!
end
```

Additional References

The following sections provide references related to stateful failover for IPSec.

Related Documents

Related Topic	Document Title
RRI	*IPSec VPN High Availability Enhancements*, Cisco IOS Release 12.2(11)T feature module
HSRP	The section "Configuring the Hot Standby Router Protocol" within the chapter "Configuring IP Services" of the *Cisco IOS IP Configuration Guide*, Release 12.3
Easy VPN Server	*Cisco Easy VPN Remote*, Cisco IOS Release 12.3(7)T feature module
IPSec and IKE configuration	The section "IP Security and Encryption" in the *Cisco IOS Security Configuration Guide*, Release 12.3
IPSec and IKE commands	*Cisco IOS Security Command Reference*, Release 12.3 T

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

New Commands

- **crypto map redundancy replay-interval**
- **debug crypto ha**
- **debug crypto ipsec ha**
- **debug crypto isakmp ha**
- **local-ip (IPC transport-SCTP local)**
- **local-port**
- **redundancy inter-device**
- **redundancy stateful**
- **remote-ip (IPC transport-SCTP remote)**
- **remote-port**
- **scheme**
- **security ipsec**
- **show crypto ha**

Modified Commands

- **clear crypto isakmp**
- **clear crypto sa**
- **clear crypto session**

- **crypto map (interface IPSec)**
- **show crypto ipsec sa**
- **show crypto isakmp sa**
- **show crypto session**
- **show redundancy**

VRF-Aware IPSec

The VRF-Aware IPSec feature introduces IP Security (IPSec) tunnel mapping to Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs). Using the VRF-Aware IPSec feature, you can map IPSec tunnels to Virtual Routing and Forwarding (VRF) instances using a single public-facing address.

Feature Specifications for VRF-Aware IPSec

Feature History	
Release	**Modification**
12.2(15)T	This feature was introduced.
Supported Platforms	
Cisco 1710, Cisco 1760, Cisco 2610-Cisco 2613, Cisco 2620-Cisco 2621, Cisco 2650-Cisco 2651, Cisco 3620, Cisco 3640, Cisco 3660, Cisco 7100, Cisco 7200, Cisco 7400	

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Restrictions for VRF-Aware IPSec, page 1358
- Information About VRF-Aware IPSec, page 1358
- How to Configure VRF-Aware IPSec, page 1360
- Configuration Examples for VRF-Aware IPSec, page 1377
- Additional References, page 1389
- Command Reference, page 1391
- Glossary, page 1393

Restrictions for VRF-Aware IPSec

- The VRF-Aware IPSec feature does not allow IPSec tunnel mapping between VRFs. For example, it does not allow IPSec tunnel mapping from VRF vpn1 to VRF vpn2.

Information About VRF-Aware IPSec

The VRF-Aware IPSec feature maps an IPSec tunnel to a MPLS VPN. To configure and use the feature, you need to understand the following concepts:

- VRF Instance, page 1358
- MPLS Distribution Protocol, page 1358
- VRF-Aware IPSec Functional Overview, page 1358

VRF Instance

A VRF instance is a per-VPN routing information repository that defines the VPN membership of a customer site attached to the Provider Edge (PE) router. A VRF comprises an IP routing table, a derived Cisco Express Forwarding (CEF) table, a set of interfaces that use the forwarding table, and a set of rules and routing protocol parameters that control the information that is included in the routing table. A separate set of routing and CEF tables is maintained for each VPN customer.

MPLS Distribution Protocol

The MPLS distribution protocol is a high-performance packet-forwarding technology that integrates the performance and traffic management capabilities of data link layer switching with the scalability, flexibility, and performance of network-layer routing.

VRF-Aware IPSec Functional Overview

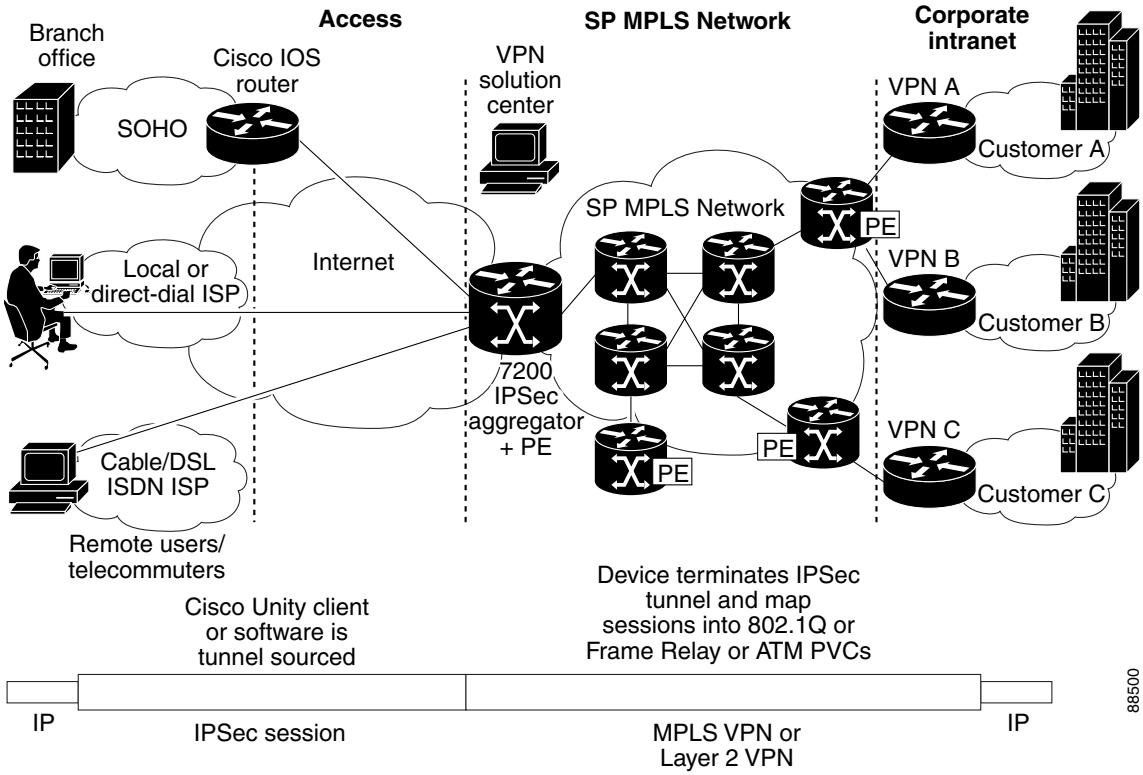
Front Door VRF (FVRF) and Inside VRF (IVRF) are central to understanding the feature.

Each IPSec tunnel is associated with two VRF domains. The outer encapsulated packet belongs to one VRF domain, which we shall call the FVRF, while the inner, protected IP packet belongs to another domain called the IVRF. Another way of stating the same thing is that the local endpoint of the IPSec tunnel belongs to the FVRF while the source and destination addresses of the inside packet belong to the IVRF.

One or more IPSec tunnels can terminate on a single interface. The FVRF of all these tunnels is the same and is set to the VRF that is configured on that interface. The IVRF of these tunnels can be different and depends on the VRF that is defined in the Internet Security Association and Key Management Protocol (ISAKMP) profile that is attached to a crypto map entry.

Figure 96 is an illustration of a scenario showing IPSec to MPLS and Layer 2 VPNs.

Figure 96 IPSec to MPLS and Layer 2 VPNs



Packet Flow into the IPSec Tunnel

- A VPN packet arrives from the Service Provider MPLS backbone network to the PE and is routed through an interface facing the Internet.
- The packet is matched against the Security Policy Database (SPD), and the packet is IPSec encapsulated. The SPD includes the IVRF and the access control list (ACL).
- The IPSec encapsulated packet is then forwarded using the FVRF routing table.

Packet Flow from the IPSec Tunnel

- An IPSec-encapsulated packet arrives at the PE router from the remote IPSec endpoint.
- IPSec performs the Security Association (SA) lookup for the Security Parameter Index (SPI), destination, and protocol.
- The packet is decapsulated using the SA and is associated with IVRF.
- The packet is further forwarded using the IVRF routing table.

How to Configure VRF-Aware IPSec

This section contains the following procedures:

- Configuring Crypto Keyrings, page 1360 (Optional)
- Configuring ISAKMP Profiles, page 1362 (Required)
- Configuring an ISAKMP Profile on a Crypto Map, page 1366 (Required)
- Configuring to Ignore Extended Authentication During IKE Phase 1 Negotiation, page 1367 (Optional)
- Verifying VRF-Aware IPSec, page 1367
- Clearing Security Associations, page 1368
- Troubleshooting VRF-Aware IPSec, page 1369

Configuring Crypto Keyrings

A crypto keyring is a repository of preshared and Rivest, Shamir, and Adelman (RSA) public keys. There can be zero or more keyrings on the Cisco IOS router.

Perform the following optional task to configure a crypto keyring.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto keyring** *keyring-name* [**vrf** *fvrf-name*]
4. **description** *string* (Optional)
5. **pre-shared-key** {**address** *address* [*mask*] | **hostname** *hostname*} **key** *key* (Optional)
6. **rsa-pubkey** {**address** *address* | **name** *fqdn*} [**encryption** | **signature**] (Optional)
7. **address** *ip-address* (Optional)
8. **serial-number** *serial-number* (Optional)
9. **key-string**
10. **text**
11. **quit**
12. **exit**
13. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto keyring** *keyring-name* [**vrf** *fvrf-name*] **Example:** `Router (config)# crypto keyring VPN1`	Defines a keyring with *keyring-name* as the name of the keyring and enters keyring configuration mode. • (Optional) The **vrf** keyword and *fvrf-name* argument imply that the keyring is bound to Front Door Virtual Routing and Forwarding (FVRF). The key in the keyring is searched if the local endpoint is in FVRF. If **vrf** is not specified, the keyring is bound to the global.
Step 4	**description** *string* `Router (config-keyring)# description The keys for VPN1`	(Optional) Specifies a one-line description of the keyring.
Step 5	**pre-shared-key** {**address** *address* [*mask*] | **hostname** *hostname*} **key** *key* **Example:** `Router (config-keyring)# pre-shared-key address 10.72.23.11 key VPN1`	(Optional) Defines a preshared key by address or host name.
Step 6	**rsa-pubkey** {**address** *address* | **name** *fqdn*} [**encryption** | **signature**] **Example:** `Router(config-keyring)# rsa-pubkey name host.vpn.com`	(Optional) Defines a Rivest, Shamir, and Adelman (RSA) public key by address or host name and enters rsa-pubkey configuration mode. • By default, the key is used for signature. • The optional **encryption** keyword specifies that the key should be used for encryption. The optional **signature** keyword specifies that the key should be used for signature. By default, the key is used for signature.
Step 7	**address** *ip-address* **Example:** `Router(config-pubkey-key)# address 10.5.5.1`	(Optional) Defines the RSA public key IP address.
Step 8	**serial-number** *serial-number* **Example:** `Router(config-pubkey-key)# serial-number 1000000`	(Optional) Specifies the serial number of the public key. The value is from 0 through infinity.

	Command or Action	Purpose
Step 9	**key-string** **Example:** `Router (config-pubkey-key)# key-string`	Enters into the text mode in which you define the public key.
Step 10	**text** **Example:** `Router (config-pubkey)# 00302017 4A7D385B 1234EF29 335FC973`	Specifies the public key. **Note** Only one public key may be added in this step.
Step 11	**quit** **Example:** `Router (config-pubkey)# quit`	Quits to the public key configuration mode.
Step 12	**exit** **Example:** `Router (config-pubkey)# exit`	Exits to the keyring configuration mode.
Step 13	**exit** **Example:** `Router(config-keyring)# exit#`	Exits to global configuration mode.

Configuring ISAKMP Profiles

An ISAKMP profile is a repository for IKE Phase 1 and IKE Phase 1.5 configuration for a set of peers. An ISAKMP profile defines items such as keepalive, trustpoints, peer identities, and XAUTH AAA list during the IKE Phase 1 and Phase 1.5 exchange. There can be zero or more ISAKMP profiles on the Cisco IOS router.

Note

- If traffic from the router to a certification authority (CA) (for authentication, enrollment, or for obtaining a certificate revocation list [CRL]) or to an Lightweight Directory Access Protocol (LDAP) server (for obtaining a CRL) needs to be routed via a VRF, the **vrf** command must be added to the trustpoint. Otherwise, such traffic will use the default routing table.
- If a profile does not specify one or more trustpoints, all trustpoints in the router will be used to attempt to validate the certificate of the peer (Internet Key Exchange (IKE) main mode or signature authentication). If one or more trustpoints are specified, only those trustpoints will be used.

Restriction

A router initiating IKE and a router responding to the IKE request should have symmetrical trustpoint configurations. For example, a responding router (in IKE Main Mode) performing RSA signature encryption and authentication might use trustpoints that were defined in the global configuration when sending the CERT-REQ payloads. However, the router might use a restricted list of trustpoints that were defined in the ISAKMP profile for the certificate verification. If the peer (the IKE initiator) is configured

to use a certificate whose trustpoint is in the global list of the responding router but not in ISAKMP profile of the responding router, the certificate will be rejected. (However, if the initiating router does not know about the trustpoints in the global configuration of the responding router, the certificate can still be authenticated.)

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp profile** *profile-name*
4. **description** *string* (Optional)
5. **vrf** i*vrf-name* (Optional)
6. **keepalive** *seconds* **retry** *retry-seconds* (Optional)
7. **self-identity** {**address** | **fqdn** | **user-fqdn** *user-fqdn*} (Optional)
8. **keyring** *keyring-name* (Optional)
9. **ca trust-point** *trustpoint-name* (Optional)
10. **match identity** {**group** *group-name* | **address** *address* [*mask*] [*fvrf]* | **host** *host-name* | **host domain** *domain-name* | **user** *user-fqdn* | **user domain** *domain-name*}
11. **client configuration address** {**initiate** | **respond**} (Optional)
12. **client authentication list** *list-name* (Optional)
13. **isakmp authorization list** *list-name* (Optional)
14. **initiate mode aggressive**
15. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp profile** *profile-name* **Example:** `Router (config)# crypto isakmp profile vpnprofile`	Defines an Internet Security Association and Key Management Protocol (ISAKMP) profile and enters into isakmp profile configuration mode.

	Command or Action	Purpose
Step 4	**description** *string* **Example:** `Router (conf-isa-prof)# description configuration for VPN profile`	(Optional) Specifies a one-line description of an ISAKMP profile.
Step 5	**vrf** i*vrf-name* **Example:** `Router (conf-isa-prof)# vrf VPN1`	(Optional) Maps the IPSec tunnel to a Virtual Routing and Forwarding (VRF) instance. **Note** The VRF also serves as a selector for matching the Security Policy Database (SPD). If the VRF is not specified in the ISAKMP profile, the IVRF of the IPSec tunnel will be the same as its FVRF.
Step 6	**keepalive** *seconds* **retry** *retry-seconds* **Example:** `Router (conf-isa-prof)# keepalive 60 retry 5`	(Optional) Allows the gateway to send dead peer detection (DPD) messages to the peer. • If not defined, the gateway uses the global configured value. • *seconds*—Number of seconds between DPD messages. The range is from 10 to 3600 seconds. • **retry** *retry-seconds*—Number of seconds between retries if the DPD message fails. The range is from 2 to 60 seconds.
Step 7	**self-identity** {**address** | **fqdn** | **user-fqdn** *user-fqdn*} **Example:** `Router (conf-isa-prof)# self-identity address`	(Optional) Specifies the identity that the local Internet Key Exchange (IKE) should use to identify itself to the remote peer. • If not defined, IKE uses the global configured value. • **address**—Uses the IP address of the egress interface. • **fqdn—**Uses the fully qualified domain name (FQDN) of the router. • **user-fqdn**—Uses the specified value.
Step 8	**keyring** *keyring-name* **Example:** `Router (conf-isa-prof)# keyring VPN1`	(Optional) Specifies the keyring to use for Phase 1 authentication. • If the keyring is not specified, the global key definitions are used.
Step 9	**ca trust-point** {*trustpoint-name*} **Example:** `Router (conf-isa-prof)# ca trustpoint VPN1-trustpoint`	(Optional) Specifies a trustpoint to validate a Rivest, Shamir, and Adelman (RSA) certificate. • If no trustpoint is specified in the ISAKMP profile, all the trustpoints that are configured on the Cisco IOS router are used to validate the certificate.

	Command or Action	Purpose
Step 10	**match identity** {**group** *group-name* | **address** *address* [*mask*] [*fvrf*] | **host** *host-name* | **host domain** *domain-name* | **user** *user-fqdn* | **user domain** *domain-name*} **Example:** `Router (conf-isa-prof)# match identity address 10.1.1.1`	Specifies the client IKE Identity (ID) that is to be matched. • **group** *group-name*—Matches the *group-name* with the ID type ID_KEY_ID. It also matches the *group-name* with the Organizational Unit (OU) field of the Distinguished Name (DN).· • **address** *address* [*mask*] *fvrf* —Matches the *address* with the ID type ID_IPV4_ADDR. The *mask* argument can be used to specify a range of addresses. The *fvrf argument* specifies that the address is in Front Door Virtual Routing and Forwarding (FVRF) • **host** *hostname*—Matches the *hostname* with the ID type ID_FQDN. • **host domain** *domainname*—Matches the *domainname* to the ID type ID_FQDN whose domain name is the same as the *domainname*. Use this command to match all the hosts in the domain. • **user** *username*—Matches the *username* with the ID type ID_USER_FQDN· • **user domain** *domainname*—Matches the ID type ID_USER_FQDN whose domain name matches the *domainname*.
Step 11	**client configuration address** {**initiate** | **respond**} **Example:** `Router (conf-isa-prof)# client configuration address initiate`	(Optional) Specifies whether to initiate the mode configuration exchange or responds to mode configuration requests.
Step 12	**client authentication list** *list-name* **Example:** `Router (conf-isa-prof)# client authentication list xauthlist`	(Optional) Authentication, authorization, and accounting (AAA) to use for authenticating the remote client during the extended authentication (XAUTH) exchange.
Step 13	**isamkp authorization list** *list-name* **Example:** `Router (conf-isa-prof)# isakmp authorization list ikessaaalist`	(Optional) Network authorization server for receiving the Phase 1 preshared key and other attribute-value (AV) pairs.
Step 14	**initiate mode aggressive** **Example:** `Router (conf-isa-prof)# initiate mode aggressive`	(Optional) Initiates aggressive mode exchange. • If not specified, IKE always initiates Main Mode exchange.
Step 15	**exit** **Example:** `Router (conf-isa-prof)# exit`	Exits to global configuration mode.

What to Do Next

Go to the section "Configuring an ISAKMP Profile on a Crypto Map."

Configuring an ISAKMP Profile on a Crypto Map

An ISAKMP profile must be applied to the crypto map. The IVRF on the ISAKMP profile is used as a selector when matching the VPN traffic. If there is no IVRF on the ISAKMP profile, the IVRF will be equal to the FVRF. Perform this required task to configure an ISAKMP profile on a crypto map.

Prerequisites

Before configuring an ISAKMP profile on a crypto map, you must first have configured your router for basic IPSec.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name* **isakmp-profile** *isakmp-profile-name* (Optional)
4. **set isakmp-profile** *profile-name* (Optional)
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name* **isakmp-profile** *isakmp-profile-name* **Example:** `Router (config)# crypto map vpnmap isakmp-profile vpnprofile`	(Optional) Specifies the Internet Key Exchange and Key Management Protocol (ISAKMP) profile for the crypto map set and enters crypto map configuration mode. • The ISAKMP profile will be used during IKE exchange.

	Command or Action	Purpose
Step 4	**set isakmp-profile** *profile-name* **Example:** `Router (config-crypto-map)# set isakmp-profile vpnprofile`	(Optional) Specifies the ISAKMP profile to use when the traffic matches the crypto map entry.
Step 5	**exit** **Example:** `Router (config-crypto-map)# exit`	Exits to global configuration mode.

Configuring to Ignore Extended Authentication During IKE Phase 1 Negotiation

To ignore XAUTH during an IKE Phase 1 negotiation, use the **no crypto xauth** command. Use the **no crypto xauth** command if you do not require extended authentication for the Unity clients.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no crypto xauth** *interface*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**no crypto xauth** *interface* **Example:** `Router(config)# no crypto xauth ethernet0`	Ignores XAUTH proposals for requests that are destined to the IP address of the interface. By default, Internet Key Exchange (IKE) processes XAUTH proposals.

Verifying VRF-Aware IPSec

To verify your VRF-Aware IPSec configurations, use the following **show** commands. These **show** commands allow you to list configuration information and security associations (SAs):

SUMMARY STEPS

- **enable**
- **show crypto ipsec sa** [**map** *map-name* | **address** | **identity** | **interface** *interface* | **peer** [**vrf** *fvrf-name*] **address** | **vrf** *ivrf-name*] [**detail**]
- **show crypto isakmp key**
- **show crypto isakmp profile**
- **show crypto key pubkey-chain rsa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto ipsec sa** [**map** *map-name*	**address**	**identity**	**interface** *interface*	**peer** [**vrf** *fvrf-name*] **address**	**vrf** *ivrf-name*] [**detail**] **Example:** `Router# show crypto ipsec sa vrf vpn1`	Allows you to view the settings used by current security associations (SAs).
Step 3	**show crypto isakmp key** **Example:** `Router# show crypto isakmp key`	Lists all the keyrings and their preshared keys. • Use this command to verify your crypto keyring configuration.
Step 4	**show crypto isakmp profile** **Example:** `Router# show crypto isakmp profile`	Lists all ISAKMP profiles and their configurations.
Step 5	**show crypto key pubkey-chain rsa** **Example:** `Router# show crypto key pubkey-chain rsa`	Views the Rivest, Shamir, and Adelman (RSA) public keys of the peer that are stored on your router. • The output is extended to show the keyring to which the public key belongs.

Clearing Security Associations

The following **clear** commands allow you to clear SAs.

SUMMARY STEPS

- **enable**
- **clear crypto sa** [**counters** | **map** *map-name* | **peer** [**vrf** *fvrf-name*] *address* | **spi** *address* {**ah** | **esp**} *spi* | **vrf** *ivrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear crypto sa** [**counters** | **map** *map-name* | **peer** [**vrf** *fvrf-name*] *address* | **spi** *address* {**ah** | **esp**} *spi* | **vrf** *ivrf-name*] **Example:** `Router# clear crypto sa vrf VPN1`	Clears the IPSec security associations (SAs).

Troubleshooting VRF-Aware IPSec

To troubleshoot VRF-Aware IPSec, use the following **debug** commands:

SUMMARY STEPS

1. **enable**
2. **debug crypto ipsec**
3. **debug crypto isakmp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto ipsec** **Example:** `Router# debug crypto ipsec`	Displays IP security (IPSec) events.
Step 3	**debug crypto isakmp** **Example:** `Router(config)# debug crypto isakmp`	Displays messages about Internet Key Exchange (IKE) events.

Debug Examples for VRF-Aware IPSec

The following sample debug outputs are for a VRF-aware IPSec configuration:

IPSec PE

```
Router# debug crypto ipsec
```

```
Crypto IPSEC debugging is on
IPSEC-PE#debug crypto isakmp
Crypto ISAKMP debugging is on
IPSEC-PE#debug crypto isakmp d
04:31:28: ISAKMP (0:12): purging SA., sa=6482B354, delme=6482B354
04:31:28: ISAKMP: Unlocking IKE struct 0x63C142F8 for declare_sa_dead(), count 0
IPSEC-PE#debug crypto isakmp detail
Crypto ISAKMP internals debugging is on
IPSEC-PE#
IPSEC-PE#
IPSEC-PE#
04:32:07: ISAKMP: Deleting peer node by peer_reap for 10.1.1.1: 63C142F8
04:32:55: ISAKMP cookie gen for src 172.16.1.1 dst 10.1.1.1
04:32:55: ISAKMP cookie 3123100B DC887D4E
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.68.1.1
04:32:55: ISAKMP cookie AA8F7B41 49A60E88
04:32:55: ISAKMP cookie gen for src 172.16.1.1 dst 10.1.1.1
04:32:55: ISAKMP cookie 3123100B DBC8E125
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.16.1.1
04:32:55: ISAKMP cookie AA8F7B41 B4BDB5B7
04:32:55: ISAKMP (0:0): received packet from 10.1.1.1 dport 500 sport 500 Global (N) NEW
SA
04:32:55: ISAKMP: local port 500, remote port 500
04:32:55: ISAKMP: hash from 729FA94 for 619 bytes
04:32:55: ISAKMP: Packet hash:
64218CC0:                   B91E2C70 095A1346          9.,p.Z.F
64218CD0: 0EDB4CA6 8A46784F B314FD3B 00        .[L&.FxO.};.
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:32:55: ISAKMP cookie AA8F7B41 F7ACF384
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:32:55: ISAKMP cookie AA8F7B41 0C07C670
04:32:55: ISAKMP: insert sa successfully sa = 6482B354
04:32:55: ISAKMP (0:13): processing SA payload. message ID = 0
04:32:55: ISAKMP (0:13): processing ID payload. message ID = 0
04:32:55: ISAKMP (0:13): peer matches vpn2-ra profile
04:32:55: ISAKMP: Looking for a matching key for 10.1.1.1 in default
04:32:55: ISAKMP: Created a peer struct for 10.1.1.1, peer port 500
04:32:55: ISAKMP: Locking peer struct 0x640BBB18, IKE refcount 1 for
crypto_ikmp_config_initialize_sa
04:32:55: ISAKMP (0:13): Setting client config settings 648252B0
04:32:55: ISAKMP (0:13): (Re)Setting client xauth list  and state
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID seems Unity/DPD but major 157 mismatch
04:32:55: ISAKMP (0:13): vendor ID is NAT-T v3
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID seems Unity/DPD but major 123 mismatch
04:32:55: ISAKMP (0:13): vendor ID is NAT-T v2
04:32:55: ISAKMP (0:13) Authentication by xauth preshared
04:32:55: ISAKMP (0:13): Checking ISAKMP transform 1 against priority 1 policy
04:32:55: ISAKMP:      encryption 3DES-CBC
04:32:55: ISAKMP:      hash SHA
04:32:55: ISAKMP:      default group 2
04:32:55: ISAKMP:      auth XAUTHInitPreShared
04:32:55: ISAKMP:      life type in seconds
04:32:55: ISAKMP:      life duration (VPI) of  0x0 0x20 0xC4 0x9B
04:32:55: ISAKMP (0:13): atts are acceptable. Next payload is 3
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID seems Unity/DPD but major 157 mismatch
04:32:55: ISAKMP (0:13): vendor ID is NAT-T v3
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID seems Unity/DPD but major 123 mismatch
04:32:55: ISAKMP (0:13): vendor ID is NAT-T v2
04:32:55: ISAKMP (0:13): processing KE payload. message ID = 0
```

```
04:32:55: ISAKMP (0:13): processing NONCE payload. message ID = 0
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID is DPD
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID seems Unity/DPD but major 175 mismatch
04:32:55: ISAKMP (0:13): vendor ID is XAUTH
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): claimed IOS but failed authentication
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID is Unity
04:32:55: ISAKMP (0:13): Input = IKE_MESG_FROM_PEER, IKE_AM_EXCH
04:32:55: ISAKMP (0:13): Old State = IKE_READY  New State = IKE_R_AM_AAA_AWAIT

04:32:55: ISAKMP cookie gen for src 11.1.1.1 dst 172.16.1.1
04:32:55: ISAKMP cookie AA8F7B41 7AE6E1DF
04:32:55: ISAKMP:        isadb_post_process_list: crawler: 4 AA 31 (6482B354)
04:32:55:       crawler my_cookie AA8F7B41 F7ACF384
04:32:55:        crawler his_cookie E46E088D F227FE4D
04:32:55: ISAKMP: got callback 1
04:32:55: ISAKMP (0:13): SKEYID state generated
04:32:55: ISAKMP: Unity/DPD ID: vendor_id_payload:
        next: 0xD, reserved: 0x0, len 0x14
04:32:55: ISAKMP: Unity/DPD ID payload dump:
63E66D70:                            0D000014              ....
63E66D80: 12F5F28C 457168A9 702D9FE2 74CC0100  .ur.Eqh)p-.btL..
63E66D90: 00                                   .
04:32:55: ISAKMP: Unity/DPD ID: vendor_id_payload:
        next: 0xD, reserved: 0x0, len 0x14
04:32:55: ISAKMP: Unity/DPD ID payload dump:
63E66D90: 0D000014 AFCAD713 68A1F1C9 6B8696FC  ..../JW.h!qIk..|
63E66DA0: 77570100 00                         wW...
04:32:55: ISAKMP (0:13): constructed NAT-T vendor-03 ID
04:32:55: ISAKMP (0:13): SA is doing pre-shared key authentication plus XAUTH using id
type ID_IPV4_ADDR
04:32:55: ISAKMP (13): ID payload
        next-payload : 10
        type         : 1
        addr         : 172.16.1.1
        protocol     : 17
        port         : 0
        length       : 8
04:32:55: ISAKMP (13): Total payload length: 12
04:32:55: ISAKMP (0:13): constructed HIS NAT-D
04:32:55: ISAKMP (0:13): constructed MINE NAT-D
04:32:55: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R)
AG_INIT_EXCH
04:32:55: ISAKMP (0:13): Input = IKE_MESG_FROM_AAA, PRESHARED_KEY_REPLY
04:32:55: ISAKMP (0:13): Old State = IKE_R_AM_AAA_AWAIT  New State = IKE_R_AM2

04:32:55: ISAKMP cookie gen for src 172.16.1.1 dst 10.1.1.1
04:32:55: ISAKMP cookie 3123100B D99DA70D
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.16.1.1
04:32:55: ISAKMP cookie AA8F7B41 9C69F917
04:32:55: ISAKMP:        isadb_post_process_list: crawler: 5 21FF 1 (6482B354)
04:32:55:       crawler my_cookie AA8F7B41 F7ACF384
04:32:55:        crawler his_cookie E46E088D F227FE4D
04:32:55: ISAKMP cookie gen for src 172.16.1.1 dst 10.1.1.1
04:32:55: ISAKMP cookie 3123100B 00583224
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.16.1.1
04:32:55: ISAKMP cookie AA8F7B41 C1B006EE
04:32:55: ISAKMP:        isadb_post_process_list: crawler: 5 21FF 1 (6482B354)
04:32:55:       crawler my_cookie AA8F7B41 F7ACF384
04:32:55:        crawler his_cookie E46E088D F227FE4D
```

```
04:32:55: ISAKMP (0:13): received packet from 10.1.1.1 dport 500 sport 500 Global (R)
AG_INIT_EXCH
04:32:55: ISAKMP: hash from 7003A34 for 132 bytes
04:32:55: ISAKMP: Packet hash:
64218CC0:                   D1202D99 2BB49D38          Q -.+4.8
64218CD0: B8FBB1BE 7CDC67D7 4E26126C 63        8{1>|\gWN&.lc
04:32:55: ISAKMP (0:13): processing HASH payload. message ID = 0
04:32:55: ISAKMP:received payload type 17
04:32:55: ISAKMP (0:13): Detected NAT-D payload
04:32:55: ISAKMP (0:13): recalc my hash for NAT-D
04:32:55: ISAKMP (0:13): NAT match MINE hash
04:32:55: ISAKMP:received payload type 17
04:32:55: ISAKMP (0:13): Detected NAT-D payload
04:32:55: ISAKMP (0:13): recalc his hash for NAT-D
04:32:55: ISAKMP (0:13): NAT match HIS hash
04:32:55: ISAKMP (0:13): processing NOTIFY INITIAL_CONTACT protocol 1
        spi 0, message ID = 0, sa = 6482B354
04:32:55: ISAKMP (0:13): Process initial contact,
bring down existing phase 1 and 2 SA's with local 172.16.1.1 remote 10.1.1.1 remote port
500
04:32:55: ISAKMP (0:13): returning IP addr to the address pool
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.16.1.1
04:32:55: ISAKMP cookie AA8F7B41 05D315C5
04:32:55: ISAKMP cookie gen for src 172.16.1.1 dst 10.1.1.1
04:32:55: ISAKMP cookie 3123100B 041A85A6
04:32:55: ISAKMP (0:13): SA has been authenticated with 10.1.1.1
04:32:55: ISAKMP: Trying to insert a peer 172.16.1.1/10.1.1.1/500/,  and inserted
successfully.
04:32:55: ISAKMP: set new node -803402627 to CONF_XAUTH
04:32:55: IPSEC(key_engine): got a queue event...
04:32:55: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R) QM_IDLE
04:32:55: ISAKMP (0:13): purging node -803402627
04:32:55: ISAKMP: Sending phase 1 responder lifetime 86400

04:32:55: ISAKMP (0:13): Input = IKE_MESG_FROM_PEER, IKE_AM_EXCH
04:32:55: ISAKMP (0:13): Old State = IKE_R_AM2  New State = IKE_P1_COMPLETE

04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.168.1.1
04:32:55: ISAKMP cookie AA8F7B41 25EEF256
04:32:55: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:32:55:       crawler my_cookie AA8F7B41 F7ACF384
04:32:55:        crawler his_cookie E46E088D F227FE4D
04:32:55: ISAKMP (0:13): Need XAUTH
04:32:55: ISAKMP (0:13): Input = IKE_MESG_INTERNAL, IKE_PHASE1_COMPLETE
04:32:55: ISAKMP (0:13): Old State = IKE_P1_COMPLETE  New State =
IKE_XAUTH_AAA_START_LOGIN_AWAIT

04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.16.1.1
04:32:55: ISAKMP cookie AA8F7B41 2CCFA491
04:32:55: ISAKMP:        isadb_post_process_list: crawler: B 27FF 12 (6482B354)
04:32:55:       crawler my_cookie AA8F7B41 F7ACF384
04:32:55:        crawler his_cookie E46E088D F227FE4D
04:32:55: ISAKMP: got callback 1
04:32:55: ISAKMP: set new node -1447732198 to CONF_XAUTH
04:32:55: ISAKMP/xauth: request attribute XAUTH_USER_NAME_V2
04:32:55: ISAKMP/xauth: request attribute XAUTH_USER_PASSWORD_V2
04:32:55: ISAKMP (0:13): initiating peer config to 10.1.1.1. ID = -1447732198
04:32:55: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R)
CONF_XAUTH
04:32:55: ISAKMP (0:13): Input = IKE_MESG_FROM_AAA, IKE_AAA_START_LOGIN
04:32:55: ISAKMP (0:13): Old State = IKE_XAUTH_AAA_START_LOGIN_AWAIT  New State =
IKE_XAUTH_REQ_SENT

04:33:00: ISAKMP (0:13): retransmitting phase 2 CONF_XAUTH    -1447732198 ...
```

```
04:33:00: ISAKMP (0:13): incrementing error counter on sa: retransmit phase 2
04:33:00: ISAKMP (0:13): incrementing error counter on sa: retransmit phase 2
04:33:00: ISAKMP (0:13): retransmitting phase 2 -1447732198 CONF_XAUTH
04:33:00: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R)
CONF_XAUTH
04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B 124D4618
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 B0C91917
04:33:03: ISAKMP:        isadb_post_process_list: crawler: B 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B 0E294692
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 091A7695
04:33:03: ISAKMP:        isadb_post_process_list: crawler: B 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP (0:13): received packet from 10.1.1.1 dport 500 sport 500 Global (R)
CONF_XAUTH
04:33:03: ISAKMP: hash from 7292D74 for 92 bytes
04:33:03: ISAKMP: Packet hash:
64218CC0:                   84A1AF24 5D92B116          .!/$].1.
64218CD0: FC2C6252 A472C5F8 152AC860 63        |,bR$rEx.*H`c
04:33:03: ISAKMP (0:13): processing transaction payload from 11.1.1.1. message ID =
-1447732198
04:33:03: ISAKMP: Config payload REPLY
04:33:03: ISAKMP/xauth: reply attribute XAUTH_USER_NAME_V2
04:33:03: ISAKMP/xauth: reply attribute XAUTH_USER_PASSWORD_V2
04:33:03: ISAKMP (0:13): deleting node -1447732198 error FALSE reason "done with xauth
request/reply exchange"
04:33:03: ISAKMP (0:13): Input = IKE_MESG_FROM_PEER, IKE_CFG_REPLY
04:33:03: ISAKMP (0:13): Old State = IKE_XAUTH_REQ_SENT  New State =
IKE_XAUTH_AAA_CONT_LOGIN_AWAIT

04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 A1B3E684
04:33:03: ISAKMP:        isadb_post_process_list: crawler: B 27FF 12 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP: got callback 1
04:33:03: ISAKMP: set new node 524716665 to CONF_XAUTH
04:33:03: ISAKMP (0:13): initiating peer config to 10.1.1.1. ID = 524716665
04:33:03: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R)
CONF_XAUTH
04:33:03: ISAKMP (0:13): Input = IKE_MESG_FROM_AAA, IKE_AAA_CONT_LOGIN
04:33:03: ISAKMP (0:13): Old State = IKE_XAUTH_AAA_CONT_LOGIN_AWAIT  New State =
IKE_XAUTH_SET_SENT
004:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B 5C83A09D
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 2BEBEFD4
04:33:03: ISAKMP:        isadb_post_process_list: crawler: B 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B DA00A46B
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 FDD27773
04:33:03: ISAKMP:        isadb_post_process_list: crawler: B 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
```

```
04:33:03: ISAKMP (0:13): received packet from 10.1.1.1 dport 500 sport 500 Global (R)
CONF_XAUTH
04:33:03: ISAKMP: hash from 7292A34 for 68 bytes
04:33:03: ISAKMP: Packet hash:
64218CC0:                   5034B99E B8BA531F          P49.8:S.
64218CD0: 6267B8BD F3006989 DC118796 63        bg8=s.i.\...c
04:33:03: ISAKMP (0:13): processing transaction payload from 11.1.1.1. message ID =
524716665
04:33:03: ISAKMP: Config payload ACK
04:33:03: ISAKMP (0:13):        XAUTH ACK Processed
04:33:03: ISAKMP (0:13): deleting node 524716665 error FALSE reason "done with
transaction"
04:33:03: ISAKMP (0:13): Input = IKE_MESG_FROM_PEER, IKE_CFG_ACK
04:33:03: ISAKMP (0:13): Old State = IKE_XAUTH_SET_SENT  New State = IKE_P1_COMPLETE

04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 E0BB50E9
04:33:03: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP (0:13): Input = IKE_MESG_INTERNAL, IKE_PHASE1_COMPLETE
04:33:03: ISAKMP (0:13): Old State = IKE_P1_COMPLETE  New State = IKE_P1_COMPLETE

04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B 7794EF6E
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 C035AAE5
04:33:03: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B F1FCC25A
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 31744F44
04:33:03: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F207FE4D
04:33:03: ISAKMP (0:13): received packet from 10.1.1.1 dport 500 sport 500 Global (R)
QM_IDLE
04:33:03: ISAKMP: set new node -1639992295 to QM_IDLE
04:33:03: ISAKMP: hash from 7293A74 for 100 bytes
04:33:03: ISAKMP: Packet hash:
64218CC0:                   9D7DF4DF FE3A6403          .}t_~:d.
64218CD0: 3F1D1C59 C5D138CE 50289B79 07        ?..YEQ8NP(.y.
04:33:03: ISAKMP (0:13): processing transaction payload from 10.1.1.1. message ID =
-1639992295
04:33:03: ISAKMP: Config payload REQUEST
04:33:03: ISAKMP (0:13): checking request:
04:33:03: ISAKMP:    IP4_ADDRESS
04:33:03: ISAKMP:    IP4_NETMASK
04:33:03: ISAKMP:    IP4_DNS
04:33:03: ISAKMP:    IP4_DNS
04:33:03: ISAKMP:    IP4_NBNS
04:33:03: ISAKMP:    IP4_NBNS
04:33:03: ISAKMP:    SPLIT_INCLUDE
04:33:03: ISAKMP:    DEFAULT_DOMAIN
04:33:03: ISAKMP (0:13): Input = IKE_MESG_FROM_PEER, IKE_CFG_REQUEST
04:33:03: ISAKMP (0:13): Old State = IKE_P1_COMPLETE  New State =
IKE_CONFIG_AUTHOR_AAA_AWAIT

04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 B02E0D67
04:33:03: ISAKMP:        isadb_post_process_list: crawler: C 27FF 12 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
```

```
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP: got callback 1
04:33:03: ISAKMP (0:13): attributes sent in message:
04:33:03:         Address: 10.2.0.0
04:33:03: ISAKMP (0:13): allocating address 10.4.1.4
04:33:03: ISAKMP: Sending private address: 10.4.1.4
04:33:03: ISAKMP: Sending DEFAULT_DOMAIN default domain name: vpn2.com
04:33:03: ISAKMP (0:13): responding to peer config from 10.1.1.1. ID = -1639992295
04:33:03: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R)
CONF_ADDR
04:33:03: ISAKMP (0:13): deleting node -1639992295 error FALSE reason ""
04:33:03: ISAKMP (0:13): Input = IKE_MESG_FROM_AAA, IKE_AAA_GROUP_ATTR
04:33:03: ISAKMP (0:13): Old State = IKE_CONFIG_AUTHOR_AAA_AWAIT  New State =
IKE_P1_COMPLETE

04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B 881D5411
04:33:03: ISAKMP cookie gen for src 11.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 6FD82541
04:33:03: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B 8A94C1BE
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 F3BA766D
04:33:03: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F207FE4D
04:33:03: ISAKMP (0:13): received packet from 10.1.1.1 dport 500 sport 500 Global (R)
QM_IDLE
04:33:03: ISAKMP: set new node 17011691 to QM_IDLE
04:33:03: ISAKMP: hash from 70029F4 for 540 bytes
04:33:03: ISAKMP: Packet hash:
64218CC0:                   AFBA30B2 55F5BC2D          /:02Uu<-
64218CD0: 3A86B1C9 00D2F5BA 77BF5589 07        :.1I.Ru:w?U..
04:33:03: ISAKMP (0:13): processing HASH payload. message ID = 17011691
04:33:03: ISAKMP (0:13): processing SA payload. message ID = 17011691
04:33:03: ISAKMP (0:13): Checking IPSec proposal 1
04:33:03: ISAKMP: transform 1, ESP_3DES
04:33:03: ISAKMP:   attributes in transform:
04:33:03: ISAKMP:      encaps is 1
04:33:03: ISAKMP:      SA life type in seconds
04:33:03: ISAKMP:      SA life duration (VPI) of  0x0 0x20 0xC4 0x9B
04:33:03: ISAKMP:      SA life type in kilobytes
04:33:03: ISAKMP:      SA life duration (VPI) of  0x0 0x46 0x50 0x0
04:33:03: ISAKMP:      authenticator is HMAC-SHA
04:33:03: ISAKMP (0:13): atts are acceptable.
04:33:03: IPSEC(validate_proposal_request): proposal part #1,
  (key eng. msg.) INBOUND local= 172.18.1.1, remote= 10.1.1.1,
    local_proxy= 0.0.0.0/0.0.0.0/0/0 (type=4),
    remote_proxy= 10.4.1.4/255.255.255.255/0/0 (type=1),
    protocol= ESP, transform= esp-3des esp-sha-hmac ,
    lifedur= 0s and 0kb,
    spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x2
04:33:03: IPSEC(kei_proxy): head = ra, map->ivrf = vpn1, kei->ivrf = vpn2
04:33:03: IPSEC(kei_proxy): head = ra, map->ivrf = vpn2, kei->ivrf = vpn2
04:33:03: IPSEC(validate_transform_proposal): transform proposal not supported for
identity:
    {esp-3des esp-sha-hmac }
04:33:03: ISAKMP (0:13): IPSec policy invalidated proposal
04:33:03: ISAKMP (0:13): Checking IPSec proposal 2
04:33:03: ISAKMP: transform 1, ESP_3DES
04:33:03: ISAKMP:   attributes in transform:
```

```
04:33:03: ISAKMP:      encaps is 1
04:33:03: ISAKMP:      SA life type in seconds
04:33:03: ISAKMP:      SA life duration (VPI) of  0x0 0x20 0xC4 0x9B
04:33:03: ISAKMP:      SA life type in kilobytes
04:33:03: ISAKMP:      SA life duration (VPI) of  0x0 0x46 0x50 0x0
04:33:03: ISAKMP:      authenticator is HMAC-MD5
04:33:03: ISAKMP (0:13): atts are acceptable.
04:33:03: IPSEC(validate_proposal_request): proposal part #1,
  (key eng. msg.) INBOUND local= 172.18.1.1, remote= 10.1.1.1,
    local_proxy= 0.0.0.0/0.0.0.0/0/0 (type=4),
    remote_proxy= 10.4.1.4/255.255.255.255/0/0 (type=1),
    protocol= ESP, transform= esp-3des esp-md5-hmac ,
    lifedur= 0s and 0kb,
    spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x2
04:33:03: IPSEC(kei_proxy): head = ra, map->ivrf = vpn1, kei->ivrf = vpn2
04:33:03: IPSEC(kei_proxy): head = ra, map->ivrf = vpn2, kei->ivrf = vpn2
04:33:03: ISAKMP (0:13): processing NONCE payload. message ID = 17011691
04:33:03: ISAKMP (0:13): processing ID payload. message ID = 17011691
04:33:03: ISAKMP (0:13): processing ID payload. message ID = 17011691
04:33:03: ISAKMP (0:13): asking for 1 spis from ipsec
04:33:03: ISAKMP (0:13): Node 17011691, Input = IKE_MESG_FROM_PEER, IKE_QM_EXCH
04:33:03: ISAKMP (0:13): Old State = IKE_QM_READY  New State = IKE_QM_SPI_STARVE
04:33:03: IPSEC(key_engine): got a queue event...
04:33:03: IPSEC(spi_response): getting spi 2749516541 for SA
        from 172.18.1.1     to 10.1.1.1        for prot 3
04:33:03: ISAKMP: received ke message (2/1)
04:33:04: ISAKMP (13): ID payload
        next-payload : 5
        type         : 1
        addr         : 10.4.1.4
        protocol     : 0
        port         : 0
04:33:04: ISAKMP (13): ID payload
        next-payload : 11
        type         : 4
        addr         : 0.0.0.0
        protocol     : 0
        port         : 0
04:33:04: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R) QM_IDLE
04:33:04: ISAKMP (0:13): Node 17011691, Input = IKE_MESG_FROM_IPSEC, IKE_SPI_REPLY
04:33:04: ISAKMP (0:13): Old State = IKE_QM_SPI_STARVE  New State = IKE_QM_R_QM2
04:33:04: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:04: ISAKMP cookie 3123100B 93DE46D2
04:33:04: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:04: ISAKMP cookie AA8F7B41 088A0A16
04:33:04: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:33:04:       crawler my_cookie AA8F7B41 F7ACF384
04:33:04:        crawler his_cookie E46E088D F227FE4D
04:33:04: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:04: ISAKMP cookie 3123100B A8F23F73
04:33:04: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:04: ISAKMP cookie AA8F7B41 93D8D879
04:33:04: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:33:04:       crawler my_cookie AA8F7B41 F7ACF384
04:33:04:        crawler his_cookie E46E088D F227FE4D
04:33:04: ISAKMP (0:13): received packet from 10.1.1.1 dport 500 sport 500 Global (R)
QM_IDLE
04:33:04: ISAKMP: hash from 7290DB4 for 60 bytes
04:33:04: ISAKMP: Packet hash:
64218CC0:                   4BB45A92 7181A2F8          K4Z.q."x
64218CD0: 73CC12F8 091875C0 054F77CD 63        sL.x..u@.OwMc
04:33:04: ISAKMP: Locking peer struct 0x640BBB18, IPSEC refcount 1 for for stuff_ke
04:33:04: ISAKMP (0:13): Creating IPSec SAs
04:33:04:         inbound SA from 10.1.1.1 to 172.18.1.1 (f/i)  0/ 2
```

```
        (proxy 10.4.1.4 to 0.0.0.0)
04:33:04:         has spi 0xA3E24AFD and conn_id 5127 and flags 2
04:33:04:         lifetime of 2147483 seconds
04:33:04:         lifetime of 4608000 kilobytes
04:33:04:         has client flags 0x0
04:33:04:         outbound SA from 172.18.1.1     to 10.1.1.1        (f/i)  0/ 2 (proxy
0.0.0.0         to 10.4.1.4       )
04:33:04:         has spi 1343294712 and conn_id 5128 and flags A
04:33:04:         lifetime of 2147483 seconds
04:33:04:         lifetime of 4608000 kilobytes
04:33:04:         has client flags 0x0
04:33:04: ISAKMP (0:13): deleting node 17011691 error FALSE reason "quick mode done
(await)"
04:33:04: ISAKMP (0:13): Node 17011691, Input = IKE_MESG_FROM_PEER, IKE_QM_EXCH
04:33:04: ISAKMP (0:13): Old State = IKE_QM_R_QM2  New State = IKE_QM_PHASE2_COMPLETE
04:33:04: IPSEC(key_engine): got a queue event...
04:33:04: IPSEC(initialize_sas): ,
  (key eng. msg.) INBOUND local= 172.18.1.1, remote= 10.1.1.1,
    local_proxy= 0.0.0.0/0.0.0.0/0/0 (type=4),
    remote_proxy= 10.4.1.4/0.0.0.0/0/0 (type=1),
    protocol= ESP, transform= esp-3des esp-md5-hmac ,
    lifedur= 2147483s and 4608000kb,
    spi= 0xA3E24AFD(2749516541), conn_id= 5127, keysize= 0, flags= 0x2
04:33:04: IPSEC(initialize_sas): ,
  (key eng. msg.) OUTBOUND local= 172.18.1.1, remote= 10.1.1.1,
    local_proxy= 0.0.0.0/0.0.0.0/0/0 (type=4),
    remote_proxy= 10.4.1.4/0.0.0.0/0/0 (type=1),
    protocol= ESP, transform= esp-3des esp-md5-hmac ,
    lifedur= 2147483s and 4608000kb,
    spi= 0x50110CF8(1343294712), conn_id= 5128, keysize= 0, flags= 0xA
04:33:04: IPSEC(kei_proxy): head = ra, map->ivrf = vpn1, kei->ivrf = vpn2
04:33:04: IPSEC(kei_proxy): head = ra, map->ivrf = vpn2, kei->ivrf = vpn2
04:33:04: IPSEC(rte_mgr): VPN Route Added 10.4.1.4 255.255.255.255 via 10.1.1.1 in vpn2
04:33:04: IPSEC(add mtree): src 0.0.0.0, dest 10.4.1.4, dest_port 0

04:33:04: IPSEC(create_sa): sa created,
  (sa) sa_dest= 172.18.1.1, sa_prot= 50,
    sa_spi= 0xA3E24AFD(2749516541),
    sa_trans= esp-3des esp-md5-hmac , sa_conn_id= 5127
04:33:04: IPSEC(create_sa): sa created,
  (sa) sa_dest= 10.1.1.1, sa_prot= 50,
    sa_spi= 0x50110CF8(1343294712),
    sa_trans= esp-3des esp-md5-hmac , sa_conn_id= 5128
04:33:53: ISAKMP (0:13): purging node -1639992295
04:33:54: ISAKMP (0:13): purging node 17011691
```

Configuration Examples for VRF-Aware IPSec

The following examples show how to configure VRF-Aware IPSec:

- Static IPSec-to-MPLS VPN Example, page 1378
- IPSec-to-MPLS VPN Using RSA Encryption Example, page 1379
- IPSec-to-MPLS VPN with RSA Signatures Example, page 1381
- Upgrade from Previous Versions of the Cisco Network-Based IPSec VPN Solution, page 1383

Static IPSec-to-MPLS VPN Example

The following sample shows a static configuration that maps IPSec tunnels to MPLS VPNs. The configurations map IPSec tunnels to MPLS VPNs "VPN1" and "VPN2." Both of the IPSec tunnels terminate on a single public-facing interface.

IPSec PE Configuration

```
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
ip vrf vpn2
 rd 101:1
 route-target export 101:1
 route-target import 101:1
!
crypto keyring vpn1
  pre-shared-key address 172.16.1.1 key vpn1
crypto keyring vpn2
  pre-shared-key address 10.1.1.1 key vpn2
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
!
crypto isakmp profile vpn1
 vrf vpn1
 keyring vpn1
 match identity address 172.16.1.1 255.255.255.255
!
crypto isakmp profile vpn2
 vrf vpn2
 keyring vpn2
 match identity address 10.1.1.1 255.255.255.255
!
crypto ipsec transform-set vpn1 esp-3des esp-sha-hmac
crypto ipsec transform-set vpn2 esp-3des esp-md5-hmac
!
crypto map crypmap 1 ipsec-isakmp
 set peer 172.16.1.1
 set transform-set vpn1
 set isakmp-profile vpn1
 match address 101
crypto map crypmap 3 ipsec-isakmp
 set peer 10.1.1.1
 set transform-set vpn2
 set isakmp-profile vpn2
 match address 102
!
interface Ethernet1/1
 ip address 172.17.1.1 255.255.0.0
 tag-switching ip
!
interface Ethernet1/2
 ip address 172.18.1.1 255.255.255.0
 crypto map crypmap
!
ip route 172.16.1.1 255.255.255.255 172.168.1.2
ip route 10.1.1.1 255.255.255.255 172.18.1.2
```

```
ip route vrf vpn1 10.2.0.0 255.255.0.0 172.18.1.2 global
ip route vrf vpn2 10.2.0.0 255.255.0.0 172.18.1.2 global
!
access-list 101 permit ip 10.1.0.0 0.0.255.255 10.2.0.0 0.0.255.255
access-list 102 permit ip 10.1.0.0 0.0.255.255 10.2.0.0 0.0.255.255
```

IPSec Customer Provided Edge (CPE) Configuration for VPN1

```
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
crypto isakmp key vpn1 address 172.18.1.1
!
!
crypto ipsec transform-set vpn1 esp-3des esp-sha-hmac
!
crypto map vpn1 1 ipsec-isakmp
 set peer 172.18.1.1
 set transform-set vpn1
 match address 101
!
interface FastEthernet1/0
 ip address 172.16.1.1 255.255.255.0
 crypto map vpn1
!
interface FastEthernet1/1
 ip address 10.2.1.1 255.255.0.0
!
access-list 101 permit ip 10.2.0.0 0.0.255.255 10.1.0.0 0.0.255.255
!
```

IPSec CPE Configuration for VPN2

```
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
!
crypto isakmp key vpn2 address 172.18.1.1
!
!
crypto ipsec transform-set vpn2 esp-3des esp-md5-hmac
!
crypto map vpn2 1 ipsec-isakmp
 set peer 172.18.1.1
 set transform-set vpn2
 match address 101
!
interface FastEthernet0
 ip address 10.1.1.1 255.255.255.0
 crypto map vpn2
!
interface FastEthernet1
 ip address 10.2.1.1 255.255.0.0
!
access-list 101 permit ip 10.2.0.0 0.0.255.255 10.1.0.0 0.0.255.255
```

IPSec-to-MPLS VPN Using RSA Encryption Example

The following example shows an IPSec-to-MPLS configuration using RSA encryption:

PE Router Configuration

```
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
crypto isakmp policy 10
 authentication rsa-encr
!
crypto keyring vpn1
 rsa-pubkey address 172.16.1.1 encryption
  key-string
   305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00DBF381 00DDECC8
   DC4AA490 40320C52 9912D876 EB36717C 63DCA95C 7E5EC02A 84F276CE 292B42D7
   D664F324 3726F4E0 39D33093 ECB81B95 482511A5 F064C4B3 D5020301 0001
   quit
!
crypto isakmp profile vpn1
 vrf vpn1
 keyring vpn1
 match identity address 172.16.1.1 255.255.255.255
!
crypto ipsec transform-set vpn1 esp-3des esp-sha-hmac
!
crypto map crypmap 1 ipsec-isakmp
 set peer 172.16.1.1
 set transform-set vpn1
 set isakmp-profile vpn1
 match address 101
!
interface Ethernet1/1
 ip address 172.17.1.1 255.255.0.0
 tag-switching ip
!
interface Ethernet1/2
 ip address 172.18.1.1 255.255.255.0
 crypto map crypmap
!
ip route 172.16.1.1 255.255.255.255 172.18.1.2
ip route vrf vpn1 10.2.0.0 255.255.0.0 172.18.1.2 global
!
access-list 101 permit ip 10.1.0.0 0.0.255.255 10.2.0.0 0.0.255.255
```

IPSec CPE Configuration for VPN1

```
crypto isakmp policy 10
 authentication rsa-encr
!
crypto key pubkey-chain rsa
 addressed-key 172.18.1.1 encryption
 key-string
  3082011B 300D0609 2A864886 F70D0101 01050003 82010800 30820103 0281FB00
  C90CC78A 6002BDBA 24683396 B7D7877C 16D08C47 E00C3C10 63CF13BC 4E09EA23
  92EB8A48 4113F5A4 8796C8BE AD7E2DC1 3B0742B6 7118CE7C 1B0E21D1 AA9724A4
  4D74FCEA 562FF225 A2B11F18 E53C4415 61C3B741 3A06E75D B4F9102D 6163EE40
  16C68FD7 6532F660 97B59118 9C8DE3E5 4E2F2925 BBB87FCB 95223D4E A5E362DB
  215CB35C 260080805 17BBE1EF C3050E13 031F3D5B 5C22D16C FC8B1EC5 074F07A5
  D050EC80 7890D9C5 EC20D6F0 173FE2BA 89F5B5F9 2EADC9A6 D461921E 3D5B60016
  ABB8B6B9 E2124A21 93F0E4AE B487461B E7F1F1C4 032A0B0E 80DC3E15 CB268EC9
  5D76B9BD 3C78CB75 CE9F68C6 484D6573 CBC3EB59 4B5F3999 8F9D0203 010001
  quit
!
crypto ipsec transform-set vpn1 esp-3des esp-sha-hmac
```

```
!
crypto map vpn1 1 ipsec-isakmp
 set peer 172.18.1.1
 set transform-set vpn1
 match address 101
!
interface FastEthernet1/0
 ip address 172.16.1.1 255.255.255.0
 crypto map vpn1
!
interface FastEthernet1/1
 ip address 10.2.1.1 255.255.0.0
!
access-list 101 permit ip 10.2.0.0 0.0.255.255 10.1.0.0 0.0.255.255
!
```

IPSec-to-MPLS VPN with RSA Signatures Example

The following shows an IPSec-to-MPLS VPN configuration using RSA signatures:

PE Router Configuration

```
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
crypto ca trustpoint bombo
 enrollment url http://172.31.68.59:80
 crl optional
!
crypto ca certificate chain bombo
 certificate 03C0
 308203BF 308202A7 A0030201 02020203 C0300D06 092A8648 86F70D01 01050500
 . . .
 quit
 certificate ca 01
 30820379 30820261 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
 . . .
 quit
!
crypto isakmp profile vpn1
 vrf vpn1
 ca trust-point bombo
 match identity address 172.16.1.1 255.255.255.255
!
crypto ipsec transform-set vpn1 esp-3des esp-sha-hmac
!
crypto map crypmap 1 ipsec-isakmp
 set peer 172.16.1.1
 set transform-set vpn1
 set isakmp-profile vpn1
 match address 101
!
interface Ethernet1/1
 ip address 172.31.1.1 255.255.0.0
 tag-switching ip
!
interface Ethernet1/2
 ip address 172.18.1.1 255.255.255.0
 crypto map crypmap
!
```

```
ip route 172.16.1.1 255.255.255.255 172.18.1.2
ip route vrf vpn1 10.2.0.0 255.255.0.0 172.18.1.2 global
!
access-list 101 permit ip 10.1.0.0 0.0.255.255 10.2.0.0 0.0.255.255
!
```

IPSec CPE Configuration for VPN1

```
crypto ca trustpoint bombo
 enrollment url http://172.31.68.59:80
 crl optional
 !
crypto ca certificate chain bombo
 certificate 03BF
  308203BD 308202A5 A0030201 02020203 BF300D06 092A8648 86F70D01 01050500
  . . .
  quit
 certificate ca 01
  30820379 30820261 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
  . . .
  quit
!
crypto ipsec transform-set vpn1 esp-3des esp-sha-hmac
!
crypto map vpn1 1 ipsec-isakmp
 set peer 172.18.1.1
 set transform-set vpn1
 match address 101
!
interface FastEthernet1/0
 ip address 172.16.1.1 255.255.255.0
 crypto map vpn1
!
interface FastEthernet1/1
 ip address 10.2.1.1 255.255.0.0
!
access-list 101 permit ip 10.2.0.0 0.0.255.255 10.1.0.0 0.0.255.255
!
```

IPSec Remote Access-to-MPLS VPN Example

The following shows an IPSec remote access-to-MPLS VPN configuration. The configuration maps IPSec tunnels to MPLS VPNs. The IPSec tunnels terminate on a single public-facing interface.

PE Router Configuration

```
aaa new-model
!
aaa group server radius vpn1
 server-private 10.1.1.1 auth-port 1645 acct-port 1646 timeout 5 retransmit 3 key vpn1
!
aaa group server radius vpn2
 server-private 10.1.1.1 auth-port 1645 acct-port 1646 timeout 5 retransmit 3 key vpn2
!
aaa authorization network aaa-list group radius
!
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
ip vrf vpn2
```

```
 rd 101:1
 route-target export 101:1
 route-target import 101:1
!
crypto isakmp profile vpn1-ra
   vrf vpn1
   match identity group vpn1-ra
   client authentication list vpn1
   isakmp authorization list aaa-list
   client configuration address initiate
   client configuration address respond
crypto isakmp profile vpn2-ra
   vrf vpn2
   match identity group vpn2-ra
   client authentication list vpn2
   isakmp authorization list aaa-list
   client configuration address initiate
   client configuration address respond
!
!
crypto ipsec transform-set vpn1 esp-3des esp-sha-hmac
crypto ipsec transform-set vpn2 esp-3des esp-md5-hmac
!
crypto dynamic-map vpn1 1
 set transform-set vpn1
 set isakmp-profile vpn1-ra
 reverse-route
!
crypto dynamic-map vpn2 1
 set transform-set vpn2
 set isakmp-profile vpn2-ra
 reverse-route
!
!
crypto map ra 1 ipsec-isakmp dynamic vpn1
crypto map ra 2 ipsec-isakmp dynamic vpn2
!
interface Ethernet1/1
 ip address 172.17.1.1 255.255.0.0
 tag-switching ip
!
interface Ethernet1/2
 ip address 172.18.1.1 255.255.255.0
 crypto map ra
!
ip local pool vpn1-ra 10.4.1.1 10.4.1.254 group vpn1-ra
ip local pool vpn2-ra 10.4.1.1 10.4.1.254 group vpn2-ra
!
```

Upgrade from Previous Versions of the Cisco Network-Based IPSec VPN Solution

The VRF-Aware IPSec feature in the Cisco network-based IPSec VPN solution release 1.5 requires that you change your existing configurations.

The sample configurations that follow indicate the changes you must make to your existing configurations. These samples include the following:

- Site-to-Site Configuration Upgrade, page 1384
- Remote Access Configuration Upgrade, page 1385

- Combination Site-to-Site and Remote Access Configuration Upgrade, page 1387

Site-to-Site Configuration Upgrade

The following configurations show the changes that are necessary for a site-to-site configuration upgrade from a previous version of the network-based IPSec VPN solution to the Cisco network-based IPSec VPN solution release 1.5:

Previous Version Site-to-Site Configuration

```
crypto isakmp key VPN1 address 172.21.25.74
 crypto isakmp key VPN2 address 172.21.21.74
!
 crypto ipsec transform-set VPN1 esp-des esp-sha-hmac
 crypto ipsec transform-set VPN2 esp-3des esp-sha-hmac
!
 crypto map VPN1 10 ipsec-isakmp
 set peer 172.21.25.74
 set transform-set VPN1
 match address 101
!
 crypto map VPN2 10 ipsec-isakmp
 set peer 172.21.21.74
 set transform-set VPN2
 match address 102
!
 interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
 interface FastEthernet0/0.2
  encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
```

New Version Site-to-Site Configuration

The following is an upgraded version of the same site-to-site configuration to the Cisco network-based IPSec VPN solution release 1.5 solution:

Note You must change to keyrings. The VRF-Aware IPSec feature requires that keys be associated with a VRF if the IKE local endpoint is in the VRF.

```
crypto keyring VPN1-KEYS vrf VPN1
 pre-shared-key address 172.21.25.74 key VPN1
!
 crypto keyring VPN2-KEYS vrf VPN2
  pre-shared-key address 172.21.21.74 key VPN2
!
 crypto ipsec transform-set VPN1 esp-des esp-sha-hmac
 crypto ipsec transform-set VPN2 esp-3des esp-sha-hmac
!
 crypto map VPN1 10 ipsec-isakmp
 set peer 172.21.25.74
  set transform-set VPN1
```

```
 match address 101
!
 crypto map VPN2 10 ipsec-isakmp
 set peer 172.21.21.74
  set transform-set VPN2
 match address 102
!
 interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
 interface FastEthernet0/0.2
 encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
```

Remote Access Configuration Upgrade

The following configurations show the changes that are necessary for a remote access configuration upgrade from a previous version of the network-based IPSec VPN solution to the Cisco network-based IPSec VPN solution release 1.5:

Previous Version Remote Access Configuration

```
crypto isakmp client configuration group VPN1-RA-GROUP
 key VPN1-RA
 pool VPN1-RA
!
 crypto isakmp client configuration group VPN2-RA-GROUP
 key VPN2-RA
 pool VPN2-RA
!
 crypto ipsec transform-set VPN1-RA esp-3des esp-sha-hmac
 crypto ipsec transform-set VPN2-RA esp-3des esp-md5-hmac
!
 crypto dynamic-map VPN1-RA 1
  set transform-set VPN1-RA
 reverse-route
!
 crypto dynamic-map VPN2-RA 1
 set transform-set VPN2-RA
  reverse-route
!
!
 crypto map VPN1 client authentication list VPN1-RA-LIST
 crypto map VPN1 isakmp authorization list VPN1-RA-LIST
 crypto map VPN1 client configuration address initiate
 crypto map VPN1 client configuration address respond
 crypto map VPN1 10 ipsec-isakmp dynamic VPN1-RA
!
 crypto map VPN2 client authentication list VPN2-RA-LIST
 crypto map VPN2 isakmp authorization list VPN2-RA-LIST
 crypto map VPN2 client configuration address initiate
 crypto map VPN2 client configuration address respond
 crypto map VPN2 10 ipsec-isakmp dynamic VPN2-RA
!
 interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
```

```
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
 interface FastEthernet0/0.2
  encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
```

New Version Remote Access Configuration

In the following instance, there is no upgrade; it is recommended that you change to the following configuration:

```
crypto isakmp client configuration group VPN1-RA-GROUP
  key VPN1-RA
  pool VPN1-RA
!
 crypto isakmp client configuration group VPN2-RA-GROUP
 key VPN2-RA
 pool VPN2-RA
!
 crypto isakmp profile VPN1-RA
 match identity group VPN1-RA-GROUP
  client authentication list VPN1-RA-LIST
 isakmp authorization list VPN1-RA-LIST
 client configuration address initiate
 client configuration address respond
!
 crypto isakmp profile VPN2-RA
 match identity group VPN2-RA-GROUP
 client authentication list VPN2-RA-LIST
 isakmp authorization list VPN2-RA-LIST
 client configuration address initiate
 client configuration address respond
!
 crypto ipsec transform-set VPN1-RA esp-3des esp-sha-hmac
 crypto ipsec transform-set VPN2-RA esp-3des esp-md5-hmac
!
 crypto dynamic-map VPN1-RA 1
 set transform-set VPN1-RA
 set isakmp-profile VPN1-RA
 reverse-route
!
 crypto dynamic-map VPN2-RA 1
 set transform-set VPN2-RA
 set isakmp-profile VPN2-RA
 reverse-route
!
 crypto map VPN1 10 ipsec-isakmp dynamic VPN1-RA
!
 crypto map VPN2 10 ipsec-isakmp dynamic VPN2-RA
!
 interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
 interface FastEthernet0/0.2
 encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
```

```
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
```

Combination Site-to-Site and Remote Access Configuration Upgrade

The following configurations show the changes that are necessary for a site-to-site and remote access configuration upgrade from a previous version of the network-based IPSec VPN solution to the Cisco network-based IPSec VPN solution release 1.5:

Previous Version Site-to-Site and Remote Access Configuration

```
crypto isakmp key VPN1 address 172.21.25.74 no-xauth
crypto isakmp key VPN2 address 172.21.21.74 no-xauth
!
 crypto isakmp client configuration group VPN1-RA-GROUP
 key VPN1-RA
 pool VPN1-RA
!
 crypto isakmp client configuration group VPN2-RA-GROUP
 key VPN2-RA
 pool VPN2-RA
!
 crypto ipsec transform-set VPN1 esp-des esp-sha-hmac
 crypto ipsec transform-set VPN2 esp-3des esp-sha-hmac
!
 crypto ipsec transform-set VPN1-RA esp-3des esp-sha-hmac
 crypto ipsec transform-set VPN2-RA esp-3des esp-md5-hmac
!
 crypto dynamic-map VPN1-RA 1
 set transform-set VPN1-RA
 reverse-route
!
 crypto dynamic-map VPN2-RA 1
 set transform-set VPN2-RA
 reverse-route
!
 crypto map VPN1 client authentication list VPN1-RA-LIST
 crypto map VPN1 isakmp authorization list VPN1-RA-LIST
 crypto map VPN1 client configuration address initiate
 crypto map VPN1 client configuration address respond
 crypto map VPN1 10 ipsec-isakmp
 set peer 172.21.25.74
 set transform-set VPN1
 match address 101
 crypto map VPN1 20 ipsec-isakmp dynamic VPN1-RA
!
 crypto map VPN2 client authentication list VPN2-RA-LIST
 crypto map VPN2 isakmp authorization list VPN2-RA-LIST
 crypto map VPN2 client configuration address initiate
 crypto map VPN2 client configuration address respond
 crypto map VPN2 10 ipsec-isakmp
 set peer 172.21.21.74
 set transform-set VPN2
 match address 102
 crypto map VPN2 20 ipsec-isakmp dynamic VPN2-RA
!
 interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
```

```
!
 interface FastEthernet0/0.2
 encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
```

New Version Site-to-Site and Remote Access Configuration

You must upgrade to this configuration:

Note For site-to-site configurations that do not require XAUTH, configure an ISAKMP profile without XAUTH configuration. For remote access configurations that require XAUTH, configure an ISAKMP profile with XAUTH.

```
crypto keyring VPN1-KEYS vrf VPN1
 pre-shared-key address 172.21.25.74 key VPN1
!
 crypto keyring VPN2-KEYS vrf VPN2
 pre-shared-key address 172.21.21.74 key VPN2
!
 crypto isakmp client configuration group VPN1-RA-GROUP
 key VPN1-RA
 pool VPN1-RA
!
 crypto isakmp client configuration group VPN2-RA-GROUP
 key VPN2-RA
 pool VPN2-RA
!
 crypto isakmp profile VPN1
 keyring VPN1-KEYS
 match identity address 172.21.25.74 VPN1
!
 crypto isakmp profile VPN2
 keyring VPN2-KEYS
 match identity address 172.21.21.74 VPN2
!
 crypto isakmp profile VPN1-RA
 match identity group VPN1-RA-GROUP
 client authentication list VPN1-RA-LIST
 isakmp authorization list VPN1-RA-LIST
 client configuration address initiate
 client configuration address respond
!
 crypto isakmp profile VPN2-RA
 match identity group VPN2-RA-GROUP
 client authentication list VPN2-RA-LIST
 isakmp authorization list VPN2-RA-LIST
 client configuration address initiate
 client configuration address respond
!
 crypto ipsec transform-set VPN1 esp-des esp-sha-hmac
 crypto ipsec transform-set VPN2 esp-3des esp-sha-hmac
!
 crypto ipsec transform-set VPN1-RA esp-3des esp-sha-hmac
 crypto ipsec transform-set VPN2-RA esp-3des esp-md5-hmac
!
 crypto dynamic-map VPN1-RA 1
 set transform-set VPN1-RA
 set isakmp-profile VPN1-RA
```

```
 reverse-route
!
 crypto dynamic-map VPN2-RA 1
 set transform-set VPN2-RA
 set isakmp-profile VPN2-RA
 reverse-route
!
 crypto map VPN1 10 ipsec-isakmp
 set peer 172.21.25.74
 set transform-set VPN1
 set isakmp-profile VPN1
 match address 101
 crypto map VPN1 20 ipsec-isakmp dynamic VPN1-RA
!
 crypto map VPN2 10 ipsec-isakmp
 set peer 172.21.21.74
 set transform-set VPN2
 set isakmp-profile VPN2
 match address 102
 crypto map VPN2 20 ipsec-isakmp dynamic VPN2-RA
!
 interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
 interface FastEthernet0/0.2
 encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
```

Additional References

For additional information related to VRF-Aware IPSec, refer to the following references:

Related Documents

Related Topic	Document Title
IPSec configuration tasks	The chapter "Configuring IPSec Network Security" in the *Cisco IOS Security Configuration Guide*, Release 12.2
IPSec commands	The chapter "IPSec Network Security Commands" in the *Cisco IOS Security Command Reference*, Release 12.2 T
IKE Phase 1 and Phase 2, agressive mode, and main mode	The chapter "Configuring Internet Key Exchange Security Protocol" in the *Cisco IOS Security Configuration Guide*, Release 12.2
IKE dead peer detection	*Easy VPN Server*
Additional VPN and MPLS configuration tasks	*Cisco IOS Security Configuration Guide*, Release 12.2

Standards

Standards[1]	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

1. Not all supported standards are listed.

MIBs

MIBs[1]	MIBs Link
• No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

1. Not all supported MIBs are listed.

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

RFCs[1]	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

1. Not all supported RFCs are listed.

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, tools, and lots more. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

New Commands

- **address**
- **ca trust-point**
- **client authentication list**
- **client configuration address**
- **crypto isakmp profile**
- **crypto keyring**
- **crypto map isakmp-profile**
- **initiate-mode**
- **isakmp authorization list**
- **keepalive (isakmp profile)**
- **keyring**
- **key-string**
- **match identity**
- **no crypto xauth**
- **pre-shared-key**
- **quit**
- **rsa-pubkey**
- **self-identity**
- **serial-number**
- **set isakmp-profile**
- **show crypto isakmp key**
- **show crypto isakmp profile**
- **vrf**

Modified Commands

- clear crypto sa
- crypto isakmp peer
- crypto map isakmp-profile
- show crypto dynamic-map
- show crypto ipsec sa
- show crypto isakmp sa
- show crypto map (IPSec)

Glossary

CA—certification authority. CA is an entity that issues digital certificates (especially X.509 certificates) and vouches for the binding between the data items in a certificate.

CLI—command-line-interface. CLI is an interface that allows the user to interact with the operating system by entering commands and optional arguments. The UNIX operating system and DOS provide CLIs.

client—Corresponding IPSec IOS peer of the UUT in the Multi Protocol Label Switching (MPLS) network.

dead peer—IKE peer that is no longer reachable.

DN—Distinguished Name. A DN is the global, authoritative name of an entry in the Open System Interconnection (OSI Directory [X.500]).

FQDN—fully qualified domain name. A FQDN is the full name of a system rather than just its host name. For example, aldebaran is a host name, and aldebaran.interop.com is an FQDN.

FR—Frame Relay. FR is an industry-standard, switch-data-link-layer protocol that handles multiple virtual circuits using high-level data link (HDLC) encapsulation between connected devices. Frame Relay is more efficient than X.25, the protocol for which it generally is considered a replacement.

FVRF—Front Door Virtual Routing and Forwarding (VRF) repository. FVRF is the VRF used to route the encrypted packets to the peer.

IDB—Interface descriptor block. An IDB subblock is an area of memory that is private to an application. This area stores private information and states variables that an application wants to associate with an IDB or an interface. The application uses the IDB to register a pointer to its subblock, not to the contents of the subblock itself.

IKE—Internet Key Exchange. IKE establishes a shared security policy and authenticates keys for services (such as IPSec) that require keys. Before any IPSec traffic can be passed, each router, firewall, and host must verify the identity of its peer. This can be done by manually entering preshared keys into both hosts or by a CA service.

IKE keepalive—Bidirectional mechanism for determining the liveliness of an IKE peer.

IPSec—Security protocol for IP.

IVRF—Inside Virtual Routing and Forwarding. IVRF is the VRF of the plaintext packets.

MPLS—Multiprotocol Label Switching. MPLS is a switching method that forwards IP traffic using a label. This label instructs the routers and the switches in the network where to forward the packets based on preestablished IP routing information.

RSA—Rivest, Shamir, and Adelman are the inventors of the RSA technique. The RSA technique is a public-key cryptographic system that can be used for encryption and authentication.

SA—Security Association. SA is an instance of security policy and keying material applied to a data flow.

VPN—Virtual Private Network. A VPN enables IP traffic to travel securely over a public TCP or IP network by encrypting all traffic from one network to another. A VPN uses "tunneling" to encrypt all information at the IP level.

VRF—Virtual Route Forwarding. VRF is A VPN routing and forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE router.

XAUTH—Extended authentication. XAUTH is an optional exchange between IKE Phase 1 and IKE Phase 2, in which the router demands additional authentication information in an attempt to authenticate the actual user (as opposed to authenticating the peer).

Note Refer to the *Internetworking Terms and Acronyms* for terms not included in this glossary.

Part 5: PKI

Implementing and Managing PKI Features Roadmap

This roadmap lists the features documented in the *Cisco IOS Security Configuration Guide* and maps them to the modules in which they appear.

Roadmap History

This roadmap was first published on May 2, 2005, and last updated on May 2, 2005.

Feature and Release Support

Table 56 lists public key infrastructure (PKI) feature support for the following Cisco IOS software release trains:

- Cisco IOS Releases 12.2T, 12.3, and 12.3T

Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table. *Not all features may be supported in your Cisco IOS software release*.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Note Table 56 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 56 Supported PKI Features

Release	Feature Name	Feature Description	Where Documented
Cisco IOS Releases 12.2T, 12.3, and 12.3T			
12.3(14)T	Administrative Secure Device Provisioning Introducer	This feature allows you to act as an administrative introducer to introduce a device into a PKI network and then provide a username as the device name for the record locator in the AAA database.	"Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI"
12.3(14)T	Persistent Self-Signed Certificates	This feature allows users the HTTPS server to generate and save a self-signed certificate in the router's startup configuration. Thus, future SSL handshakes between the client and the HTTPS server can use the same self-signed certificate without user intervention.	"Configuring Certificate Enrollment for a PKI"
12.3(14)T	Secure Device Provisioning Certificate-Based Authorization	This feature allows certificates issued by other authority (CA) servers to be used for SDP introductions.	"Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI"
12.3(14)T	Subordinate Certificate Server	This enhancement allows you to configure a subordinate certificate server to grant all or certain SCEP or manual certificate requests.	"Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment"
12.3(14)T	USB Storage	This feature explains how to store RSA keys on a device external to the router via a USB eToken. The SmartCard technology (which is owned by Aladdin Knowledge Systems) in a USB key form factor (also referred to as a USB eToken) provides secure configuration distribution and allows users to store PKI credentials, such as RSA keys, for deployment.	"Storing PKI Credentials External to the Router"
12.3(11)T	The Certificate Server Auto Archive enhancement	This enhancement enables the CA certificate and CA key to be backed up automatically just once after they are generated by the certificate server. As a result, it is not necessary to generate an exportable CA key if CA backup is desirable.	"Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment"
12.3(11)T	PKI AAA Authorization Using the Entire Subject Name	This feature provides users with the ability to query the AAA server using the entire subject name from the certificate as a unique AAA username.	"Configuring Revocation and Authorization of Certificates in a PKI"
12.3(11)T	PKI Status	This enhancement added the **status** keyword to the **show crypto pki trustpoints** command, which allows you to view the current status of the trustpoint. Prior to this enhancement, you had to issue the **show crypto pki certificates** and the **show crypto pki timers** commands for the current status.	"Configuring Certificate Enrollment for a PKI" and "Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment"
12.3(11)T	Reenroll Using Existing Certificates	This feature allows users to reenroll a router with a Cisco IOS CA via existing certificates from a third-party vendor CA.	"Configuring Certificate Enrollment for a PKI"
12.3(8)T	Easy Secure Device Deployment	This feature introduces support for SDP (formerly called EzSDD), which offers a web-based enrollment interface that enables network administrators to deploy new devices in large networks.	"Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI"

Table 56 Supported PKI Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.3(8)T	Easy Secure Device Deployment AAA Integration	This feature integrates an external AAA database, allowing the introducer to be authenticated against a AAA database instead of having to use the enable password of the local Cisco certificate server.	"Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI"
12.3(7)T	The Certificate Server Registration Authority (RA) Mode enhancement	A certificate server can be configured to run in RA mode.	"Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment"
12.3(7)T	The "crypto pki" commands should be a synonym for "crypto ca" commands	This enhancement changes all commands that begin as "crypto ca" to "crypto pki." Although the router will still accept crypto ca, all output will be read back as crypto pki.	All modules that contain crypto ca commands.
12.3(7)T	Key Rollover for Certificate Renewal	This feature allows the certificate renewal request to be made before the certificate expires and retains the old key and certificate until the new certificate is available.	"Configuring Certificate Enrollment for a PKI"
12.3(7)T	PKI: Query Multiple Servers During Certificate Revocation Check	This feature introduces the ability for Cisco IOS software to make multiple attempts to retrieve the CRL, allowing operations to continue when a particular server is not available. In addition, the ability to override the CDPs in a certificate with a manually configured CDP has been introduced. Manually overriding the CDPs in a certificate can be advantageous when a particular server is unavailable for an extended period of time. The certificate's CDPs can be replaced with a URL or directory specification without reissuing all of the certificates that contain the original CDP.	"Configuring Revocation and Authorization of Certificates in a PKI"
12.3(7)T	Protected Private Key Storage	This feature allows a user to encrypt and lock the RSA private keys that are used on a Cisco IOS router, thereby, preventing unauthorized use of the private keys.	"Deploying RSA Keys Within a PKI"
12.3(4)T	Import of RSA Key Pair and Certificates in PEM Format	This feature allows customers to use PEM-formatted files to import or export RSA key pairs. PEM-formatted files allow customers to directly use existing RSA key pairs on their Cisco IOS routers instead of generating new keys. Also, customers can issue certificate requests and receive issued certificates in PEM-formatted files.	"Deploying RSA Keys Within a PKI" and "Configuring Certificate Enrollment for a PKI"
12.3(4)T	Using Certificate ACLs to Ignore Revocation Check and Expired Certificates	This feature allows a certificate that meets specified criteria to be accepted regardless of the validity period of the certificate, or if the certificate meets the specified criteria, revocation checking does not have to be performed. Certificate ACLs are used to specify the criteria that the certificate must meet to be accepted or to avoid revocation checking. In addition, if AAA communication is protected by a certificate, this feature provides for the AAA checking of the certificate to be ignored.	"Configuring Revocation and Authorization of Certificates in a PKI"

Table 56 Supported PKI Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.3(4)T	Cisco IOS Certificate Server	This feature introduces support for the Cisco IOS CS, which offers users a CA that is directly integrated with Cisco IOS software to more easily deploy basic PKI networks.	"Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment"
12.3(4)T	Direct HTTP Enrollment with CA Servers	This feature allows users to configure an enrollment profile if their CA server does not support SCEP and they do not want to use an RA as a proxy. The enrollment profile allows users to send HTTP requests directly to the CA server instead of the RA proxy.	"Configuring Certificate Enrollment for a PKI"
12.3(2)T	Online Certificate Status Protocol (OCSP)	This feature allows users to enable OCSP instead of CRLs to check certificate status. Unlike CRLs, which provide only periodic certificate status, OCSP can provide timely information regarding the status of a certificate.	"Configuring Revocation and Authorization of Certificates in a PKI"
12.3(1)	PKI Integration with AAA Server	This feature provides additional scalability for authorization by generating a AAA username from the certificate presented by the peer. A AAA server is queried to determine whether the certificate is authorized for use by the internal component. The authorization is indicated by a component-specified label that must be present in the AV pair for the user.	"Configuring Revocation and Authorization of Certificates in a PKI"
12.2(15)T	Certificate Security Attribute-Based Access Control	Under the IPSec protocol, CA interoperability permits Cisco IOS devices and a CA to communicate so that the Cisco IOS device can obtain and use digital certificates from the CA. Certificates contain several fields that are used to determine whether a device or user is authorized to perform a specified action. This feature adds fields to the certificate that allow specifying an ACL, to create a certificate-based ACL.	"Configuring Revocation and Authorization of Certificates in a PKI"
12.2(15)T	Exporting and Importing RSA Keys	This feature allows you to transfer security credentials between devices by exporting and importing RSA keys. The key pair that is shared between two devices will allow one device to immediately and transparently take over the functionality of the other router.	"Deploying RSA Keys Within a PKI"
12.2(15)T	Multiple-Tier CA Hierarchy	This enhancement enables users to set up a PKI in a hierarchical framework to support multiple CAs. Within a hierarchical PKI, all enrolled peers can validate the certificate of one another as long as the peers share a trusted root CA certificate or a common subordinate CA.	"Configuring Certificate Enrollment for a PKI"
12.2(13)T	Manual Certificate Enrollment (TFTP Cut-and-Paste)	This feature allows users to generate a certificate request and accept CA certificates as well as the router's certificates via a TFTP server or manual cut-and-paste operations.	"Configuring Certificate Enrollment for a PKI"
12.2(8)T	Certificate Autoenrollment	This feature introduces certificate autoenrollment, which allows the router to automatically request a certificate from the CA that is using the parameters in the configuration.	"Configuring Certificate Enrollment for a PKI"

Table 56 Supported PKI Features (continued)

Release	Feature Name	Feature Description	Where Documented
12.2(8)T	Certificate Enrollment Enhancements	This feature introduces five new **crypto ca trustpoint** subcommands that provide new options for certificate requests and allow users to specify fields in the configuration instead of having to go through prompts.	"Configuring Certificate Enrollment for a PKI"
12.2(8)T	Multiple RSA Key Pair Support	This feature allows a user to configure a router to have multiple RSA key pairs. Thus, the Cisco IOS software can maintain a different key pair for each identity certificate.	"Deploying RSA Keys Within a PKI"
12.2(8)T	Trustpoint CLI	This feature introduces the **crypto ca trustpoint** command, which adds support for trustpoint CAs.	"Configuring Certificate Enrollment for a PKI"

Cisco IOS PKI Overview: Understanding and Planning a PKI

Cisco IOS public key infrastructure (PKI) provides certificate management to support security protocols such as IP Security (IPSec), secure shell (SSH), and secure socket layer (SSL).

This module identifies and describes concepts that are needed to understand, plan for, and implement a PKI.

Module History

This module was first published on May 2, 2005, and last updated on May 2, 2005.

Contents

- Information About Cisco IOS PKI, page 1403
- Planning for a PKI, page 1407
- Where to Go Next, page 1408
- Additional References, page 1408
- Glossary, page 1409

Information About Cisco IOS PKI

Before implementing a basic PKI, you should understand the following concepts:

- What Is Cisco IOS PKI?, page 1404
- RSA Keys Overview, page 1405
- What Are CAs?, page 1405
- Certificate Enrollment: How It Works, page 1406
- Certificate Revocation: Why It Occurs, page 1407

What Is Cisco IOS PKI?

A PKI is composed of the following entities:

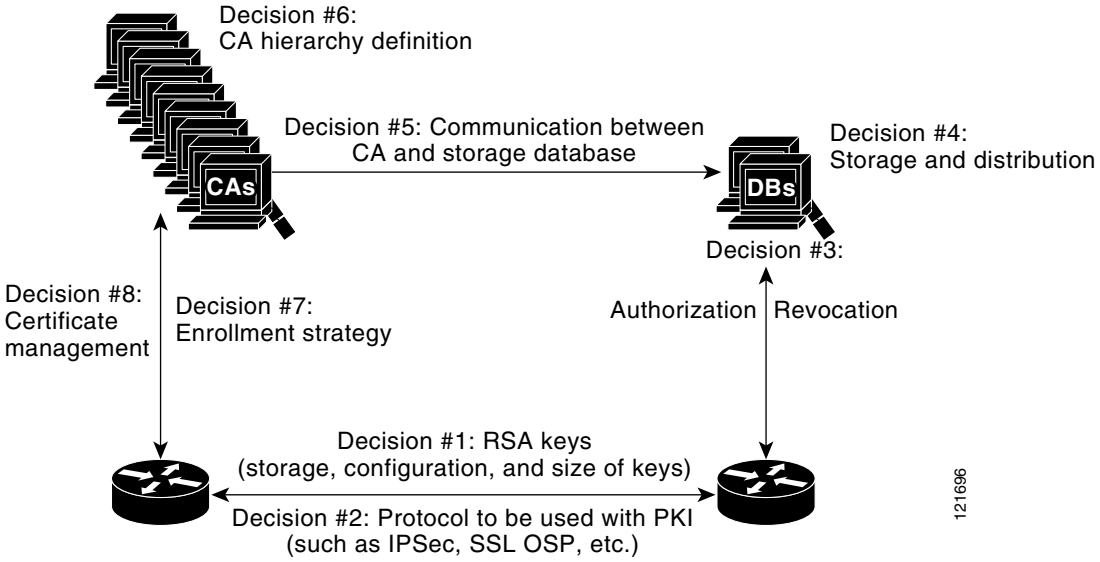
- Peers communicating on a secure network
- At least one certification authority (CA) that grants and maintains certificates
- Digital certificates, which contain information such as the certificate validity period, peer identity information, encryptions keys that are used for secure communications, and the signature of the issuing CA
- An optional registration authority (RA) to offload the CA by processing enrollment requests
- A distribution mechanism (such as Lightweight Directory Access Protocol [LDAP] or HTTP) for certificate revocation lists (CRLs)

PKI provides customers with a scalable, secure mechanism for distributing, managing, and revoking encryption and identity information in a secured data network. Every entity (a person or a device) participating in the secured communicated is enrolled in the PKI in a process where the entity generates an Rivest, Shamir, and Adelman (RSA) key pair (one private key and one public key) and has their identity validated by a trusted entity (also known as a CA or trustpoint).

After each entity enrolls in a PKI, every peer (also known as an end host) in a PKI is granted a digital certificate that has been issued by a CA. When peers must negotiate a secured communication session, they exchange digital certificates. Based on the information in the certificate, a peer can validate the identity of another peer and establish an encrypted session with the public keys contained in the certificate.

Although you can plan for and set up your PKI in a number of different ways, Figure 97 shows the major components that make up a PKI and suggests an order in which each decision within a PKI can be made. Figure 97 is a suggested approach; you can choose to set up your PKI from a different perspective.

Figure 97 Deciding How to Set Up Your PKI



RSA Keys Overview

An RSA key pair consists of a public key and a private key. When setting up your PKI, you must include the public key in the certificate enrollment request. After the certificate has been granted, the public key will be included in the certificate so that peers can use it to encrypt data that is sent to the router. The private key is kept on the router and used both to decrypt the data sent by peers and to digitally sign transactions when negotiating with peers.

RSA key pairs contain a key modulus value. The modulus determines the size of the RSA key. The larger the modulus, the more secure the RSA key. However, keys with large modulus values take longer to generate, and encryption and decryption operations take longer with larger keys.

What Are CAs?

A CA, also known as a trustpoint, manages certificate requests and issues certificates to participating network devices. These services (managing certificate requests and issuing certificates) provide centralized key management for the participating devices and are explicitly trusted by the receiver to validate identities and to create digital certificates. Before any PKI operations can begin, the CA generates its own public key pair and creates a self-signed CA certificate; thereafter, the CA can sign certificate requests and begin peer enrollment for the PKI.

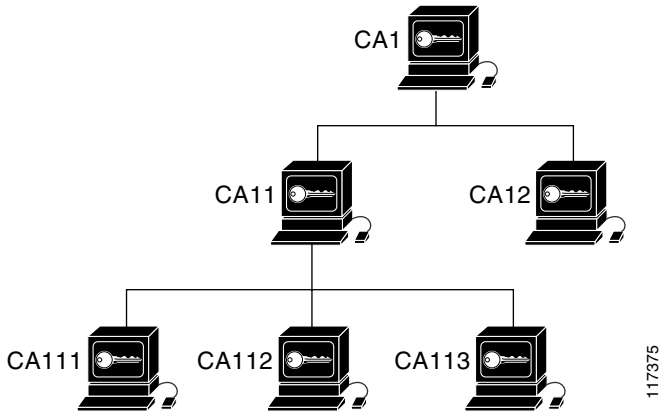
You can use a CA provided by a third-party CA vendor, or you can use an "internal" CA, which is the Cisco IOS Certificate Server.

Hierarchical PKI: Multiple CAs

PKI can be set up in a hierarchical framework to support multiple CAs. At the top of the hierarchy is a root CA, which holds a self-signed certificate. The trust within the entire hierarchy is derived from the RSA key pair of the root CA. The subordinate CAs within the hierarchy can be enrolled with either the root CA or with another subordinate CA. These enrollment options are how multiple tiers of CAs are configured. Within a hierarchical PKI, all enrolled peers, can validate the certificate of one another if the peers share a trusted root CA certificate or a common subordinate CA.

Figure 98 shows the enrollment relationships among CAs within a three-tiered hierarchy.

Figure 98 Three-Tiered CA Hierarchy Sample Topology



Each CA corresponds to a trustpoint. For example, CA11 and CA12 are subordinate CAs, holding CA certificates that have been issued by CA1; CA111, CA112, and CA113 are also subordinate CAs, but their CA certificates have been issued by CA11.

When to Use Multiple CAs

Multiple CAs provide users with added flexibility and reliability. For example, subordinate CAs can be placed in branch offices while the root CA is at the office headquarters. Also, different granting policies can be implemented per CA, so you can set up one CA to automatically grant certificate requests while another CA within the hierarchy requires each certificate request to be manually granted.

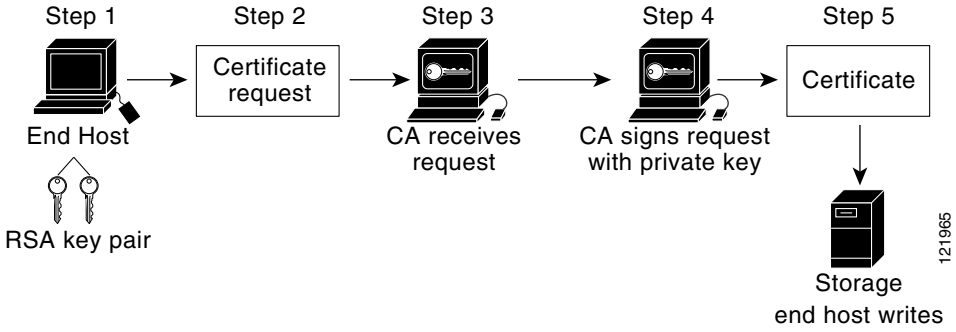
Scenarios in which at least a two-tier CA is recommended are as follows:

- Large and very active networks in which a large number of certificates are revoked and reissued. A multiple tier CA helps to control the size of the CRLs.
- When online enrollment protocols are used, the root CA can be kept offline with the exception of issuing subordinate CA certificates. This scenario provides added security for the root CA.

Certificate Enrollment: How It Works

Certificate enrollment is the process of obtaining a certificate from a CA. Each end host that wants to participate in the PKI must obtain a certificate. Certificate enrollment occurs between the end host requesting the certificate and the CA. Figure 99 and the following steps describe the certificate enrollment process.

Figure 99 Certificate Enrollment Process



1. The end host generates an RSA key pair.
2. The end host generates a certificate request and forwards it to the CA (or the RA, if applicable).
3. The CA receives the certificate enrollment request, and, depending on your network configuration, one of the following options occurs:
   c. Manual intervention is required to approve the request.
   d. The end host is configured to automatically request a certificate from the CA. Thus, operator intervention is no longer required at the time the enrollment request is sent to the CA server.

Note If you configure the end host to automatically request certificates from the CA, you should have an additional authorization mechanism.

4. After the request is approved, the CA signs the request with its private key and returns the completed certificate to the end host.
5. The end host writes the certificate to a storage area such as NVRAM.

Certificate Enrollment Via Secure Device Provisioning

Secure Device Provisioning (SDP) is a web-based certificate enrollment interface that can be used to easily deploy PKI between two end devices, such as a Cisco IOS client and a Cisco IOS certificate server.

SDP (also referred to as Trusted Transitive Introduction [TTI]) is a communication protocol that provides a bidirectional introduction between two end entities, such as a new network device and a Virtual Private Network (VPN). SDP involves the following three entities:

- Introducer—A mutually trusted device that introduces the petitioner to the registrar. The introducer can be a device user, such as a system administrator.
- Petitioner—A new device that is joined to the secure domain.
- Registrar—A certificate server or other server that authorizes the petitioner.

SDP is implemented over a web browser in three phases—welcome, introduction, and completion. Each phase is shown to the user via a web page. For more information on each phase and how SDP works, see the "Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI" module.

Certificate Revocation: Why It Occurs

After each participant has successfully enrolled in the PKI, the peers are ready to begin negotiations for a secure connection with each other. Thus, the peers present their certificates for validation followed by a revocation check. After the peer verifies that the other peer's certificate was issued by an authenticated CA, the CRL or Online Certificate Status Protocol (OCSP) server is checked to ensure that the certificate has not been revoked by the issuing CA. The certificate usually contains a certificate distribution point (CDP) in the form of a URL. Cisco IOS software uses the CDP to locate and retrieve the CRL. If the CDP server does not respond, the Cisco IOS software reports an error, which may result in the peer's certificate being rejected.

Planning for a PKI

Planning for a PKI requires evaluating the requirements and expected use for each of the PKI components shown in Figure 97. It is recommended that you (or the network administrator) thoroughly plan the PKI before beginning any PKI configuration.

Although there are a number of approaches to consider when planning the PKI, this document begins with peer-to-peer communication and proceeds as shown in Figure 97. However you or the network administrator choose to plan the PKI, understand that certain decisions influence other decisions within the PKI. For example, the enrollment and deployment strategy could influence the planned CA hierarchy. Thus, it is important to understand how each component functions within the PKI and how certain component options are dependent upon decisions made earlier in the planning process.

Where to Go Next

As suggested in Figure 97, you begin to configure a PKI by setting up and deploying RSA keys. For more information, see the module "Deploying RSA Keys Within a PKI."

Additional References

The following sections provide references related to Cisco IOS PKI.

Related Documents

Related Topic	Document Title
PKI commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*, Release 12.4
Certificate enrollment: supported methods, enrollment profiles, configuration tasks	"Configuring Certificate Enrollment for a PKI" module
Certificate revocation and authorization: configuration tasks	"Configuring Revocation and Authorization of Certificates in a PKI" module
Cisco IOS certificate server overview information and configuration tasks	"Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment" module
Secure Device Provisioning: functionality overview and configuration tasks	"Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI" module
Storing RSA keys and certificates on a USB eToken	"Storing PKI Credentials External to the Router" module

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2459	*Internet X.509 Public Key Infrastructure Certificate and CRL Profile*
RFC 2511	*Internet X.509 Certificate Request Message Format*
RFC 2527	*Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework*
RFC 2528	*Internet X.509 Public Key Infrastructure*
RFC 2559	*Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv2*
RFC 2560	*X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*
RFC 2585	*Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP*
RFC 2587	*Internet X.509 Public Key Infrastructure LDAPv2 Schema*
RFC 2875	*Diffie-Hellman Proof-of-Possession Algorithms*
RFC 3029	*Internet X.509 Public Key Infrastructure Data Validation and Certification Server Protocols*

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Glossary

CDP—certificate distribution point. Field within a digital certificate containing information that describes how to retrieve the CRL for the certificate. The most common CDPs are HTTP and LDAP URLs. A CDP may also contain other types of URLs or an LDAP directory specification. Each CDP contains one URL or directory specification.

certificates—Electronic documents that bind a user's or device's name to its public key. Certificates are commonly used to validate a digital signature.

CRL—certificate revocation list. Electronic document that contains a list of revoked certificates. The CRL is created and digitally signed by the CA that originally issued the certificates. The CRL contains dates for when the certificate was issued and when it expires. A new CRL is issued when the current CRL expires.

CA—certification authority. Service responsible for managing certificate requests and issuing certificates to participating IPSec network devices. This service provides centralized key management for the participating devices and is explicitly trusted by the receiver to validate identities and to create digital certificates.

peer certificate—Certificate presented by a peer, which contains the peer's public key and is signed by the trustpoint CA.

PKI—public key infrastructure. System that manages encryption keys and identity information for components of a network that participate in secured communications.

RA—registration authority. Server that acts as a proxy for the CA so that CA functions can continue when the CA is offline. Although the RA is often part of the CA server, the RA could also be an additional application, requiring an additional device to run it.

RSA keys—Public key cryptographic system developed by Ron Rivest, Adi Shamir, and Leonard Adleman. An RSA key pair (a public and a private key) is required before you can obtain a certificate for your router.

Note Refer to *Internetworking Terms and Acronyms* for terms not included in this glossary.

Deploying RSA Keys Within a PKI

This module explains how to set up and deploy Rivest, Shamir, and Adelman (RSA) keys within a public key infrastructure (PKI). An RSA key pair (a public and a private key) is required before you can obtain a certificate for your router; that is, the end host must generate a pair of RSA keys and exchange the public key with the certification authority (CA) to obtain a certificate and enroll in a PKI.

Module History

This module was first published on May 2, 2005, and last updated on May 2, 2005.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all features. To find information about feature support and configuration, use the "Feature Information for RSA Keys Within a PKI" section on page 1429.

Contents

- Prerequisites for Configuring RSA Keys for a PKI, page 1411
- Information About RSA Keys Configuration, page 1412
- How to Set Up and Deploy RSA Keys Within a PKI, page 1414
- Configuration Examples for RSA Key Pair Deployment, page 1423
- Where to Go Next, page 1428
- Additional References, page 1428
- Feature Information for RSA Keys Within a PKI, page 1429

Prerequisites for Configuring RSA Keys for a PKI

- Before setting up and deploying RSA keys for a PKI, you should be familiar with the module "Cisco IOS PKI Overview: Understanding and Planning a PKI."
- As of Cisco IOS Release 12.3(7)T, all commands that begin as "crypto ca" have been changed to begin as "crypto pki." Although the router will still accept crypto ca commands, all output will be read back as crypto pki.

Information About RSA Keys Configuration

To deploy RSA keys within a PKI, you should understand the following concepts:

- RSA Keys Overview, page 1412
- Reasons to Store Multiple RSA Keys on a Router, page 1412
- Benefits of Exportable RSA Keys, page 1413
- Passphrase Protection While Importing and Exporting RSA Keys, page 1413

RSA Keys Overview

An RSA key pair consists of a public key and a private key. When setting up your PKI, you must include the public key in the certificate enrollment request. After the certificate has been granted, the public key will be included in the certificate so that peers can use it to encrypt data that is sent to the router. The private key is kept on the router and used both to decrypt the data sent by peers and to digitally sign transactions when negotiating with peers.

RSA key pairs contain a key modulus value. The modulus determines the size of the RSA key. The larger the modulus, the more secure the RSA key. However, keys with large modulus values take longer to generate, and encryption and decryption operations take longer with larger keys.

If you want a modulus value between 512 and 1024, enter an integer value that is a multiple of 64. If you want a value higher than 1024, enter 1536 or 2048. If you enter a value greater than 512, key generation may take a minute or longer.

Usage RSA Keys Versus General-Purpose RSA Keys

There are two mutually exclusive types of RSA key pairs—usage keys and general-purpose keys. When you generate RSA key pairs (via the **crypto key generate rsa** command), you will be prompted to select either usage keys or general-purpose keys.

Usage RSA Keys

Usage keys consist of two RSA key pairs—one RSA key pair is generated and used for encryption and one RSA key pair is generated and used for signatures. With usage keys, each key is not unnecessarily exposed. (Without usage keys, one key is used for both authentication methods, increasing the exposure of that key.)

General-Purpose RSA Keys

General-purpose keys consist of only one RSA key pair that used for both encryption and signatures. General-purpose key pairs are used more frequently than usage key pairs.

Reasons to Store Multiple RSA Keys on a Router

Configuring multiple RSA key pairs allows the Cisco IOS software to maintain a different key pair for each CA with which it is dealing or the software cab maintain multiple key pairs and certificates with the same CA. Thus, the Cisco IOS software can match policy requirements for each CA without compromising the requirements specified by the other CAs, such as key length, key lifetime, and general-purpose versus usage keys.

Named key pairs (which are specified via the **label** *key-label* option) allow you to have multiple RSA key pairs, enabling the Cisco IOS software to maintain a different key pair for each identity certificate.

Benefits of Exportable RSA Keys

Caution Exportable RSA keys should be carefully evaluated before use because using exportable RSA keys introduces the risk that these keys might be exposed.

Any existing RSA keys are NOT exportable. New keys are generated as nonexportable by default. It is not possible to convert an existing nonexportable key to an exportable key.

As of Cisco IOS Release 12.2(15)T, users can share the private RSA key pair of a router with standby routers, therefore transferring the security credentials between networking devices. The key pair that is shared between two routers will allow one router to immediately and transparently take over the functionality of the other router. If the main router were to fail, the standby router could be dropped into the network to replace the failed router without the need to regenerate keys, reenroll with the CA, or manually redistribute keys.

Exporting and importing an RSA key pair also enables users to place the same RSA key pair on multiple routers so that all management stations using Secure Shell (SSH) can be configured with a single public RSA key.

Exportable RSA Keys in PEM-Formatted Files

Using privacy-enhanced mail (PEM)-formatted files to import or export RSA keys can be helpful for customers who are running Cisco IOS software Release 12.3(4)T or later and who are using secure socket layer (SSL) or SSH applications to manually generate RSA key pairs and import the keys back into their PKI applications. PEM-formatted files allow customers to directly use existing RSA key pairs on their Cisco IOS routers instead of generating new keys.

Passphrase Protection While Importing and Exporting RSA Keys

You have to include a passphrase to encrypt the PKCS12 file or the PEM file that will be exported, and when the PKCS12 or PEM file is imported, the same passphrase has to be entered to decrypt it. Encrypting the PKCS12 or PEM file when it is being exported, deleted, or imported protects the file from unauthorized access and use while it is being transported or stored on an external device.

The passphrase can be any phrase that is at least eight characters in length; it can include spaces and punctuation, excluding the question mark (?), which has special meaning to the Cisco IOS parser.

How to Convert an Exportable RSA Key Pair to a Nonexportable RSA Key Pair

Passphrase protection protects the external PKCS12 or PEM file from unauthorized access and use. To prevent an RSA key pair from being exported, it must be labeled “nonexportable.” To convert an exportable RSA key pair into a nonexportable key pair, the key pair must be exported and then reimported without specifying the “exportable” keyword.

How to Set Up and Deploy RSA Keys Within a PKI

This section contains the following procedures:

- Generating an RSA Key Pair, page 1414
- Generating and Storing Multiple RSA Key Pairs, page 1415
- Exporting and Importing RSA Keys, page 1416
- Encrypting and Locking Private Keys on a Router, page 1419
- Removing RSA Key Pair Settings, page 1422

Generating an RSA Key Pair

Perform this task to manually generate an RSA key pair.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto key generate rsa** {**general-keys** | **usage-keys**} [**label** *key-label*] [**modulus** *modulus-size*] [**exportable**]
4. **exit**
5. **show crypto key mypubkey rsa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto key generate rsa** {**general-keys** | **usage-keys**} [**label** *key-label*] [**modulus** *modulus-size*] [**exportable**] **Example:** `Router(config)# crypto key generate rsa general-keys modulus 360`	Generates RSA key pairs. • If a *key-label* argument is not specified, the default value, which is the fully qualified domain name (FQDN) of the router, is used.

	Command or Action	Purpose
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 5	**show crypto key mypubkey rsa** **Example:** `Router# show crypto key mypubkey rsa`	(Optional) Displays the RSA public keys of your router. This step allows you to verify that the RSA key pair has been successfully generated.

What to Do Next

After you have successfully generated an RSA key pair, you can proceed to any of the additional tasks in this module to generate additional RSA key pairs, perform export and import of RSA key pairs, or configure additional security parameters for the RSA key pair (such as encrypting or locking the private key).

Generating and Storing Multiple RSA Key Pairs

Perform this task to configure the router to generate and store multiple RSA key pairs and associate the key pairs with a trustpoint.

A trustpoint (also known as a CA) manages certificate requests and issues certificates to participating network devices. These services provide centralized key management for the participating devices and are explicitly trusted by the receiver to validate identities and to create digital certificates. Before any PKI operations can begin, the CA generates its own public key pair and creates a self-signed CA certificate; thereafter, the CA can sign certificate requests and begin peer enrollment for the PKI.

Prerequisites

You must have already generated an RSA key pair as shown in the task "Generating an RSA Key Pair."

SUMMARY STEPS

1. **crypto pki trustpoint** *name*
2. **rsakeypair** *key-label* [*key-size* [*encryption-key-size*]]
3. **exit**
4. **exit**
5. **show crypto key mypubkey rsa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint fancy-ca`	Creates a trustpoint and enters ca-trustpoint configuration mode.
Step 2	**rsakeypair** *key-label* [*key-size* [*encryption-key-size*]] **Example:** `Router(ca-trustpoint)# rsakeypair fancy-keys`	Specifies the key pair that is to be used with the trustpoint. • Specify the *key-size* argument for generating the key and specify the *encryption-key-size* argument to request separate encryption, signature keys, and certificates.
Step 3	**exit** **Example:** `Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 5	**show crypto key mypubkey rsa** **Example:** `Router# show crypto key mypubkey rsa`	(Optional) Displays the RSA public keys of your router. This step allows you to verify that the RSA key pair has been successfully generated.

Exporting and Importing RSA Keys

This section contains the following tasks that can be used for exporting and importing RSA keys. Whether you choose to using PKCS12 files or PEM files, you exportable RSA keys allows you to use existing RSA keys on Cisco IOS routers instead of having to generate new RSA keys if the main router were to fail.

- Exporting and Importing RSA Keys in PKCS12 Files, page 1416
- Exporting and Importing RSA Keys in PEM-Formatted Files, page 1418

Exporting and Importing RSA Keys in PKCS12 Files

Exporting and importing RSA key pairs enables users to transfer security credentials between devices. The key pair that is shared between two devices allows one device to immediately and transparently take over the functionality of the other router.

Prerequisites for Exporting and Importing RSA Key in PKCS12 Files

You must generate an RSA key pair and mark it "exportable" as specified in the task "Generating an RSA Key Pair."

Restrictions for Exporting and Importing RSA Keys in PKCS12 Files

- You cannot export RSA keys that existed on the router before your system was upgraded to Cisco IOS Release 12.2(15)T or later. You have to generate new RSA keys and label them as "exportable" after you upgrade the Cisco IOS software.
- When you import a PKCS12 file that was generated by a third-party application, the PKCS12 file must include a CA certificate.
- If you want reexport an RSA key pair after you have already exported the key pair and imported them to a target router, you must specify the **exportable** keyword when you are importing the RSA key pair.

SUMMARY STEPS

1. **crypto pki trustpoint** *name*
2. **rsakeypair** *key-label* [*key-size* [*encryption-key-size*]]
3. **exit**
4. **crypto pki export** *trustpointname* **pkcs12** *destination-url passphrase*
5. **crypto pki import** *trustpointname* **pkcs12** *source-url passphrase*
6. **exit**
7. **show crypto key mypubkey rsa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint darla-ca`	Creates the trustpoint name that it to be associated with the RSA key pair and enters ca-trustpoint configuration mode.
Step 2	**rsakeypair** *key-label* [*key-size* [*encryption-key-size*]] **Example:** `Router(ca-trustpoint)# rsakeypair darla-keys`	Specifies the key pair that is to be used with the trustpoint.
Step 3	**exit** **Example:** `Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode.
Step 4	**crypto pki export** *trustpointname* **pkcs12** *destination-url passphrase* **Example:** `Router(config)# crypto pki export darla-ca pkcs12 tftp://tftpserver/darla-keys PASSWORD`	Exports the RSA keys via the trustpoint name. **Note** You can export the trustpoint using any of the following file system types: flash, FTP, null, NVRAM, RCP, SCP, system, TFTP, Webflash, Xmodem, or Ymodem

	Command or Action	Purpose
Step 5	**crypto pki import** *trustpointname* **pkcs12** *source-url passphrase* **Example:** `Router(config)# crypto pki import darla-ca pkcs12 tftp://tftpserver/darla-keys PASSWORD`	Imports the RSA keys to the target router.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 7	**show crypto key mypubkey rsa** **Example:** `Router# show crypto key mypubkey rsa`	(Optional) Displays the RSA public keys of your router.

Exporting and Importing RSA Keys in PEM-Formatted Files

Perform this task to export or import RSA key pairs in PEM files.

Prerequisites for Exporting and Importing RSA Keys in PEM-Formatted Files

You must generate an RSA key pair and mark it "exportable" as specified in the task "Generating an RSA Key Pair."

Restrictions for Exporting and Importing RSA Keys in PEM Formatted Files

You cannot export and import RSA keys that were generated without an exportable flag before your system was upgraded to Cisco IOS Release 12.3(4)T or later. You have to generate new RSA keys after you upgrade the Cisco IOS software.

SUMMARY STEPS

1. **crypto key generate rsa** {**usage-keys** | **general-keys**} **label** *key-label* [**exportable**]
2. **crypto key export rsa** *key-label* **pem** {**terminal** | **url** *url*} {**3des** | **des**} *passphrase*
3. **crypto key import rsa** *key-label* **pem** [**usage-keys**] {**terminal** | **url** *url*} [**exportable**] *passphrase*
4. **exit**
5. **show crypto key mypubkey rsa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto key generate rsa** {**usage-keys** | **general-keys**} **label** *key-label* [**exportable**] **Example:** `Router(config)# crypto key generate rsa general-keys label mykey exportable`	Generates RSA key pairs. To use PEM files, the RSA key pair must be labeled exportable.
Step 2	**crypto key export rsa** *key-label* **pem** {**terminal** | **url** *url*} {**3des** | **des**} *passphrase* **Example:** `Router(config)# crypto key export rsa mycs pem url nvram: 3des PASSWORD`	Exports the generated RSA key pair. **Tip** Be sure to keep the PEM file safe. For example, you may want to store it on another backup router.
Step 3	**crypto key import rsa** *key-label* **pem** [**usage-keys**] {**terminal** | **url** *url*} [**exportable**] *passphrase* **Example:** `Router(config)# crypto key import rsa mycs2 pem url nvram: PASSWORD`	Imports the generated RSA key pair. **Note** If you do not want the key to be exportable from your CA, import it back to the CA after it has been exported as a nonexportable key pair. Thus, the key cannot be taken off again.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 5	**show crypto key mypubkey rsa** **Example:** `Router# show crypto key mypubkey rsa`	(Optional) Displays the RSA public keys of your router.

Encrypting and Locking Private Keys on a Router

Digital signatures are used to authenticate one device to another device. To use digital signatures, private information (the private key) must be stored on the device that is providing the signature. The stored private information may aid an attacker who steals the hardware device that contains the private key; for example, a thief might be able to use the stolen router to initiate a secure connection to another site by using the RSA private keys stored in the router.

Note RSA keys are lost during password recovery operations. If you lose your password, the RSA keys will be deleted when you perform the password recovery operation. (This function prevents an attacker from performing password recovery and then using the keys.)

To protect the private RSA key from an attacker, a user can encrypt the private key that is stored in NVRAM via a passphrase. Users can also “lock” the private key, which blocks new connection attempts from a running router and protects the key in the router if the router is stolen by an attempted attacker.

Perform this task to encrypt and lock the private key that is saved to NVRAM.

Prerequisites

Before encrypting or locking a private key, you should perform the following tasks:

- Generate an RSA key pair as shown in the task "Generating an RSA Key Pair."
- Optionally, you can authenticate and enroll each router with the CA server.

Note The RSA keys must be unlocked while enrolling the CA. The keys can be locked while authenticating the router with the CA because the private key of the router is not used during authentication.

Restrictions for Encrypting and Locking Private Keys

Backward Compatibility Restriction

Any image prior to Cisco IOS Release 12.3(7)T does not support encrypted keys. To prevent your router from losing all encrypted keys, ensure that only unencrypted keys are written to NVRAM before booting an image prior to Cisco IOS Release 12.3(7)T.

If you must download an image prior to Cisco IOS Release 12.3(7)T, decrypt the key and immediately save the configuration so the downloaded image does not overwrite the configuration.

Interaction with Applications

An encrypted key is not effective after the router boots up until you manually unlock the key (via the **crypto key unlock rsa** command). Depending on which key pairs are encrypted, this functionality may adversely affect applications such as IP Security (IPSec), SSH, and SSL; that is, management of the router over a secure channel may not be possible until the necessary key pair is unlocked.

SUMMARY STEPS

1. **crypto key encrypt** [**write**] **rsa** [**name** *key-name*] **passphrase** *passphrase*
2. **exit**
3. **show crypto key mypubkey rsa**
4. **crypto key lock rsa** [**name** *key-name*] **passphrase** *passphrase*
5. **show crypto key mypubkey rsa**
6. **crypto key unlock rsa** [**name** *key-name*] **passphrase** *passphrase*
7. **configure terminal**
8. **crypto key decrypt** [**write**] **rsa** [**name** *key-name*] **passphrase** *passphrase*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto key encrypt** [**write**] **rsa** [**name** *key-name*] **passphrase** *passphrase* **Example:** `Router(config)# crypto key encrypt write rsa name pki.cisco.com passphrase cisco1234`	Encrypts the RSA keys. After this command is issued, the router can continue to use the key; the key remains unlocked. **Note** If the **write** keyword is not issued, the configuration must be manually written to NVRAM; otherwise, the encrypted key will be lost next time the router is reloaded.
Step 2	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 3	**show crypto key mypubkey rsa** **Example:** `Router# show crypto key mypubkey rsa`	(Optional) Shows that the private key is encrypted (protected) and unlocked. **Note** You can also use this command to verify that applications such as IKE and SSH are properly working after the key has been encrypted.
Step 4	**crypto key lock rsa** [**name** *key-name*] **passphrase** *passphrase* **Example:** `Router# crypto key lock rsa name pki.cisco.com passphrase cisco1234`	(Optional) Locks the encrypted private key on a running router. **Note** After the key is locked, it cannot be used to authenticate the router to a peer device. This behavior disables any IPSec or SSL connections that use the locked key. Any existing IPSec tunnels created on the basis of the locked key will be closed. If all RSA keys are locked, SSH will automatically be disabled.
Step 5	**show crypto key mypubkey rsa** **Example:** `Router# show crypto key mypubkey rsa`	(Optional) Shows that the private key is protected and locked. The output will also show failed connection attempts via applications such as IKE, SSH, and SSL.
Step 6	**crypto key unlock rsa** [**name** *key-name*] **passphrase** *passphrase* **Example:** `Router# crypto key unlock rsa name pki.cisco.com passphrase cisco1234`	(Optional) Unlocks the private key. **Note** After this command is issued, you can continue to establish IKE tunnels.

	Command or Action	Purpose
Step 7	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 8	**crypto key decrypt** [**write**] **rsa** [**name** *key-name*] **passphrase** *passphrase* **Example:** `Router(config)# crypto key decrypt write rsa name pki.cisco.com passphrase cisco1234`	(Optional) Deletes the encrypted key and leaves only the unencrypted key. **Note** The **write** keyword immediately saves the unencrypted key to NVRAM. If the **write** keyword is not issued, the configuration must be manually written to NVRAM; otherwise, the key will remain encrypted the next time the router is reloaded.

Removing RSA Key Pair Settings

You might want to remove an RSA key pair for one of the following reasons:

- During manual PKI operations and maintenance, old RSA keys can be removed and replaced with new keys.
- An existing CA is replaced and the new CA requires newly generated keys; for example, the required key size might have changed in an organization so you would have to delete the old 1024-bit keys and generate new 2048-bit keys.

Perform this task to remove all RSA keys or the specified RSA key pair that has been generated by your router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto key zeroize rsa** [*key-pair-label*]
4. **exit**
5. **show crypto key mypubkey rsa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto key zeroize rsa** [*key-pair-label*] **Example:** `Router(config)# crypto key zeroize rsa yancey-keys`	Deletes RSA key pairs from your router. • If the *key-pair-label* argument is not specified, all RSA keys that have been generated by your router will be deleted.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 5	**show crypto key mypubkey rsa** **Example:** `Router# show crypto key mypubkey rsa`	(Optional) Displays the RSA public keys of your router. This step allows you to verify that the RSA key pair has been successfully generated.

Configuration Examples for RSA Key Pair Deployment

This section contains the following configuration examples:

- Generating and Specifying RSA Keys: Example, page 1423
- Exporting and Importing RSA Keys: Examples, page 1423
- Encrypting and Locking Private Keys on a Router: Examples, page 1427

Generating and Specifying RSA Keys: Example

The following example is a sample trustpoint configuration that shows how to generate and specify the RSA key pair “exampleCAkeys”:

```
crypto key generate rsa general-purpose exampleCAkeys
crypto ca trustpoint exampleCAkeys
 enroll url http://exampleCAkeys/certsrv/mscep/mscep.dll
 rsakeypair exampleCAkeys 1024 1024
```

Exporting and Importing RSA Keys: Examples

This section contains the following configuration examples:

- Exporting and Importing RSA Keys in PKCS12 Files: Example, page 1424
- Generating, Exporting, Importing, and Verifying RSA Keys in PEM Files: Example, page 1424
- Exporting Router RSA Key Pairs and Certificates from PEM Files: Example, page 1425
- Importing Router RSA Key Pairs and Certificate from PEM Files: Example, page 1427

Exporting and Importing RSA Keys in PKCS12 Files: Example

In the following example, an RSA key pair "mynewkp" is generated on Router A, and a trustpoint name "mynewtp" is created and associated with the RSA key pair. The trustpoint is exported to a TFTP server, so that it can be imported on Router B. By importing the trustpoint "mynewtp" to Router B, the user has imported the RSA key pair "mynewkp" to Router B.

Router A

```
crypto key generate rsa general label mykeys exportable
! The name for the keys will be:mynewkp
Choose the size of the key modulus in the range of 360 to 2048 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take
a few minutes.
How many bits in the modulus [512]:
% Generating 512 bit RSA keys ...[OK]
!
crypto pki trustpoint mynewtp
 rsakeypair mykeys
 exit

crypto pki export mytp pkcs12 flash:myexport companyname
Destination filename [myexport]?
Writing pkcs12 file to tftp:/mytftpserver/myexport
CRYPTO_PKI:Exported PKCS12 file successfully.
Verifying checksum... OK (0x3307)
!
Feb 18 17:30:09 GMT:%CRYPTO-6-PKCS12EXPORT_SUCCESS:PKCS #12 Successfully Exported.
```

Router B

```
crypto pki import mynewtp pkcs12 flash:myexport companyname
Source filename [myexport]?
CRYPTO_PKI:Imported PKCS12 file successfully.

!
Feb 18 18:07:50 GMT:%CRYPTO-6-PKCS12IMPORT_SUCCESS:PKCS #12 Successfully Imported.
```

Generating, Exporting, Importing, and Verifying RSA Keys in PEM Files: Example

The following example shows how to generate, export, bring the key back (import), and verify the status of the RSA key pair "mycs":

```
! Generate the key pair
!
Router(config)# crypto key generate rsa general-purpose label mycs exportable
The name for the keys will be: mycs

Choose the size of the key modulus in the range of 360 to 2048 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take a few minutes.

How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys ...[OK]
!
! Archive the key pair to a remote location, and use a good password.
!
Router(config)# crypto key export rsa mycs pem url nvram:3des PASSWORD

% Key name:mycs
Usage:General Purpose Key
Exporting public key...
Destination filename [mycs.pub]?
```

```
Writing file to nvram:mycs.pub
Exporting private key...
Destination filename [mycs.prv]?
Writing file to nvram:mycs.prv
!
! Import the key as a different name.
!
Router(config)# crypto key import rsa mycs2 pem url nvram:mycs PASSWORD

% Importing public key or certificate PEM file...
Source filename [mycs.pub]?
Reading file from nvram:mycs.pub
% Importing private key PEM file...
Source filename [mycs.prv]?
Reading file from nvram:mycs.prv% Key pair import succeeded.
!
! After the key has been imported, it is no longer exportable.
!
! Verify the status of the key.
!
Router# show crypto key mypubkey rsa

% Key pair was generated at:18:04:56 GMT Jun 6 2003
Key name:mycs
Usage:General Purpose Key
Key is exportable.
Key Data:
30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00E65253
9C30C12E 295AB73F B1DF9FAD 86F88192 7D4FA4D2 8BA7FB49 9045BAB9 373A31CB
A6B1B8F4 329F2E7E 8A50997E AADBCFAA 23C29E19 C45F4F05 DBB2FA51 4B7E9F79
A1095115 759D6BC3 5DFB5D7F BCF655BF 6317DB12 A8287795 7D8DC6A3 D31B2486
C9C96D2C 2F70B50D 3B4CDDAE F661041A 445AE11D 002EEF08 F2A627A0 5B020301 0001
% Key pair was generated at:18:17:25 GMT Jun 6 2003
Key name:mycs2
Usage:General Purpose Key
Key is not exportable.
Key Data:
30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00E65253
9C30C12E 295AB73F B1DF9FAD 86F88192 7D4FA4D2 8BA7FB49 9045BAB9 373A31CB
A6B1B8F4 329F2E7E 8A50997E AADBCFAA 23C29E19 C45F4F05 DBB2FA51 4B7E9F79
A1095115 759D6BC3 5DFB5D7F BCF655BF 6317DB12 A8287795 7D8DC6A3 D31B2486
C9C96D2C 2F70B50D 3B4CDDAE F661041A 445AE11D 002EEF08 F2A627A0 5B020301 0001
```

Exporting Router RSA Key Pairs and Certificates from PEM Files: Example

The following example shows how to generate and export the RSA key pair "aaa" and certificates of the router in PEM files that are associated with the trustpoint "mycs." This example also shows PEM-formatted files, which include PEM boundaries before and after the base64-encoded data, that are used by other SSL and SSH applications.

```
Router(config)# crypto key generate rsa general-keys label aaa exportable

The name for the keys will be:aaa
Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose
Keys. Choosing a key modulus greater than 512 may take a few minutes.
!
How many bits in the modulus [512]:
% Generating 512 bit RSA keys ...[OK]
!
Router(config)# crypto pki trustpoint mycs
Router(ca-trustpoint)# enrollment url http://mycs
Router(ca-trustpoint)# rsakeypair aaa
```

```
Router(ca-trustpoint)# exit
Router(config)# crypto pki authenticate mycs
Certificate has the following attributes:
Fingerprint:C21514AC 12815946 09F635ED FBB6CF31
% Do you accept this certificate? [yes/no]: y
Trustpoint CA certificate accepted.
!
Router(config)# crypto pki enroll mycs
%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this password to the CA
Administrator in order to revoke your certificate.
For security reasons your password will not be saved in the configuration.
Please make a note of it.

Password:
Re-enter password:

% The fully-qualified domain name in the certificate will be: Router
% The subject name in the certificate will be:bizarro.cisco.com
% Include the router serial number in the subject name? [yes/no]: n
% Include an IP address in the subject name? [no]: n
Request certificate from CA? [yes/no]: y
% Certificate request sent to Certificate Authority
% The certificate request fingerprint will be displayed.
% The 'show crypto ca certificate' command will also show the fingerprint.

Router(config)# Fingerprint:8DA777BC 08477073 A5BE2403 812DD157

00:29:11:%CRYPTO-6-CERTRET:Certificate received from Certificate Authority

Router(config)# crypto ca export aaa pem terminal 3des cisco123
% CA certificate:
-----BEGIN CERTIFICATE-----
MIICAzCCAa2gAwIBAgIBATANBgkqhkiG9w0BAQUFADBOMQswCQYDVQQGEwJVUzES
<snip>
waDeNOSI3WlDa0AWq5DkVBkxwgn0TqIJXJOCttjHnWHK1LMcMVGn
-----END CERTIFICATE-----

% Key name:aaa
Usage:General Purpose Key
-----BEGIN RSA PRIVATE KEY-----
Proc-Type:4,ENCRYPTED
DEK-Info:DES-EDE3-CBC,ED6B210B626BC81A

Urguv0jnjwOgowWVUQ2XR5nbzzYHI2vGLunpH/IxIsJuNjRVjbAAUpGk7VnPCT87
<snip>
kLCOtxzEv7JHc72gMku9uUlrLSnFH5slzAtoC0czfU4=
-----END RSA PRIVATE KEY-----

% Certificate:
-----BEGIN CERTIFICATE-----
MIICTjCCAfigAwIBAgICIQUwDQYJKoZIhvcNAQEFBQAwTjELMAkGA1UEBhMCVVMx
<snip>
6xlBaIsuMxnHmr89KkKkYlU6
-----END CERTIFICATE-----
```

Importing Router RSA Key Pairs and Certificate from PEM Files: Example

The following example shows how to import the RSA key pairs and certificate to the trustpoint "ggg" from PEM files via TFTP:

```
Router(config)# crypto pki import ggg pem url tftp://10.1.1.2/johndoe/msca cisco1234
% Importing CA certificate...
Address or name of remote host [10.1.1.2]?
Destination filename [johndoe/msca.ca]?
Reading file from tftp://10.1.1.2/johndoe/msca.ca
Loading johndoe/msca.ca from 10.1.1.2 (via Ethernet0):!
[OK - 1082 bytes]

% Importing private key PEM file...
Address or name of remote host [10.1.1.2]?
Destination filename [johndoe/msca.prv]?
Reading file from tftp://10.1.1.2/johndoe/msca.prv
Loading johndoe/msca.prv from 10.1.1.2 (via Ethernet0):!
[OK - 573 bytes]

% Importing certificate PEM file...
Address or name of remote host [10.1.1.2]?
Destination filename [johndoe/msca.crt]?
Reading file from tftp://10.1.1.2/johndoe/msca.crt
Loading johndoe/msca.crt from 10.1.1.2 (via Ethernet0):!
[OK - 1289 bytes]
% PEM files import succeeded.
Router(config)#
```

Encrypting and Locking Private Keys on a Router: Examples

This section contains the following configuration examples:

- Configuring and Verifying an Encrypted Key: Example, page 1427
- Configuring and Verifying a Locked Key: Example, page 1428

Configuring and Verifying an Encrypted Key: Example

The following example shows how to encrypt the RSA key "pki1-72a.cisco.com." Thereafter, the **show crypto key mypubkey rsa** command is issued to verify that the RSA key is encrypted (protected) and unlocked.

```
Router(config)# crypto key encrypt rsa name pki1-72a.cisco.com passphrase cisco1234
Router(config)# exit
Router# show crypto key mypubkey rsa

% Key pair was generated at:00:15:32 GMT Jun 25 2003
Key name:pki1-72a.cisco.com
Usage:General Purpose Key
*** The key is protected and UNLOCKED. ***
Key is not exportable.
Key Data:
305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00E0CC9A 1D23B52C
CD00910C ABD392AE BA6D0E3F FC47A0EF 8AFEE340 0EC1E62B D40E7DCC
23C4D09E
03018B98 E0C07B42 3CFD1A32 2A3A13C0 1FF919C5 8DE9565F 1F020301 0001
% Key pair was generated at:00:15:33 GMT Jun 25 2003
Key name:pki1-72a.cisco.com.server
Usage:Encryption Key
```

```
Key is exportable.
Key Data:
307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00D3491E 2A21D383
854D7DA8 58AFBDAC 4E11A7DD E6C40AC6 66473A9F 0C845120 7C0C6EC8 1FFF5757
3A41CE04 FDCB40A4 B9C68B4F BC7D624B 470339A3 DE739D3E F7DDB549 91CD4DA4
DF190D26 7033958C 8A61787B D40D28B8 29BCD0ED 4E6275C0 6D020301 0001
Router#
```

Configuring and Verifying a Locked Key: Example

The following example shows how to lock the key "pki1-72a.cisco.com." Thereafter, the **show crypto key mypubkey rsa** command is issued to verify that the key is protected (encrypted) and locked.

```
Router# crypto key lock rsa name pki1-72a.cisco.com passphrase cisco1234
!
Router# show crypto key mypubkey rsa

% Key pair was generated at:20:29:41 GMT Jun 20 2003
Key name:pki1-72a.cisco.com
Usage:General Purpose Key
*** The key is protected and LOCKED. ***
Key is exportable.
Key Data:
305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00D7808D C5FF14AC
0D2B55AC 5D199F2F 7CB4B355 C555E07B 6D0DECBE 4519B1F0 75B12D6F 902D6E9F
B6FDAD8D 654EF851 5701D5D7 EDA047ED 9A2A619D 5639DF18 EB020301 0001
```

Where to Go Next

After you have generated an RSA key pair, you should set up the trustpoint. If you have already set up the trustpoint, you should authenticate and enroll the routers in a PKI. For information on enrollment, see the module "Configuring Certificate Enrollment for a PKI."

Additional References

The following sections provide references related to configuring RSA keys for a PKI.

Related Documents

Related Topic	Document Title
Overview of PKI, including RSA keys, certificate enrollment, and CAs	"Cisco IOS PKI Overview: Understanding and Planning a PKI" module
PKI commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*, Release 12.4

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Feature Information for RSA Keys Within a PKI

Table 57 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for a specific command was introduced, see the command reference documentation.

For information on a feature in this technology that is not documented here, see the Implementing and Managing PKI Features Roadmap.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Table 57 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 57 Feature Information for RSA Keys Within a PKI

Feature Name	Software Releases	Feature Configuration Information
Exporting and Importing RSA Keys	12.2(15)T	This feature allows you to transfer security credentials between devices by exporting and importing RSA keys. The key pair that is shared between two devices will allow one device to immediately and transparently take over the functionality of the other router. The following sections provide information about this feature: • Benefits of Exportable RSA Keys • Exporting and Importing RSA Keys in PKCS12 Files The following commands were introduced or modified by this feature: **crypto ca export pkcs12**, **crypto ca import pkcs12**, **crypto key generate rsa (IKE)**
Import of RSA Key Pair and Certificates in PEM Format	12.3(4)T	This feature allows customers to use PEM-formatted files to import or export RSA key pairs. PEM-formatted files allow customers to directly use existing RSA key pairs on their Cisco IOS routers instead of generating new keys. The following sections provide information about this feature: • Benefits of Exportable RSA Keys • Exporting and Importing RSA Keys in PEM-Formatted Files The following commands were introduced by this feature: **crypto ca export pem**, **crypto ca import pem**, **crypto key export pem**, **crypto key import pem**

Table 57 Feature Information for RSA Keys Within a PKI (continued)

Feature Name	Software Releases	Feature Configuration Information
Multiple RSA Key Pair Support	12.2(8)T	This feature allows a user to configure a router to have multiple RSA key pairs. Thus, the Cisco IOS software can maintain a different key pair for each identity certificate. The following sections provide information about this feature: • Reasons to Store Multiple RSA Keys on a Router • Generating and Storing Multiple RSA Key Pairs The following commands were introduced or modified by this feature: **crypto key generate rsa**, **crypto key zeroize rsa**, **rsakeypair**
Protected Private Key Storage	12.3(7)T	This feature allows a user to encrypt and lock the RSA private keys that are used on a Cisco IOS router, thereby, preventing unauthorized use of the private keys. The following section provides information about this feature: • Encrypting and Locking Private Keys on a Router The following commands were introduced or modified by this feature: **crypto key decrypt rsa**, **crypto key encrypt rsa**, **crypto key lock rsa**, **crypto key unlock rsa**, **show crypto key mypubkey rsa**

Configuring Authorization and Revocation of Certificates in a PKI

After a certificate is validated as a properly signed certificate, it is authorized—via methods such as, certificate maps, PKI-AAA, or a certificate-based access control list (ACL)—and the revocation status is checked by the issuing certification authority (CA) to ensure that the certificate has not been revoked.

This module describes how to configure authorization and revocation of certificates in a public key infrastructure (PKI).

Module History

This module was first published on May 2, 2005, and last updated on May 2, 2005.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all features. To find information about feature support and configuration, use the "Feature Information for Certificate Authorization and Revocation" section on page 1460.

Contents

- Prerequisites for Authorization and Revocation of Certificates, page 1433
- Information About Authorization and Revocation of Certificates, page 1434
- How to Configure Authorization and Revocation of Certificates for Your PKI, page 1440
- Configuration Examples for Setting Up Authorization and Revocation of Certificates, page 1450
- Additional References, page 1460
- Feature Information for Certificate Authorization and Revocation, page 1460

Prerequisites for Authorization and Revocation of Certificates

Plan Your PKI Strategy

Tip It is strongly recommended that you plan your entire PKI strategy before you begin to deploy actual certificates.

Authorization and revocation can occur only after you or a network administrator have completed the following tasks:

- Configured the CA
- Enrolled peer devices with the CA
- Identified and configured the protocol (such as IP Security [IPSec] or secure socket layer [SSL]) that is to be used for peer-to-peer communication.

You should decide which authorization and revocation strategy you are going to configure before enrolling peer devices because the peer device certificates might have to contain authorization and revocation-specific information.

"crypto ca" to "crypto pki" command-line interface (CLI) change

As of Cisco IOS Release 12.3(7)T, all commands that begin as "crypto ca" have been changed to begin as "crypto pki." Although the router will still accept crypto ca commands, all output will be read back as crypto pki.

Information About Authorization and Revocation of Certificates

Before configuring certificate authorization and revocation, you should understand the following concepts:

- PKI Authorization, page 1434
- PKI and AAA Server Integration for Certificate Status, page 1435
- CRLs or OCSP Server: Choosing a Certificate Revocation Mechanism, page 1436
- When to Use Certificate-Based ACLs for Authorization or Revocation, page 1438

PKI Authorization

PKI authentication does not provide authorization. Current solutions for authorization are specific to the router that is being configured, although a centrally managed solution is often required.

There is not a standard mechanism by which certificates are defined as authorized for some tasks and not for others. This authorization information can be captured in the certificate itself if the application is aware of the certificate-based authorization information. But this solution does not provide a simple mechanism for real-time updates to the authorization information and forces each application to be aware of the specific authorization information embedded in the certificate.

When the certificate-based ACL mechanism is configured as part of the trustpoint authentication, the application is no longer responsible for determining this authorization information, and it is no longer possible to specify for which application the certificate is authorized. In some cases, the certificate-based ACL on the router gets so large that it cannot be managed. Additionally, it is beneficial to retrieve certificate-based ACL indications from an external server. (For more information on using certificate-based ACLs for authentication, see the section "When to Use Certificate-Based ACLs for Authorization or Revocation.")

Current solutions to the real-time authorization problem involve specifying a new protocol and building a new server (with associated tasks, such as management and data distribution).

PKI and AAA Server Integration for Certificate Status

Integrating your PKI with a AAA server provides an alternative online certificate status solution that leverages the existing AAA infrastructure. Certificates can be listed in the AAA database with appropriate levels of authorization. For components that do not explicitly support PKI-AAA, a default label of "all" from the AAA server provides authorization. Likewise, a label of "none" from the AAA database indicates that the specified certificate is not valid. (The absence of any application label is equivalent, but "none" is included for completeness and clarity). If the application component does support PKI-AAA, the component may be specified directly; for example, the application component could be "ipsec," "ssl," or "osp." (ipsec=IP Security, ssl=Secure Sockets Layer, and osp=Open Settlement Protocol.)

Note

- Currently, no application component supports specification of the application label.
- There may be a time delay when accessing the AAA server. If the AAA server is not available, the authorization fails.

RADIUS or TACACS+: Choosing a AAA Server Protocol

The AAA server can be configured to work with either the RADIUS or TACACS+ protocol. When you are configuring the AAA server for the PKI integration, you must set the RADIUS or TACACS attributes that are required for authorization.

If the RADIUS protocol is used, the password that is configured for the username in the AAA server should be set to "cisco," which is acceptable because the certificate validation provides authentication and the AAA database is only being used for authorization. When the TACACS protocol is used, the password that is configured for the username in the AAA server is irrelevant because TACACS supports authorization without requiring authentication (the password is used for authentication).

In addition, if you are using TACACS, you must add a PKI service to the AAA server. The custom attribute "cert-application=all" is added under the PKI service for the particular user or usergroup to authorize the specific username.

Attribute-Value Pairs for PKI and AAA Server Integration

Table 58 lists the attribute-value (AV) pairs that are to be used when setting up PKI integration with a AAA server. (Note the values shown in the table are possible values.) The AV pairs must match the client configuration. If they do not match, the peer certificate is not authorized.

Note Users can sometimes have AV pairs that are different from those of every other user. As a result, a unique username is required for each user. The **all** parameter (within the **authorization username** command) specifies that the entire subject name of the certificate will be used as the authorization username.

Table 58 AV Pairs That Must Match

AV Pair	Value
cisco-avpair=pki:cert-application=all	Valid values are “all” and “none.”
cisco-avpair=pki:cert-trustpoint=msca	The value is a Cisco IOS command-line interface (CLI) configuration trustpoint label. **Note** The cert-trustpoint AV pair is normally optional. If it is specified, the Cisco IOS router query must be coming from a certificate trustpoint that has a matching label, and the certificate that is authenticated must have the specified certificate serial number.
cisco-avpair=pki:cert-serial=16318DB700 0100001671	The value is a certificate serial number. **Note** The cert-serial AV pair is normally optional. If it is specified, the Cisco IOS router query must be coming from a certificate trustpoint that has a matching label, and the certificate that is authenticated must have the specified certificate serial number.
cisco-avpair=pki:cert-lifetime-end=1:00 jan 1, 2003	The cert-lifetime-end AV pair is available to artificially extend a certificate lifetime beyond the time period that is indicated in the certificate itself. If the cert-lifetime-end AV pair is used, the cert-trustpoint and cert-serial AV pairs must also be specified. The value must match the following form: hours:minutes month day, year. **Note** Only the first three characters of a month are used: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec. If more than three characters are entered for the month, the remaining characters are ignored (for example Janxxxx).

CRLs or OCSP Server: Choosing a Certificate Revocation Mechanism

After a certificate is validated as a properly signed certificate, a certificate revocation method is performed to ensure that the certificate has not been revoked by the issuing CA. Cisco IOS software supports two revocation mechanisms—certificate revocation lists (CRLs) and Online Certificate Status Protocol (OCSP). (Cisco IOS software also supports AAA integration for certificate checking; however, additional authorization functionality is included. For more information on PKI and AAA certificate authorization and status check, see the section “PKI and AAA Server Integration for Certificate Status.”) The following sections explain how each revocation mechanism works:

- What Is a CRL?, page 1437
- What Is OCSP?, page 1438

What Is a CRL?

A CRL contains a list of revoked certificates. The CRL is created and digitally signed by the CA that originally issued the certificates. The CRL contains dates for when each certificate was issued and when it expires.

CAs publish new CRLs periodically or when a certificate they are responsible for has been revoked. A new CRL is issued when the current CRL expires. When the CRL expires, the router deletes it from its cache. A new CRL is downloaded when a certificate is presented for verification; however, if a newer version of the CRL that lists the certificate under examination is on the server but the router is still using the CRL in its cache, the router will not know that the certificate has been revoked. The certificate will pass the revocation check even though it should have been denied.

When a CA issues a certificate, the CA can include in the certificate the CRL distribution point (CDP) for that certificate. Cisco IOS client devices use CDPs to locate and load the correct CRL. The Cisco IOS client supports multiple CDPs, but the Cisco IOS CA currently supports only one CDP; however, third-party vendor CAs may support multiple CDPs or different CDPs per certificate. If a CDP is not specified in the certificate, the client device will use the default Simple Certificate Enrollment Protocol (SCEP) method to retrieve the CRL. (The CDP location can be specified via the **cdp-url** command.)

When implementing CRLs, you should consider the following design considerations:

- CRL lifetimes and the Security Association (SA) and Internet Key Exchange (IKE) lifetimes

  The CRL lifetime determines the length of time between CA-issued updates to the CRL. (The default CRL lifetime value, which is 168 hours [1 week], can be changed via the **lifetime crl** command.)

- The method and location of the CDP
  - The method determines how the CRL is retrieved; some possible choices include HTTP, lightweight directory access protocol (LDAP), SCEP, or TFTP.

    HTTP, TFTP, and LDAP are the most commonly used methods. Although Cisco IOS software defaults to SCEP, an HTTP CDP is recommended for large installations using CRLs because HTTP can be made highly scalable.
  - The location determines from where the CRL is retrieved; for example, you can specify the server and file path from which to retrieve the CRL.

Querying All CDPs During Revocation Check

When a CDP server does not respond to a request, the Cisco IOS software reports an error, which may result in the peer's certificate being rejected. To prevent a possible certificate rejection and if there are multiple CDPs in a certificate, the Cisco IOS software will attempt to use the CDPs in the order in which they appear in the certificate. The router will attempt to retrieve a CRL using each CDP URL or directory specification. If an error occurs using a CDP, an attempt will be made using the next CDP.

Note Prior to Cisco IOS Release 12.3(7)T, the Cisco IOS software makes only one attempt to retrieve the CRL, even when the certificate contains more than one CDP.

Tip Although the Cisco IOS software will make every attempt to obtain the CRL from one of the indicated CDPs, it is recommended that you use an HTTP CDP server with high-speed redundant HTTP servers to avoid application timeouts because of slow CDP responses.

What Is OCSP?

OCSP is an online mechanism that is used to determine certificate validity. OCSP can provide real-time certificate status checking, whereas CRLs provide only periodic certificate status.

A network administrator can configure a central OCSP server to collect and update CRLs from different CA servers; thus, the devices within the network can rely on the OCSP server to check the certificate status without retrieving and caching each CRL for every peer. When peers have to check the revocation status of a certificate, they send a query to the OCSP server that includes the serial number of the certificate in question. The OCSP server copies the CRL to determine if the CA has listed the certificate as being revoked; the server then responds to the peer. The dialog between the OCSP server and the peer consumes less bandwidth than most CRL downloads.

If the OCSP server is using a CRL, CRL time limitations will be applicable; that is, a CRL that is still valid might be used by the OCSP server although a new CRL has been issued by the CRL containing additional certificate revocation information. Because fewer devices are downloading the CRL information on a regular basis, you can decrease the CRL lifetime value or configure the OCSP server not to cache the CRL. For more information, check your OCSP server documentation.

When to Use an OCSP Server

OCSP may be more appropriate than CRLs if your PKI has any of the following characteristics:

- There are a large number of revoked certificates or multiple CRLs. CRLs in the cache can consume a large quantity of memory; an OCSP server offers real-time updates so it does not require a cache, and therefore does not consume as much memory as using CRLs.
- CRLs expire frequently, causing the CDP to handle a larger load of CRLs.
- Real-time certificate revocation status is necessary. Because CRLs are updated only periodically, a certificate can pass a revocation check even though it should have been revoked because the latest CRL may not yet have been cached by the client device.

When to Use Certificate-Based ACLs for Authorization or Revocation

Certificates contain several fields that are used to determine whether a device or user is authorized to perform a specified action.

Because certificate-based ACLs are configured on the device, they do not scale well for large numbers of ACLs; however, certificate-based ACLs do provide very granular control of specific device behavior. Certificate-based ACLs are also leveraged by additional features to help determine when PKI components such as revocation, authorization, or a trustpoint should be used. They provide a general mechanism allowing users to select a specific certificate or a group of certificates that are being validated for either authorization or additional processing.

Certificate-based ACLs specify one or more fields within the certificate and an acceptable value for each specified field. You can specify which fields within a certificate should be checked and which values those fields may or may not have.

There are six logical tests for comparing the field with the value—equal, not equal, contains, does not contain, less than, and greater than or equal. If more than one field is specified within a single certificate-based ACL, the tests of all of the fields within the ACL must succeed to match the ACL. The same field may be specified multiple times within the same ACL. More than one ACL may be specified, and ACL will be processed in turn until a match is found or all of the ACLs have been processed.

Ignore Revocation Checks Using a Certificate-Based ACL

Certificate-based ACLs can be configured to instruct your router to ignore the revocation check and expired certificates of a valid peer. Thus, a certificate that meets the specified criteria can be accepted regardless of the validity period of the certificate, or if the certificate meets the specified criteria, revocation checking does not have to be performed. You can also use a certificate-based ACL to ignore the revocation check when the communication with a AAA server is protected with a certificate.

Ignoring Revocation Lists

To allow a trustpoint to enforce CRLs except for specific certificates, enter the **match certificate** command with the **skip revocation-check** keyword. This type of enforcement is most useful in a hub-and-spoke configuration in which you also want to allow direct spoke-to-spoke connections. In pure hub-and-spoke configurations, all spokes connect only to the hub, so CRL checking is necessary only on the hub. For one spoke to communicate directly with another spoke, the **match certificate** command with the **skip revocation-check** keyword can be used for neighboring peer certificates instead of requiring a CRL on each spoke.

Ignoring Expired Certificates

To configure your router to ignore expired certificates, enter the **match certificate** command with the **allow expired-certificate** keyword. This command has the following purposes:

- If the certificate of a peer has expired, this command may be used to "allow" the expired certificate until the peer can obtain a new certificate.
- If your router clock has not yet been set to the correct time, the certificate of a peer will appear to be not yet valid until the clock is set. This command may be used to allow the certificate of the peer even though your router clock is not set.

Note

- If Network Time Protocol (NTP) is available only via the IPSec connection (usually via the hub in a hub-and-spoke configuration), the router clock can never be set. The tunnel to the hub cannot be "brought up" because the certificate of the hub is not yet valid.
- "Expired" is a generic term for a certificate that is expired or that is not yet valid. The certificate has a start and end time. An expired certificate, for purposes of the ACL, is one for which the current time of the router is outside the start and end times specified in the certificate.

Skipping the AAA Check of the Certificate

If the communication with an AAA server is protected with a certificate, and you want to skip the AAA check of the certificate, use the **match certificate** command with the **skip authorization-check** keyword. For example, if a virtual private network (VPN) tunnel is configured so that all AAA traffic goes over that tunnel, and the tunnel is protected with a certificate, you can use the **match certificate** command with the **skip authorization-check** keyword to skip the certificate check so that the tunnel can be established.

The **match certificate** command and the **skip authorization-check** keyword should be configured after PKI integration with an AAA server is configured.

Note If the AAA server is available only via an IPSec connection, the AAA server cannot be contacted until after the IPSec connection is established. The IPSec connection cannot be "brought up" because the certificate of the AAA server is not yet valid.

How to Configure Authorization and Revocation of Certificates for Your PKI

This section contains the following procedures:

- Configuring PKI Integration with a AAA Server, page 1440
- Configuring a Revocation Mechanism for Cisco IOS Certificate Status Checking, page 1444
- Overriding Certificate Revocation and Authorization Settings, page 1447

Configuring PKI Integration with a AAA Server

Perform this task to generate a AAA username from the certificate presented by the peer and specify which fields within a certificate should be used to build the AAA database username.

Restrictions When Using the Entire Subject Name for PKI Authorization

The following restrictions should be considered when using the **all** keyword as the subject name for the **authorization username** command:

- Some AAA servers limit the length of the username (for example, to 64 characters). As a result, the entire certificate subject name cannot be longer than the limitation of the server.
- Some AAA servers limit the available character set that may be used for the username (for example, a space [ ] and an equal sign [=] may not be acceptable). You cannot use the **all** keyword for a AAA server having such a character-set limitation.
- The **subject-name** command in the trustpoint configuration may not always be the final AAA subject name. If the fully qualified domain name (FQDN), serial number, or IP address of the router are included in a certificate request, the subject name field of the issued certificate will also have these components. To turn off the components, use the **fqdn**, **serial-number**, and **ip-address** commands with the **none** keyword.
- CA servers sometimes change the requested subject name field when they issue a certificate. For example, CA servers of some vendors switch the relative distinguished names (RDNs) in the requested subject names to the following order: CN, OU, O, L, ST, and C. However, another CA server might append the configured LDAP directory root (for example, O=cisco.com) to the end of the requested subject name.
- Depending on the tools you choose for displaying a certificate, the printed order of the RDNs in the subject name could be different. Cisco IOS software always displays the least significant RDN first, but other software, such as Open Source Secure Socket Layer (OpenSSL), does the opposite. Therefore, if you are configuring a AAA server with a full distinguished name (DN) (subject name) as the corresponding username, ensure that the Cisco IOS software style (that is, with the least-significant RDN first) is used.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authorization network** *listname* [*method*]

5. **crypto pki trustpoint** *name*
6. **enrollment url** *url*
7. **revocation-check** *method*
8. **exit**
9. **authorization username** {**subjectname** *subjectname*}
10. **authorization list** *listname*
11. **tacacs-server host** *hostname* [**key** *string*]

    or

    radius-server host *hostname* [**key** *string*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables the AAA access control model.
Step 4	**aaa authorization network** *listname* [*method*] **Example:** `Router (config)# aaa authorization network maxaaa group tacacs+`	Sets the parameters that restrict user access to a network. • *method*—Can be **group radius**, **group tacacs+**, or **group group-name**.
Step 5	**crypto pki trustpoint** *name* **Example:** `Route (config)# crypto pki trustpoint msca`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 6	**enrollment url** *url* **Example:** `Router (ca-trustpoint)# enrollment url http://caserver.mycompany.com`	Specifies the enrollment parameters of your CA. • The *url* argument is the URL of the CA to which your router should send certificate requests.
Step 7	**revocation-check** *method* **Example:** `Router (ca-trustpoint)# revocation-check crl`	(Optional) Checks the revocation status of a certificate.

	Command or Action	Purpose
Step 8	**exit** **Example:** `Router (ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 9	**authorization username** {**subjectname** *subjectname*} **Example:** `Router (config)# authorization username subjectname serialnumber`	Sets parameters for the different certificate fields that are used to build the AAA username. The *subjectname* argument can be any of the following: • **all**—Entire distinguished name (subject name) of the certificate. • **commonname**—Certification common name. • **country**—Certificate country. • **email**—Certificate e-mail. • **ipaddress**—Certificate IP address. • **locality**—Certificate locality. • **organization**—Certificate organization. • **organizationalunit**—Certificate organizational unit. • **postalcode**—Certificate postal code. • **serialnumber**—Certificate serial number. • **state**—Certificate state field. • **streetaddress**—Certificate street address. • **title**—Certificate title. • **unstructuredname**—Certificate unstructured name.
Step 10	**authorization list** *listname* **Example:** `Route (config)# authorization list maxaaa`	Specifies the AAA authorization list.
Step 11	**tacacs-server host** *hostname* [**key** *string*] **Example:** `Router(config)# tacacs-server host 10.2.2.2 key a_secret_key` or **radius-server host** *hostname* [**key** *string*] **Example:** `Router(config)# radius-server host 10.1.1.1 key another_secret_key`	Specifies a TACACS+ host. or Specifies a RADIUS host.

Troubleshooting Tips

To display debug messages for the trace of interaction (message type) between the CA and the router, use the **debug crypto pki transactions** command. (See sample output, which shows a successful PKI integration with AAA server exchange and a failed PKI integration with AAA server exchange.)

Successful Exchange

```
Router# debug crypto pki transactions

Apr 22 23:15:03.695: CRYPTO_PKI: Found a issuer match
Apr 22 23:15:03.955: CRYPTO_PKI: cert revocation status unknown.
Apr 22 23:15:03.955: CRYPTO_PKI: Certificate validated without
revocation check
```

Each line that shows “CRYPTO_PKI_AAA” indicates the state of the AAA authorization checks. Each of the AAA AV pairs is indicated, and then the results of the authorization check are shown.

```
Apr 22 23:15:04.019: CRYPTO_PKI_AAA: checking AAA authorization
(ipsecca_script_aaalist, PKIAAA-L, <all>)
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: reply attribute ("cert-application"
= "all")
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: reply attribute ("cert-trustpoint"
= "yni-u10")
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: reply attribute ("cert-serial" =
"15DE")
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: authorization passed
Apr 22 23:12:30.327: CRYPTO_PKI: Found a issuer match
```

Failed Exchange

```
Router# debug crypto pki transactions

Apr 22 23:11:13.703: CRYPTO_PKI_AAA: checking AAA authorization =
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: reply attribute ("cert-application"
=
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: reply attribute ("cert-trustpoint"
=
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: reply attribute ("cert-serial" =3D
=
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: reply attribute =
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: parsed cert-lifetime-end as: 21:30
=
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: timezone specific extended =
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: cert-lifetime-end is expired
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: cert-lifetime-end check failed.
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: authorization failed
```

In the above failed exchange, the certificate has expired.

Configuring a Revocation Mechanism for Cisco IOS Certificate Status Checking

Perform this task to set up a certificate revocation mechanism—CRLs or OCSP— that is to be used to check the status of certificates in a PKI.

The revocation-check Command

Use the **revocation-check** command to specify at least one method (OCSP, CRL, or skip the revocation check) that is to be used to ensure that the certificate of a peer has not been revoked. For multiple methods, the order in which the methods are applied is determined by the order specified via this command.

If your router does not have the applicable CRL and is unable to obtain one or if the OCSP server returns an error, your router will reject the peer's certificate—unless you include the **none** keyword in your configuration. If the **none** keyword is configured, a revocation check will not be performed and the certificate will always be accepted.

Prerequisites

- Before issuing any client certificates, the appropriate settings on the server (such as setting the CDP) should be configured.
- When configuring an OCSP server to return the revocation status for a CA server, the OCSP server must be configured with an OCSP response signing certificate that is issued by that CA server. Ensure that the signing certificate is in the correct format, or the router will not accept the OCSP response. See your OCSP manual for additional information.

Restrictions

- OCSP transports messages over HTTP, so there may be a time delay when you access the OCSP server.
- If the OCSP server depends on normal CRL processing to check revocation status, the same time delay that affects CRLs will also apply to OCSP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **ocsp url** *url*
5. **revocation-check** *method1* [*method2* [*method3*]]
6. **exit**
7. **exit**
8. **show crypto pki certificates**
9. **show crypto pki trustpoints** [**status** | *label* [**status**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint hazel`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	**ocsp url** *url* **Example:** `Router(ca-trustpoint)# ocsp url http://ocsp-server`	(Optional) Specifies the URL of an OCSP server so that the trustpoint can check the certificate status. This URL will override the URL of the OCSP server (if one exists) in Authority Info Access (AIA) extension of the certificate.
Step 5	**revocation-check** *method1* [*method2* [*method3*]] **Example:** `Router(ca-trustpoint)# revocation-check ocsp none`	Checks the revocation status of a certificate. • **crl**—Certificate checking is performed by a CRL. This is the default option. • **none**—Certificate checking is ignored. • **ocsp**—Certificate checking is performed by an OCSP server. If a second and third method are specified, each method will be used only if the previous method returns an error, such as a server being down.
Step 6	**exit** **Example:** `Router(ca-trustpoint)# exit`	Returns to global configuration mode.
Step 7	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.
Step 8	**show crypto pki certificates** **Example:** `Router# show crypto pki certificates`	(Optional) Displays information about your certificates.
Step 9	**show crypto pki trustpoints** [**status** | *label* [**status**]] **Example:** `Router# show crypto pki trustpoints`	Displays information about the trustpoint configured in router.

Examples

The following is a sample OCSP response signing certificate. Note that only the OCSP-related extensions are in bold.

```
Certificate:
        Data:
            Version: v3
            Serial Number:0x14
            Signature Algorithm:MD5withRSA - 1.2.840.113549.1.1.4
            Issuer:CN=CA server,OU=PKI,O=Cisco Systems
            Validity:
                Not Before:Thursday, August 8, 2002 4:38:05 PM PST
                Not  After:Tuesday, August 7, 2003 4:38:05 PM PST
            Subject:CN=OCSP server,OU=PKI,O=Cisco Systems
            Subject Public Key Info:
                Algorithm:RSA - 1.2.840.113549.1.1.1
                Public Key:
                    Exponent:65537
                    Public Key Modulus:(1024 bits) :
                        <snip>

            Extensions:
                Identifier:Subject Key Identifier - 2.5.29.14
                    Critical:no
                    Key Identifier:
                        <snip>

                Identifier:Authority Key Identifier - 2.5.29.35
                    Critical:no
                    Key Identifier:
                        <snip>

                Identifier:OCSP NoCheck:- 1.3.6.1.5.5.7.48.1.5
                     Critical:no
               Identifier:Extended Key Usage:- 2.5.29.37
                     Critical:no
                     Extended Key Usage:
                     OCSPSigning
                Identifier:CRL Distribution Points - 2.5.29.31
                    Critical:no
                    Number of Points:1
                    Point 0
                        Distribution Point:
[URIName:ldap://CA-server/CN=CA server,OU=PKI,O=Cisco Systems]
        Signature:
            Algorithm:MD5withRSA - 1.2.840.113549.1.1.4
            Signature:
            <snip>
```

Overriding Certificate Revocation and Authorization Settings

Perform this task to specify a certificate-based ACL, to ignore revocation checks or expired certificates, and manually override the default CDP location, as appropriate.

Overview: Configuring Certificate-Based ACLs to Ignore Revocation Checks

To configure your router to use certificate-based ACLs to ignore revocation checks and expired certificates, perform the following steps:

- Identify an existing trustpoint or create a new trustpoint to be used when verifying the certificate of the peer. Authenticate the trustpoint if it has not already been authenticated. The router may enroll with this trustpoint if you want. Do not set optional CRLs for the trustpoint if you plan to use the **match certificate** command and **skip revocation-check** keyword.
- Determine the unique characteristics of the certificates that should not have their CRL checked and of the expired certificates that should be allowed.
- Define a certificate map to match the characteristics identified in the prior step.
- You can add the **match certificate** command and **skip revocation-check** keyword and the **match certificate command** and **allow expired-certificate** keyword to the trustpoint that was created or identified in the first step.

Manually Overriding CDPs in a Certificate

Users can override the CDPs in a certificate with a manually configured CDP. Manually overriding the CDPs in a certificate can be advantageous when a particular server is unavailable for an extended period of time. The certificate's CDPs can be replaced with a URL or directory specification without reissuing all of the certificates that contain the original CDP.

Prerequisites

- The trustpoint should be defined and authenticated before attaching certificate maps to the trustpoint.
- The certificate map must be configured before the CDP override feature can be enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki certificate map** *label sequence-number*
4. *field-name match criteria match-value*
5. **exit**
6. **crypto pki trustpoint** *name*
7. **match certificate** *certificate-map-label* [**allow expired-certificate | skip revocation-check | skip authorization-check**]
8. **match certificate** *certificate-map-label* **override cdp** {**url | directory**} *string*
9. **exit**
10. **show crypto pki certificates**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki certificate map** *label sequence-number* **Example:** `Router(config)# crypto pki certificate map Group 10`	Defines values in a certificate that should be matched or not matched and enters ca-certificate-map configuration mode.
Step 4	*field-name match criteria match-value* **Example:** `Router(ca-certificate-map)# subject-name co Cisco`	Specifies one or more certificate fields together with their matching criteria and the value to match. The *field-name* is one of the following case-insensitive name strings or a date: • **subject-name** • **issuer-name** • **unstructured-subject-name** • **alt-subject-name** • **name** • **valid-start** • **expires-on** **Note** Date field format is dd mm yyyy hh:mm:ss or mmm dd yyyy hh:mm:ss. The *match-criteria* is one of the following logical operators: • **eq** —equal (valid for name and date fields) • **ne** —not equal (valid for name and date fields) • **co** —contains (valid only for name fields) • **nc** —does not contain (valid only for name fields) • **lt** —less than (valid only for date fields) • **ge** —greater than or equal (valid only for date fields) The *match-value* is the name or date to test with the logical operator assigned by match-criteria. **Note** Use this command only when setting up a certificate-based ACL—not when setting up a certificate-based ACL to ignore revocation checks or expired certificates.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(ca-certificate-map)# exit`	Returns to global configuration mode.
Step 6	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint Access2`	Declares the trustpoint and a given name.
Step 7	**match certificate** *certificate-map-label* [**allow expired-certificate** | **skip revocation-check** | **skip authorization-check**] **Example:** `Router(ca-trustpoint)# match certificate Group skip revocation-check`	(Optional) Associates the certificate-based ACL (that was defined via the **crypto pki certificate map** command) to a trustpoint. • *certificate-map-label*—Must match the *label* argument specified via the **crypto pki certificate map** command. • **allow expired-certificate**—Ignores expired certificates. • **skip revocation-check**—Allows a trustpoint to enforce CRLs except for specific certificates. • **skip authorization-check**—Skips the AAA check of a certificate when PKI integration with an AAA server is configured
Step 8	**match certificate** *certificate-map-label* **override cdp** {**url** | **directory**} *string* **Example:** `Router(ca-trustpoint)# match certificate Group1 override cdp url http://server.cisco.com`	(Optional) Manually overrides the existing CDP entries for a certificate with a URL or directory specification. • *certificate-map-label*—A user-specified label that must match the *label* argument specified in a previously defined **crypto pki certificate map** command. • **url**—Specifies that the certificate's CDPs will be overridden with an HTTP or LDAP URL. • **directory**—Specifies that the certificate's CDPs will be overridden with an LDAP directory specification. • *string*—The URL or directory specification. **Note** Some applications may time out before all CDPs have been tried and will report an error message. The error message will not affect the router, and the Cisco IOS software will continue attempting to retrieve a CRL until all CDPs have been tried.

	Command or Action	Purpose
Step 9	**exit** **Example:** `Router(ca-trustpoint)# exit` **Example:** `Router(config)# exit`	Returns to global configuration mode and privileged EXEC configuration mode.
Step 10	**show crypto pki certificates** **Example:** `Router# show crypto pki certificates`	(Optional) Displays the components of the certificates installed on the router if the CA certificate has been authenticated.

Troubleshooting Tips

If you ignored revocation check or expired certificates, you should carefully check your configuration. Verify that the certificate map properly matches the certificate or certificates that should be allowed or the AAA checks that should be skipped. In a controlled environment, try modifying the certificate map and determine what is not working as expected.

Configuration Examples for Setting Up Authorization and Revocation of Certificates

This section contains the following configuration examples:

- Configuring and Verifying PKI AAA Authorization: Examples, page 1450
- Configuring a Revocation Mechanism: Examples, page 1455
- Configuring a Hub Router at a Central Site for Certificate Revocation Checks: Example, page 1456

Configuring and Verifying PKI AAA Authorization: Examples

This section provides configuration examples of PKI AAA authorizations:

- Router Configuration: Example, page 1451
- Debug of a Successful PKI AAA Authorization: Example, page 1453
- Debugs of a Failed PKI AAA Authorization: Example, page 1454

Router Configuration: Example

The following **show running-config** command output shows the working configuration of a router that is set up to authorize VPN connections using the PKI Integration with AAA Server feature:

```
Router# show running-config

Building configuration...
!
version 12.3
!
hostname 7200-1
!
aaa new-model
!
!
aaa authentication login default group tacacs+
aaa authentication login no_tacacs enable
aaa authentication ppp default group tacacs+
aaa authorization exec ACSHouLab group tacacs+
aaa authorization network ACSHouLab group tacacs+
aaa accounting exec ACSHouLab start-stop group tacacs+
aaa accounting network default start-stop group ACSHouLab
aaa session-id common
!
ip domain name gril.com
!
crypto pki trustpoint EM-CERT-SERV
 enrollment url http://10.3.3.3:80
 serial-number
 crl optional
 rsakeypair STOREVPN 1024
 auto-enroll
 authorization list ACSHouLab
!
crypto pki certificate chain EM-CERT-SERV
 certificate 04
  30820214 3082017D A0030201 02020104 300D0609 2A864886 F70D0101 04050030
  17311530 13060355 0403130C 454D2D43 4552542D 53455256 301E170D 30343031
  31393232 30323535 5A170D30 35303131 38323230 3235355A 3030312E 300E0603
  55040513 07314437 45424434 301C0609 2A864886 F70D0109 02160F37 3230302D
  312E6772 696C2E63 6F6D3081 9F300D06 092A8648 86F70D01 01010500 03818D00
  30818902 818100BD F3B837AA D925F391 2B64DA14 9C2EA031 5A7203C4 92F8D6A8
  7D2357A6 BCC8596F A38A9B10 47435626 D59A8F2A 123195BB BE5A1E74 B1AA5AE0
  5CA162FF 8C3ACA4F B3EE9F27 8B031642 B618AE1B 40F2E3B4 F996BEFE 382C7283
  3792A369 236F8561 8748AA3F BC41F012 B859BD9C DB4F75EE 3CEE2829 704BD68F
  FD904043 0F555702 03010001 A3573055 30250603 551D1F04 1E301C30 1AA018A0
  16861468 7474703A 2F2F3633 2E323437 2E313037 2E393330 0B060355 1D0F0404
  030205A0 301F0603 551D2304 18301680 1420FC4B CF0B1C56 F5BD4C06 0AFD4E67
  341AE612 D1300D06 092A8648 86F70D01 01040500 03818100 79E97018 FB955108
  12F42A56 2A6384BC AC8E22FE F1D6187F DA5D6737 C0E241AC AAAEC75D 3C743F59
  08DEEFF2 0E813A73 D79E0FA9 D62DC20D 8E2798CD 2C1DC3EC 3B2505A1 3897330C
  15A60D5A 8A13F06D 51043D37 E56E45DF A65F43D7 4E836093 9689784D C45FD61D
  EC1F160C 1ABC8D03 49FB11B1 DA0BED6C 463E1090 F34C59E4
  quit
 certificate ca 01
  30820207 30820170 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  17311530 13060355 0403130C 454D2D43 4552542D 53455256 301E170D 30333132
  31363231 34373432 5A170D30 36313231 35323134 3734325A 30173115 30130603
  55040313 0C454D2D 43455254 2D534552 5630819F 300D0609 2A864886 F70D0101
  01050003 818D0030 81890281 8100C14D 833641CF D784F516 DA6B50C0 7B3CB3C9
  589223AB 99A7DC14 04F74EF2 AAEEE8F5 E3BFAE97 F2F980F7 D889E6A1 2C726C69
  54A29870 7E7363FF 3CD1F991 F5A37CFF 3FFDD3D0 9E486C44 A2E34595 C2D078BB
```

```
  E9DE981E B733B868 AA8916C0 A8048607 D34B83C0 64BDC101 161FC103 13C06500
  22D6EE75 7D6CF133 7F1B515F 32830203 010001A3 63306130 0F060355 1D130101
  FF040530 030101FF 300E0603 551D0F01 01FF0404 03020186 301D0603 551D0E04
  16041420 FC4BCF0B 1C56F5BD 4C060AFD 4E67341A E612D130 1F060355 1D230418
  30168014 20FC4BCF 0B1C56F5 BD4C060A FD4E6734 1AE612D1 300D0609 2A864886
  F70D0101 04050003 81810085 D2E386F5 4107116B AD3AC990 CBE84063 5FB2A6B5
  BD572026 528E92ED 02F3A0AE 1803F2AE AA4C0ED2 0F59F18D 7B50264F 30442C41
  0AF19C4E 70BD3CB5 0ADD8DE8 8EF636BD 24410DF4 DB62DAFC 67DA6E58 3879AA3E
  12AFB1C3 2E27CB27 EC74E1FC AEE2F5CF AA80B439 615AA8D5 6D6DEDC3 7F9C2C79
  3963E363 F2989FB9 795BA8
  quit
!
!
crypto isakmp policy 10
 encr 3des
 group 2
!
!
crypto ipsec transform-set ISC_TS_1 esp-3des esp-sha-hmac
!
crypto ipsec profile ISC_IPSEC_PROFILE_2
 set security-association lifetime kilobytes 530000000
 set security-association lifetime seconds 14400
 set transform-set ISC_TS_1
!
!
controller ISA 1/1
!
!
interface Tunnel0
 description MGRE Interface provisioned by ISC
 bandwidth 10000
 ip address 10.17.17.2 255.255.255.0
 no ip redirects
 ip mtu 1408
 ip nhrp map multicast dynamic
 ip nhrp network-id 101
 ip nhrp holdtime 500
 ip nhrp server-only
 no ip split-horizon eigrp 101
 tunnel source FastEthernet2/1
 tunnel mode gre multipoint
 tunnel key 101
 tunnel protection ipsec profile ISC_IPSEC_PROFILE_2
!
interface FastEthernet2/0
 ip address 10.1.1.1 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet2/1
 ip address 10.2.2.2 255.255.255.0
 duplex auto
 speed auto
!
!
tacacs-server host 192.43.233.55 single-connection
tacacs-server directed-request
tacacs-server key gril lab
!
ntp master 1
!
end
```

Debug of a Successful PKI AAA Authorization: Example

The following **show debugging** command output shows a successful authorization using the PKI Integration with AAA Server feature:

```
Router# show debugging

General OS:
  TACACS access control debugging is on
  AAA Authentication debugging is on
  AAA Authorization debugging is on
Cryptographic Subsystem:
Crypto PKI Trans debugging is on

Router#
May 28 19:36:11.117: CRYPTO_PKI: Trust-Point EM-CERT-SERV picked up
May 28 19:36:12.789: CRYPTO_PKI: Found a issuer match
May 28 19:36:12.805: CRYPTO_PKI: cert revocation status unknown.
May 28 19:36:12.805: CRYPTO_PKI: Certificate validated without revocation check
May 28 19:36:12.813: CRYPTO_PKI_AAA: checking AAA authorization (ACSHouLab,
POD-5.gril.com, <all>)
May 28 19:36:12.813: AAA/BIND(00000042): Bind i/f
May 28 19:36:12.813: AAA/AUTHOR (0x42): Pick method list 'ACSHouLab'
May 28 19:36:12.813: TPLUS: Queuing AAA Authorization request 66 for processing
May 28 19:36:12.813: TPLUS: processing authorization request id 66
May 28 19:36:12.813: TPLUS: Protocol set to None .....Skipping
May 28 19:36:12.813: TPLUS: Sending AV service=pki
May 28 19:36:12.813: TPLUS: Authorization request created for 66(POD-5.gril.com)
May 28 19:36:12.813: TPLUS: Using server 198.43.233.55
May 28 19:36:12.813: TPLUS(00000042)/0/NB_WAIT/203A4628: Started 5 sec timeout
May 28 19:36:12.813: TPLUS(00000042)/0/NB_WAIT: wrote entire 46 bytes request
May 28 19:36:12.813: TPLUS: Would block while reading pak header
May 28 19:36:12.817: TPLUS(00000042)/0/READ: read entire 12 header bytes (expect 27 bytes)
May 28 19:36:12.817: TPLUS(00000042)/0/READ: read entire 39 bytes response
May 28 19:36:12.817: TPLUS(00000042)/0/203A4628: Processing the reply packet
May 28 19:36:12.817: TPLUS: Processed AV cert-application=all
May 28 19:36:12.817: TPLUS: received authorization response for 66: PASS
May 28 19:36:12.817: CRYPTO_PKI_AAA: reply attribute ("cert-application" = "all")
May 28 19:36:12.817: CRYPTO_PKI_AAA: authorization passed
Router#
Router#
May 28 19:36:18.681: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 101: Neighbor 17.17.17.1 (Tunnel0) is
up: new adjacency
Router#

Router# show crypto isakmp sa

dst             src             state          conn-id slot
00.2.2.2    10.247.102.20   QM_IDLE             84    0
```

Debugs of a Failed PKI AAA Authorization: Example

The following **show debugging** command output shows that the router is not authorized to connect using VPN. The messages are typical of those that you might see in such a situation.

In this example, the peer username was configured as not authorized, which was done by moving the username to a Cisco Secure ACS group called VPN_Router_Disabled in Cisco Secure ACS. The router 7200-1.gril.com has been configured to check with a Cisco Secure ACS AAA server prior to establishing a VPN connection to any peer.

```
Router# show debugging

General OS:
  TACACS access control debugging is on
  AAA Authentication debugging is on
  AAA Authorization debugging is on
Cryptographic Subsystem:
  Crypto PKI Trans debugging is on

Router#
May 28 19:48:29.837: CRYPTO_PKI: Trust-Point EM-CERT-SERV picked up
May 28 19:48:31.509: CRYPTO_PKI: Found a issuer match
May 28 19:48:31.525: CRYPTO_PKI: cert revocation status unknown.
May 28 19:48:31.525: CRYPTO_PKI: Certificate validated without revocation check
May 28 19:48:31.533: CRYPTO_PKI_AAA: checking AAA authorization (ACSHouLab,
POD-5.gril.com, <all>)
May 28 19:48:31.533: AAA/BIND(00000044): Bind i/f
May 28 19:48:31.533: AAA/AUTHOR (0x44): Pick method list 'ACSHouLab'
May 28 19:48:31.533: TPLUS: Queuing AAA Authorization request 68 for processing
May 28 19:48:31.533: TPLUS: processing authorization request id 68
May 28 19:48:31.533: TPLUS: Protocol set to None .....Skipping
May 28 19:48:31.533: TPLUS: Sending AV service=pki
May 28 19:48:31.533: TPLUS: Authorization request created for 68(POD-5.gril.com)
May 28 19:48:31.533: TPLUS: Using server 198.43.233.55
May 28 19:48:31.533: TPLUS(00000044)/0/NB_WAIT/203A4C50: Started 5 sec timeout
May 28 19:48:31.533: TPLUS(00000044)/0/NB_WAIT: wrote entire 46 bytes request
May 28 19:48:31.533: TPLUS: Would block while reading pak header
May 28 19:48:31.537: TPLUS(00000044)/0/READ: read entire 12 header bytes (expect 6 bytes)
May 28 19:48:31.537: TPLUS(00000044)/0/READ: read entire 18 bytes response
May 28 19:48:31.537: TPLUS(00000044)/0/203A4C50: Processing the reply packet
May 28 19:48:31.537: TPLUS: received authorization response for 68: FAIL
May 28 19:48:31.537: CRYPTO_PKI_AAA: authorization declined by AAA, or AAA server not
found.
May 28 19:48:31.537: CRYPTO_PKI_AAA: No cert-application attribute found. Failing.
May 28 19:48:31.537: CRYPTO_PKI_AAA: authorization failed
May 28 19:48:31.537: CRYPTO_PKI: AAA authorization for list 'ACSHouLab', and user
'POD-5.gril.com' failed.
May 28 19:48:31.537: %CRYPTO-5-IKMP_INVAL_CERT: Certificate received from 63.247.102.20
is bad: certificate invalid
May 28 19:48:39.821: CRYPTO_PKI: Trust-Point EM-CERT-SERV picked up
May 28 19:48:41.481: CRYPTO_PKI: Found a issuer match
May 28 19:48:41.501: CRYPTO_PKI: cert revocation status unknown.
May 28 19:48:41.501: CRYPTO_PKI: Certificate validated without revocation check
May 28 19:48:41.505: CRYPTO_PKI_AAA: checking AAA authorization (ACSHouLab,
POD-5.gril.com, <all>)
May 28 19:48:41.505: AAA/BIND(00000045): Bind i/f
May 28 19:48:41.505: AAA/AUTHOR (0x45): Pick method list 'ACSHouLab'
May 28 19:48:41.505: TPLUS: Queuing AAA Authorization request 69 for processing
May 28 19:48:41.505: TPLUS: processing authorization request id 69
May 28 19:48:41.505: TPLUS: Protocol set to None .....Skipping
May 28 19:48:41.505: TPLUS: Sending AV service=pki
May 28 19:48:41.505: TPLUS: Authorization request created for 69(POD-5.gril.com)
May 28 19:48:41.505: TPLUS: Using server 192.43.233.55
```

```
May 28 19:48:41.509: TPLUS(00000045)/0/IDLE/63B22834: got immediate connect on new 0
May 28 19:48:41.509: TPLUS(00000045)/0/WRITE/63B22834: Started 5 sec timeout
May 28 19:48:41.509: TPLUS(00000045)/0/WRITE: wrote entire 46 bytes request
May 28 19:48:41.509: TPLUS(00000045)/0/READ: read entire 12 header bytes (expect 6 bytes)
May 28 19:48:41.509: TPLUS(00000045)/0/READ: read entire 18 bytes response
May 28 19:48:41.509: TPLUS(00000045)/0/63B22834: Processing the reply packet
May 28 19:48:41.509: TPLUS: received authorization response for 69: FAIL
May 28 19:48:41.509: CRYPTO_PKI_AAA: authorization declined by AAA, or AAA server not
found.
May 28 19:48:41.509: CRYPTO_PKI_AAA: No cert-application attribute found. Failing.
May 28 19:48:41.509: CRYPTO_PKI_AAA: authorization failed
May 28 19:48:41.509: CRYPTO_PKI: AAA authorization for list 'ACSHouLab', and user
'POD-5.gril.com' failed.
May 28 19:48:41.509: %CRYPTO-5-IKMP_INVAL_CERT: Certificate received from 63.247.102.20
is bad: certificate invalid
Router#

Router# show crypto iskmp sa

dst             src             state          conn-id slot
10.2.2.2       10.247.102.20   MM_KEY_EXCH         95    0
```

Configuring a Revocation Mechanism: Examples

This section contains the following configuration examples that can be used when specifying a revocation mechanism for your PKI:

- Configuring an OCSP Server: Example, page 1455
- Specifying a CRL and Then an OCSP Server: Example, page 1455
- Specifying an OCSP Server: Example, page 1455

Configuring an OCSP Server: Example

The following example shows how to configure the router to use the OCSP server that is specified in the AIA extension of the certificate:

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# revocation-check ocsp
```

Specifying a CRL and Then an OCSP Server: Example

The following example shows how to configure the router to download the CRL from the CDP. If the CRL is unavailable, the OCSP server that is specified in the AIA extension of the certificate will be used. If both options fail, certificate verification will also fail.

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# revocation-check crl ocsp
```

Specifying an OCSP Server: Example

The following example shows how to configure your router to use the OCSP server at the HTTP URL “http://myocspserver:81.” If the server is down, the revocation check will be ignored.

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# ocsp url http://myocspserver:81
Router(ca-trustpoint)# revocation-check ocsp none
```

Configuring a Hub Router at a Central Site for Certificate Revocation Checks: Example

The following example shows a hub router at a central site that is providing connectivity for several branch offices to the central site.

The branch offices are also able to communicate directly with each other using additional IPSec tunnels between the branch offices.

The CA publishes CRLs on an HTTP server at the central site. The central site checks CRLs for each peer when setting up an IPSec tunnel with that peer.

The example does not show the IPSec configuration—only the PKI-related configuration is shown.

Home Office Hub Configuration

```
crypto pki trustpoint VPN-GW
 enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
 serial-number none
 fqdn none
 ip-address none
 subject-name o=Home Office Inc,cn=Central VPN Gateway
 revocation-check crl
```

Central Site Hub Router

```
Router# show crypto ca certificate

Certificate
  Status: Available
  Certificate Serial Number: 2F62BE14000000000CA0
  Certificate Usage: General Purpose
  Issuer:
    cn=Central Certificate Authority
    o=Home Office Inc
  Subject:
    Name: Central VPN Gateway
    cn=Central VPN Gateway
    o=Home Office Inc
  CRL Distribution Points:
    http://ca.home-office.com/CertEnroll/home-office.crl
  Validity Date:
    start date: 00:43:26 GMT Sep 26 2003
    end   date: 00:53:26 GMT Sep 26 2004
    renew date: 00:00:00 GMT Jan 1 1970
  Associated Trustpoints: VPN-GW
CA Certificate
  Status: Available
  Certificate Serial Number: 1244325DE0369880465F977A18F61CA8
  Certificate Usage: Signature
  Issuer:
    cn=Central Certificate Authority
    o=Home Office Inc
  Subject:
    cn=Central Certificate Authority
    o=Home Office Inc
  CRL Distribution Points:
    http://ca.home-office.com/CertEnroll/home-office.crl
  Validity Date:
    start date: 22:19:29 GMT Oct 31 2002
    end   date: 22:27:27 GMT Oct 31 2017
  Associated Trustpoints: VPN-GW
```

Trustpoint on the Branch Office Router

```
crypto pki trustpoint home-office
 enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
 serial-number none
 fqdn none
 ip-address none
 subject-name o=Home Office Inc,cn=Branch 1
 revocation-check crl
```

A certificate map is entered on the branch office router.

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
branch1(config)# crypto pki certificate map central-site 10
branch1(ca-certificate-map)#
```

The output from the **show certificate** command on the central site hub router shows that the certificate was issued by the following:

cn=Central Certificate Authority

o=Home Office Inc

These two lines are combined into one line using a comma (,) to separate them, and the original lines are added as the first criteria for a match.

```
Router (ca-certificate-map)# issuer-name co cn=Central Certificate Authority, ou=Home
Office Inc
!The above line wrapped but should be shown on one line with the line above it.
```

The same combination is done for the subject name from the certificate on the central site router (note that the line that begins with "Name:" is not part of the subject name and must be ignored when creating the certificate map criteria). This is the subject name to be used in the certificate map.

cn=Central VPN Gateway

o=Home Office Inc

```
Router (ca-certificate-map)# subject-name eq cn=central vpn gateway, o=home office inc
```

Now the certificate map is added to the trustpoint that was configured earlier.

```
Router (ca-certificate-map)# crypto pki trustpoint home-office
Router (ca-trustpoint)# match certificate central-site skip revocation-check
Router (ca-trustpoint)# exit
Router (config)# exit
```

The configuration is checked (most of configuration is not shown).

```
Router# write term
!Many lines left out
.
.
.
crypto pki trustpoint home-office
 enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
 serial-number none
 fqdn none
 ip-address none
 subject-name o=Home Office Inc,cn=Branch 1
 revocation-check crl
 match certificate central-site skip revocation-check
!
!
```

```
crypto pki certificate map central-site 10
 issuer-name co cn = Central Certificate Authority, ou = Home Office Inc
 subject-name eq cn = central vpn gateway, o = home office inc
!many lines left out
```

Note that the issuer-name and subject-name lines have been reformatted to make them consistent for later matching with the certificate of the peer.

If the branch office is checking the AAA, the trustpoint will have lines similar to the following:

```
crypto pki trustpoint home-office
 auth list allow_list
 auth user subj commonname
```

After the certificate map has been defined as was done above, the following command is added to the trustpoint to skip AAA checking for the central site hub.

```
match certificate central-site skip authorization-check
```

In both cases, the branch site router has to establish an IPSec tunnel to the central site to check CRLs or to contact the AAA server. However, without the **match certificate** command and **central-site skip authorization-check** (argument and keyword), the branch office cannot establish the tunnel until it has checked the CRL or the AAA server. (The tunnel will not be established unless the **match certificate** command and **central-site skip authorization-check** argument and keyword are used.)

The **match certificate** command and **allow expired-certificate** keyword would be used at the central site if the router at a branch site had an expired certificate and it had to establish a tunnel to the central site to renew its certificate.

Trustpoint on the Central Site Router

```
crypto pki trustpoint VPN-GW
 enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
 serial-number none
 fqdn none
 ip-address none
 subject-name o=Home Office Inc,cn=Central VPN Gateway
 revocation-check crl
```

Trustpoint on the Branch 1 Site Router

```
Router# show crypto ca certificate

Certificate
  Status: Available
  Certificate Serial Number: 2F62BE14000000000CA0
  Certificate Usage: General Purpose
  Issuer:
    cn=Central Certificate Authority
    o=Home Office Inc
  Subject:
    Name: Branch 1 Site
    cn=Branch 1 Site
    o=Home Office Inc
  CRL Distribution Points:
    http://ca.home-office.com/CertEnroll/home-office.crl
  Validity Date:
    start date: 00:43:26 GMT Sep 26 2003
    end   date: 00:53:26 GMT Oct 3 2003
    renew date: 00:00:00 GMT Jan 1 1970
  Associated Trustpoints: home-office
CA Certificate
  Status: Available
  Certificate Serial Number: 1244325DE0369880465F977A18F61CA8
```

```
  Certificate Usage: Signature
  Issuer:
    cn=Central Certificate Authority
    o=Home Office Inc
  Subject:
    cn=Central Certificate Authority
    o=Home Office Inc
  CRL Distribution Points:
    http://ca.home-office.com/CertEnroll/home-office.crl
  Validity Date:
    start date: 22:19:29 GMT Oct 31 2002
    end   date: 22:27:27 GMT Oct 31 2017
  Associated Trustpoints: home-office
```

A certificate map is entered on the central site router.

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router (config)# crypto pki certificate map branch1 10
Router (ca-certificate-map)# issuer-name co cn=Central Certificate Authority, ou=Home
Office Inc
!The above line wrapped but should be part of the line above it.
Router (ca-certificate-map)# subject-name eq cn=Brahcn 1 Site,o=home office inc
```

The certificate map is added to the trustpoint.

```
Router (ca-certificate-map)# crypto pki trustpoint VPN-GW
Router (ca-trustpoint)# match certificate branch1 allow expired-certificate
Router (ca-trustpoint)# exit
Router (config) #exit
```

The configuration should be checked (most of the configuration is not shown).

```
Router# write term

!many lines left out

crypto pki trustpoint VPN-GW
 enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
 serial-number none
 fqdn none
 ip-address none
 subject-name o=Home Office Inc,cn=Central VPN Gateway
 revocation-check crl
 match certificate branch1 allow expired-certificate
!
!
crypto pki certificate map central-site 10
 issuer-name co cn = Central Certificate Authority, ou = Home Office Inc
 subject-name eq cn = central vpn gateway, o = home office inc
! many lines left out
```

The **match certificate** command and **branch1 allow expired-certificate** (argument and keyword) and the certificate map should be removed as soon as the branch router has a new certificate.

Additional References

The following sections provide references related to PKI certificate authorization and revocation.

Related Documents

Related Topic	Document Title
PKI commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*, Release 12.4
Overview of PKI, including RSA keys, certificate enrollment, and CAs	"Cisco IOS PKI Overview: Understanding and Planning a PKI" module
RSA key generation and deployment	"Deploying RSA Keys Within a PKI" module
Certificate enrollment: supported methods, enrollment profiles, configuration tasks	"Configuring Certificate Enrollment for a PKI" module
Cisco IOS certificate server overview information and configuration tasks	"Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment" module

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Feature Information for Certificate Authorization and Revocation

Table 59 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for a specific command was introduced, see the command reference documentation.

For information on a feature in this technology that is not documented here, see the Implementing and Managing PKI Features Roadmap.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Table 59 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 59 Feature Information for PKI Certificate Authorization and Revocation

Feature Name	Software Releases	Feature Configuration Information
Certificate Security Attribute-Based Access Control	12.2(15)T	Under the IPSec protocol, CA interoperability permits Cisco IOS devices and a CA to communicate so that the Cisco IOS device can obtain and use digital certificates from the CA. Certificates contain several fields that are used to determine whether a device or user is authorized to perform a specified action. This feature adds fields to the certificate that allow specifying an ACL, to create a certificate-based ACL. The following sections provide information about this feature: • When to Use Certificate-Based ACLs for Authorization or Revocation • Overriding Certificate Revocation and Authorization Settings The following commands were introduced or modified by this feature: **crypto pki certificate map**, **crypto pki trustpoint**, **match certificate**
Online Certificate Status Protocol (OCSP)	12.3(2)T	This feature allows users to enable OCSP instead of CRLs to check certificate status. Unlike CRLs, which provide only periodic certificate status, OCSP can provide timely information regarding the status of a certificate. The following sections provide information about this feature: • CRLs or OCSP Server: Choosing a Certificate Revocation Mechanism • Configuring a Revocation Mechanism for Cisco IOS Certificate Status Checking The following commands were introduced by this feature: **ocsp url**, **revocation-check**
PKI AAA Authorization Using the Entire Subject Name	12.3(11)T	This feature provides users with the ability to query the AAA server using the entire subject name from the certificate as a unique AAA username. The following sections provide information about this feature: • Attribute-Value Pairs for PKI and AAA Server Integration • Configuring PKI Integration with a AAA Server The following command was modified by this feature: **authorization username**

Table 59 Feature Information for PKI Certificate Authorization and Revocation (continued)

Feature Name	Software Releases	Feature Configuration Information
PKI Integration with AAA Server	12.3(1)	This feature provides additional scalability for authorization by generating a AAA username from the certificate presented by the peer. A AAA server is queried to determine whether the certificate is authorized for use by the internal component. The authorization is indicated by a component-specified label that must be present in the AV pair for the user. The following sections provide information about this feature: • PKI and AAA Server Integration for Certificate Status • Configuring PKI Integration with a AAA Server The following commands were introduced by this feature: **authorization list**, **authorization username**

Table 59 Feature Information for PKI Certificate Authorization and Revocation (continued)

Feature Name	Software Releases	Feature Configuration Information
PKI: Query Multiple Servers During Certificate Revocation Check	12.3(7)T	This feature introduces the ability for Cisco IOS software to make multiple attempts to retrieve the CRL, allowing operations to continue when a particular server is not available. In addition, the ability to override the CDPs in a certificate with a manually configured CDP has been introduced. Manually overriding the CDPs in a certificate can be advantageous when a particular server is unavailable for an extended period of time. The certificate's CDPs can be replaced with a URL or directory specification without reissuing all of the certificates that contain the original CDP. The following sections provide information about this feature: • Querying All CDPs During Revocation Check • Manually Overriding CDPs in a Certificate The following command was introduced by this feature: **match certificate override cdp**
Using Certificate ACLs to Ignore Revocation Check and Expired Certificates	12.3(4)T	This feature allows a certificate that meets specified criteria to be accepted regardless of the validity period of the certificate, or if the certificate meets the specified criteria, revocation checking does not have to be performed. Certificate ACLs are used to specify the criteria that the certificate must meet to be accepted or to avoid revocation checking. In addition, if AAA communication is protected by a certificate, this feature provides for the AAA checking of the certificate to be ignored. The following sections provide information about this feature: • Ignore Revocation Checks Using a Certificate-Based ACL • Overview: Configuring Certificate-Based ACLs to Ignore Revocation Checks The following command was modified by this feature: **match certificate**

Configuring Certificate Enrollment for a PKI

Certificate enrollment, which is the process of obtaining a certificate from a certification authority (CA), occurs between the end host requesting the certificate and the CA. Each peer that participates in the public key infrastructure (PKI) must enroll with a CA. This module describes the different methods available for certificate enrollment and how to set up each method for a participating PKI peer.

Module History

This module was first published on May 2, 2005, and last updated on May 2, 2005.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all features. To find information about feature support and configuration, use the "Feature Information for PKI Certificate Enrollment" section on page 1489.

Contents

- Prerequisites for PKI Certificate Enrollment, page 1465
- Information About Certificate Enrollment for a PKI, page 1466
- How to Configure Certificate Enrollment for a PKI, page 1469
- Configuration Examples for PKI Certificate Enrollment Requests, page 1485
- Additional References, page 1489
- Feature Information for PKI Certificate Enrollment, page 1489

Prerequisites for PKI Certificate Enrollment

Before configuring peers for certificate enrollment, you should have the following items:

- A generated Rivest, Shamir, and Adelman (RSA) key pair to enroll and a PKI in which to enroll.
- Your CA should be authenticated.
- Familiarity with the module "Cisco IOS PKI Overview: Understanding and Planning a PKI."

"crypto ca" to "crypto pki" command-line interface (CLI) change

As of Cisco IOS Release 12.3(7)T, all commands that begin as "crypto ca" have been changed to begin as "crypto pki." Although the router will still accept crypto ca commands, all output will be read back as crypto pki.

Information About Certificate Enrollment for a PKI

Before configuring peers to request a certificate and enroll in the PKI, you should understand the following concepts:

- What Are CAs?, page 1466
- Authentication of the CA, page 1467
- Supported Certificate Enrollment Methods, page 1467
- Registration Authorities, page 1468
- Automatic Certificate Enrollment, page 1468
- Certificate Enrollment Profiles, page 1469

What Are CAs?

A CA, also known as a trustpoint, manages certificate requests and issues certificates to participating network devices. These services (managing certificate requests and issuing certificates) provide centralized key management for the participating devices and are explicitly trusted by the receiver to validate identities and to create digital certificates. Before any PKI operations can begin, the CA generates its own public key pair and creates a self-signed CA certificate; thereafter, the CA can sign certificate requests and begin peer enrollment for the PKI.

You can use a CA provided by a third-party CA vendor, or you can use an "internal" CA, which is the Cisco IOS Certificate Server.

Hierarchical PKI: Multiple CAs

PKI can be set up in a hierarchical framework to support multiple CAs. At the top of the hierarchy is a root CA, which holds a self-signed certificate. The trust within the entire hierarchy is derived from the RSA key pair of the root CA. The subordinate CAs within the hierarchy can be enrolled with either the root CA or with another subordinate CA. Multiple tiers of CAs are configured by either the root CA or with another subordinate CA. Within a hierarchical PKI, all enrolled peers, can validate the certificate of one another if the peers share a trusted root CA certificate or a common subordinate CA.

When to Use Multiple CAs

Multiple CAs provide users with added flexibility and reliability. For example, subordinate CAs can be placed in branch offices while the root CA is at the office headquarters. Also, different granting policies can be implemented per CA, so you can set up one CA to automatically grant certificate requests while another CA within the hierarchy requires each certificate request to be manually granted.

Scenarios in which at least a two-tier CA is recommended are as follows:

- Large and very active networks in which a large number of certificates are revoked and reissued. A multiple tier CA helps to control the size of the certificate revocation lists (CRLs).
- When online enrollment protocols are used, the root CA can be kept offline except to issue subordinate CA certificates. This scenario provides added security for the root CA.

Authentication of the CA

Before certificate enrollment can occur, the certificate of the CA must be authenticated before the device can be issued its own certificate. Authentication of the CA typically occurs only when you initially configure CA support at your router. To authenticate the CA, issue the **crypto pki authenticate** command, which authenticates the CA to your router by obtaining the self-signed certificate of the CA that contains the public key of the CA.

Authentication via the fingerprint Command

After Cisco IOS Release 12.3(12), you can issue the **fingerprint** command to preenter a fingerprint that can be matched against the fingerprint of a CA certificate during authentication.

A fingerprint is not preentered for a trustpoint, and if the authentication request is interactive, you must verify the fingerprint that is displayed during authentication of the CA certificate. If the authentication request is noninteractive, the certificate will be rejected without a preentered fingerprint.

Note If the authentication request is made using the command-line interface (CLI), the request is an interactive request. If the authentication request is made using HTTP or another management tool, the request is a noninteractive request.

Supported Certificate Enrollment Methods

Cisco IOS software supports the following methods to obtain a certificate from a CA:

- Simple Certificate Enrollment Protocol (SCEP)—A Cisco proprietary enrollment protocol that uses HTTP to communicate with the CA or registration authority (RA). SCEP is the most commonly used method for sending and receiving requests and certificates.
- PKCS12—The router imports certificates in PKCS12 format from an external server.
- IFS (IOS File System)—The router uses any file system that is supported by Cisco IOS software (such as TFTP, FTP, flash, and NVRAM) to send a certificate request and to receive the issued certificate. Users may enable IFS certificate enrollment when their CA does not support SCEP.

  Note Prior to Cisco IOS Release 12.3(4)T, only the TFTP file system is supported within IFS.

- Manual (cut-and-paste)—The router displays the certificate request on the console terminal, allowing the user to enter the issued certificate on the terminal. A user may manually cut-and-paste certificate requests and certificates when he or she does not have a network connection between the router and CA.
- Enrollment profiles—The router sends HTTP-based enrollment requests directly to the CA server instead of the RA proxy. Enrollment profiles can be used if a CA server does not support SCEP and the user does not want to use an RA as a proxy.
- Self-signed certificate enrollment for a trustpoint—The secure HTTP (HTTPS) server generates a self-signed certificate that is to be used during the secure socket layer (SSL) handshake, establishing a secure connection between the HTTPS server and the client. The self-signed certificate is then saved in the router's startup configuration (NVRAM). The saved, self-signed certificate can then be used for future SSL handshakes, eliminating the user intervention that was necessary to accept the certificate every time the router reloaded.

Registration Authorities

Some CA servers do not support SCEP directly; thus, an RA has to process the SCEP request for the CA. An RA is essentially a server that acts as a proxy for the CA so that CA functions can continue when the CA is offline. Although the RA is often part of the CA server, the RA can also be an additional application, requiring an additional device to run it.

Automatic Certificate Enrollment

Certificate autoenrollment allows the router to automatically request a certificate from the CA sever, thereby, eliminating operator intervention when the enrollment request is sent to the CA server.

Automatic enrollment is performed on startup for any trustpoint CA that is configured and that does not have a valid client certificate. (There must be a valid server certificate.) When the certificate expires, a new certificate is automatically requested. Although automatic enrollment does not provide seamless certificate renewal, it does provide unattended recovery from expiration.

Note When automatic enrollment is configured, clients automatically request client certificates. The CA server performs its own authorization checks; if these checks include a policy to automatically issue certificates, all clients will automatically receive certificates, which is not very secure. Thus, automatic certificate enrollment should be combined with additional authentication and authorization mechanisms (such as Secure Device Provisioning (SDP), leveraging existing certificates, and one-time passwords).

Certificate Autoenrollment with Key Rollover

By default, the automatic certificate enrollment function requests a new certificate when the old certificate expires. Connectivity can be lost while the request is being serviced because the existing certificate and key pairs are deleted immediately after the new key is generated. The new key does not have a certificate to match it until the process is complete, and incoming Internet Key Exchange (IKE) connections cannot be established until the new certificate is issued. Key rollover allows the certificate renewal request to be made before the certificate expires by retaining the old key and certificate until the new certificate is available.

Key Regeneration

An optional renewal percentage parameter can be used to allow a new certificate to be requested when a specified percentage of the lifetime of the certificate has passed. For example, if the renewal percentage is configured as 90 and the certificate has a lifetime of one year, a new certificate is requested 36.5 days before the old certificate expires.

The **regenerate** keyword of the **auto-enroll** command provides seamless key rollover by creating a new key pair with a temporary name and retaining the old certificate and key pair until a new certificate is received from the CA. When the new certificate is received, the old certificate and key pair are discarded, and the new key pair is renamed with the name of the original key pair.

Certificate Enrollment Profiles

Enrollment profiles allow users to specify certificate authentication, enrollment, and reenrollment parameters when prompted. The values for these parameters are referenced by two templates that make up the profile. One template contains parameters for the HTTP request that is sent to the CA server to obtain the certificate of the CA (also known as certificate authentication); the other template contains parameters for the HTTP request that is sent to the CA for certificate enrollment.

Configuring two templates enables users to specify different URLs or methods for certificate authentication and enrollment; for example, authentication (getting the certificate of the CA) can be performed via TFTP (using the **authentication url** command) while enrollment can be performed manually (using the **enrollment terminal** command).

Prior to Cisco IOS Release 12.3(11)T, certificate requests could be sent only in a PKCS10 format; however, an additional parameter has now been added to the profile, allowing users to specify the PKCS7 format for certificate renewal requests.

Note A single enrollment profile can have up to three separate sections for each task—certificate authentication, enrollment, and reenrollment.

How to Configure Certificate Enrollment for a PKI

This section contains the following enrollment option procedures. Note that if you configure enrollment or autoenrollment (the first task), you cannot configure manual certificate enrollment. Also, if you configure manual certificate enrollment, you cannot configure an enrollment profile.

- Configuring Certificate Enrollment or Autoenrollment, page 1469
- Configuring Manual Certificate Enrollment, page 1473
- Configuring a Persistent Self-Signed Certificate for Enrollment via SSL, page 1478
- Configuring a Certificate Enrollment Profile for Enrollment or Reenrollment, page 1481

Configuring Certificate Enrollment or Autoenrollment

Perform this task to configure certificate enrollment or automatic certificate enrollment for peers participating in your PKI.

Prerequisites for Autoenrollment

Before configuring automatic certificate enrollment requests, ensure that all necessary enrollment information is configured.

Restrictions for Autoenrollment

RSA Key Pair Restriction for Autoenrollment

Trustpoints configured to generate a new key pair using the **regenerate** command or the **regenerate** keyword of the **auto-enroll** command must not share key pairs with other trustpoints. To give each trustpoint its own key pair, use the **rsakeypair** command in ca-trustpoint configuration mode. Sharing key pairs among regenerating trustpoints is not supported and will cause loss of service on some of the trustpoints because of key and certificate mismatches.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment** [**mode**] [**retry period** *minutes*] [**retry count** *number*] **url** *url* [**pem**]
5. **subject-name** [*x.500-name*]
6. **ip address** {*ip address* | *interface* | **none**}
7. **serial-number** [**none**]
8. **auto-enroll** [*percent*] [**regenerate**]
9. **usage** *method1* [*method2* [*method3*]]
10. **password** *string*
11. **rsakeypair** *key-label* [*key-size* [*encryption-key-size*]]
12. **fingerprint** *ca-fingerprint*
13. **exit**
14. **crypto pki authenticate** *name*
15. **exit**
16. **copy system:running-config nvram:startup-config**
17. **show crypto pki certificates**
18. **show crypto pki trustpoints** [**status** | *label* [**status**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint mytp`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	**enrollment** [**mode**] [**retry period** *minutes*] [**retry count** *number*] **url** *url* [**pem**] **Example:** `Router(ca-trustpoint)# enrollment url http://cat.example.com`	Specifies the URL of the CA on which your router should send certificate requests. • **mode**—Specifies RA mode if your CA system provides an RA. • **retry period** *minutes*—Specifies the wait period between certificate request retries. The default is 1 minute between retries. • **retry count** *number*— Specifies the number of times a router will resend a certificate request when it does not receive a response from the previous request. (Specify from 1 to 100 retries.) • **url** *url*—URL of the file system where your router should send certificate requests. For enrollment method options, see the **enrollment** command in the *Cisco IOS Security Command Reference*. • **pem**—Adds privacy-enhanced mail (PEM) boundaries to the certificate request.
Step 5	**subject-name** [*x.500-name*] **Example:** `Router(ca-trustpoint)# subject-name cat`	(Optional) Specifies the requested subject name that will be used in the certificate request. • *x.500-name*—If it is not specified, the fully qualified domain name (FQDN), which is the default subject name, will be used.
Step 6	**ip address** {*ip address* | *interface* | **none**} **Example:** `Router(ca-trustpoint)# ip address 192.168.1.66`	(Optional) Includes the IP address of the specified interface in the certificate request. Issue the **none** keyword if no IP address should be included. **Note** If this command is enabled, you will not be prompted for an IP address during enrollment for this trustpoint.
Step 7	**serial-number** [**none**] **Example:** `Router(ca-trustpoint)# serial-number`	(Optional) Specifies the router serial number in the certificate request, unless the **none** keyword is issued.

	Command or Action	Purpose
Step 8	**auto-enroll** [*percent*] [**regenerate**] **Example:** `Router(ca-trustpoint)# auto-enroll`	(Optional) Enables autoenrollment, allowing you to automatically request a router certificate from the CA. • By default, only the Domain Name System (DNS) name of the router is included in the certificate. • Use the *percent* argument to specify that a new certificate will be requested after the percentage of the lifetime of the current certificate is reached. • Use the **regenerate** keyword to generate a new key for the certificate even if a named key already exists. **Note** If the key pair being rolled over is exportable, the new key pair will also be exportable. The following comment will appear in the trustpoint configuration to indicate whether the key pair is exportable: "! RSA key pair associated with trustpoint is exportable."
Step 9	**usage** *method1* [*method2* [*method3*]] **Example:** `Router(ca-trustpoint)# usage ssl-client`	(Optional) Specifies the intended use for the certificate. Available options are **ike**, **ssl-client**, and **ssl-server**; the default is **ike**.
Step 10	**password** *string* **Example:** `Router(ca-trustpoint)# password meow`	(Optional) Specifies the revocation password for the certificate. If this command is enabled, you will not be prompted for a password during enrollment for this trustpoint. **Note** When SCEP is used, this password can be used to authorize the certificate request—often via a one-time password or similar mechanism.
Step 11	***rsakeypair*** *key-label* [*key-size* [*encryption-key-size*]] **Example:** `Router(ca-trustpoint)# rsakeypair cat`	(Optional) Specifies which key pair to associate with the certificate. • A key pair with *key-label* will be generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command was issued. • Specify the *key-size* argument for generating the key, and specify the *encryption-key-size* argument to request separate encryption, signature keys, and certificates. **Note** If this command is not enabled, the FQDN key pair is used.
Step 12	**fingerprint** *ca-fingerprint* **Example:** `Router(ca-trustpoint)# fingerprint 12EF53FA 355CD23E 12EF53FA 355CD23E`	(Optional) Specifies a fingerprint that can be matched against the fingerprint of a CA certificate during authentication. **Note** If the fingerprint is not provided and authentication of the CA certificate is interactive, the fingerprint will be displayed for verification.

	Command or Action	Purpose
Step 13	**exit** **Example:** `Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 14	**crypto pki authenticate** *name* **Example:** `Router(config)# crypto pki authenticate mytp`	Retrieves the CA certificate and authenticates it. • Check the certificate fingerprint if prompted. **Note** This command is optional if the CA certificate is already loaded into the configuration
Step 15	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 16	**copy system:running-config nvram:startup-config** **Example:** `Router# copy system:running-config nvram:startup-config`	(Optional) Copies the running configuration to the NVRAM startup configuration. **Note** Autoenroll will not update NVRAM if the running configuration has been modified but not written to NVRAM.
Step 17	**show crypto pki certificates** **Example:** `Router# show crypto pki certificates`	(Optional) Displays information about your certificates, the certificates of the CA, and RA certificates.
Step 18	**show crypto pki trustpoints** [**status** | *label* [**status**]] **Example:** `Router# show crypto pki trustpoints mytp status`	(Optional) Displays the trustpoints that are configured in the router. • The **status** keyword displays how enrollment is proceeding, when any change will occur, and what error conditions exist.

Configuring Manual Certificate Enrollment

Manual certificate enrollment can be set up via TFTP or the cut-and-paste method. Both options can be used if your CA does not support SCEP or if a network connection between the router and CA is not possible. Perform one of the following tasks to set up manual certificate enrollment:

- Configuring Cut-and-Paste Certificate Enrollment, page 1474
- Configuring TFTP Certificate Enrollment, page 1476

PEM-Formatted Files for Certificate Enrollment Requests

Using PEM-formatted files for certificate requests can be helpful for customers who are using terminal or profile-based enrollment to request certificates from their CA server. Customers using PEM-formatted files can directly use existing certificates on their Cisco IOS routers.

Restrictions for Manual Certificate Enrollment

Switching Enrollment URLs When Using SCEP

Cisco does not recommend switching URLs if SCEP is used; that is, if the enrollment URL is "http://," do not change the enrollment URL between getting the CA certificate and enrolling the certificate. A user can switch between TFTP and cut-and-paste; for example, a user can paste the CA certificate via the **enrollment terminal** command and then enter **no enrollment terminal** and **enrollment url** *tftp://certserver/file_specification* to TFTP the requests and router certificates.

Key Rollover Restriction

Do not regenerate the keys manually using the **crypto key generate** command; key rollover will occur when the **crypto pki enroll** command is issued.

Configuring Cut-and-Paste Certificate Enrollment

Perform this task to configure manual certificate enrollment via the cut-and-paste method for peers participating in your PKI.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment terminal** [**pem**]
5. **fingerprint** *ca-fingerprint*
6. **exit**
7. **crypto pki authenticate** *name*
8. **crypto pki enroll** *name*
9. **crypto pki import** *name* **certificate**
10. **exit**
11. **show crypto pki certificates**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint mytp`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 4	**enrollment terminal** [**pem**] **Example:** `Router(ca-trustpoint)# enrollment terminal`	Specifies manual cut-and-paste certificate enrollment. • **pem**—Configures the trustpoint to generate PEM-formatted certificate requests to the terminal console.
Step 5	**fingerprint** *ca-fingerprint* **Example:** `Router(ca-trustpoint)# fingerprint 12EF53FA 355CD23E 12EF53FA 355CD23E`	(Optional) Specifies a fingerprint that can be matched against the fingerprint of a CA certificate during authentication. **Note** If the fingerprint is not provided, it will be displayed for verification.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 7	**crypto pki authenticate** *name* **Example:** `Router(config)# crypto pki authenticate mytp`	Retrieves the CA certificate and authenticates it.
Step 8	**crypto pki enroll** *name* **Example:** `Router(config)# crypto pki enroll mytp`	Generates certificate request and displays the request for copying and pasting into the certificate server. Thereafter, you can manually import the certificate at the terminal in the following step.
Step 9	**crypto pki import** *name* **certificate** **Example:** `Router(config)# crypto pki import mytp certificate`	Imports a certificate manually at the terminal. You must enter this command twice if usage keys (signature and encryption keys) are used. The first time the command is entered, one of the certificates is pasted into the router; the second time the command is entered, the other certificate is pasted into the router. (It does not matter which certificate is pasted first.)
Step 10	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 11	**show crypto pki certificates** **Example:** `Router# show crypto pki certificates`	(Optional) Displays information about your certificates, the certificates of the CA, and RA certificates.

Configuring TFTP Certificate Enrollment

Perform this task to configure manual certificate enrollment using a TFTP server.

Prerequisites for TFTP Certificate Enrollment

- You must know the correct URL to use if you are configuring certificate enrollment via TFTP.
- The router must be able to write a file to the TFTP server for the **crypto pki enroll** command.

Caution Some TFTP servers require that the file must exist on the server before it can be written.

Most TFTP servers require that the file be "write-able" by the world. This requirement may pose a risk because any router or other device may write or overwrite the certificate request; thus, the replacement certificate request will not used by the CA administrator, who must first check the enrollment request fingerprint before granting the certificate request.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment** [**mode**] [**retry period** *minutes*] [**retry count** *number*] **url** *url* [**pem**]
5. **fingerprint** *ca-fingerprint*
6. **exit**
7. **crypto pki authenticate** *name*
8. **crypto pki enroll** *name*
9. **crypto pki import** *name* **certificate**
10. **exit**
11. **show crypto pki certificates**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint mytp`	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.

	Command or Action	Purpose
Step 4	**enrollment** [**mode**] [**retry period** *minutes*] [**retry count** *number*] **url** *url* [**pem**] **Example:** `Router(ca-trustpoint)# enrollment url tftp://certserver/enrollment_parms`	Specifies enrollment parameters for your CA. • For TFTP, the *url* argument must be in the form tftp://certserver/file_specification.
Step 5	**fingerprint** *ca-fingerprint* **Example:** `Router(ca-trustpoint)# fingerprint 12EF53FA 355CD23E 12EF53FA 355CD23E`	(Optional) Specifies the fingerprint of the CA certificate received via an out-of-band method from the CA administrator. **Note** If the fingerprint is not provided, it will be displayed for verification.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 7	**crypto pki authenticate** *name* **Example:** `Router(config)# crypto pki authenticate mytp`	Retrieves the CA certificate and authenticates it.
Step 8	**crypto pki enroll** *name* **Example:** `Router(config)# crypto pki enroll mytp`	Generates certificate request and writes the request out to the TFTP server.
Step 9	**crypto pki import** *name* **certificate** **Example:** `Router(config)# crypto pki import mytp certificate`	Imports a certificate via TFTP at the terminal. You must enter this command twice if usage keys (signature and encryption keys) are used. The first time the command is entered, one of the certificates is pasted into the router; the second time the command is entered, the other certificate is pasted into the router. (It does not matter which certificate is pasted first.)
Step 10	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 11	**show crypto pki certificates** **Example:** `Router# show crypto pki certificates`	(Optional) Displays information about your certificates, the certificates of the CA, and RA certificates.

Configuring a Persistent Self-Signed Certificate for Enrollment via SSL

This section contains the following concepts and tasks:

- Persistent Self-Signed Certificates Overview, page 1478
- Configuring a Trustpoint and Specifying Self-Signed Certificate Parameters, page 1478
- Enabling the HTTPS Server, page 1480

Note These tasks are optional because if you enable the HTTPS server, it generates a self-signed certificate automatically using default values.

Persistent Self-Signed Certificates Overview

The SSL protocol can be used to establish a secure connection between an HTTPS server and a client (web browser). During the SSL handshake, the client expects the SSL server's certificate to be verifiable using a certificate the client already possesses.

If Cisco IOS software does not have a certificate that the HTTPS server can use, the server generates a self-signed certificate by calling a PKI application programming interface (API). When the client receives this self-signed certificate and is unable to verify it, intervention is needed. The client asks you if the certificate should be accepted and saved for future use. If you accept the certificate, the SSL handshake continues.

Future SSL handshakes between the same client and the server use the same certificate. However, if the router is reloaded, the self-signed certificate is lost. The HTTPS server must then create a new self-signed certificate. This new self-signed certificate does not match the previous certificate so you are once again asked to accept it.

Requesting acceptance of the router's certificate each time that the router reloads may present an opportunity for an attacker to substitute an unauthorized certificate when you are being asked to accept the certificate. Persistent self-signed certificates overcome all these limitations by saving a certificate in the router's startup configuration.

Restrictions

You can configure only one trustpoint for a persistent self-signed certificate.

Configuring a Trustpoint and Specifying Self-Signed Certificate Parameters

Perform the following task to configure a trustpoint and specify self-signed certificate parameters.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment selfsigned**
5. **subject-name** [*x.500-name*]
6. **rsakeypair** *key-label* [*key-size* [*encryption-key-size*]]

7. **crypto pki enroll** *name*
8. **end**
9. **show crypto pki certificates** [*trustpoint-name* [**verbose**]]
10. **show crypto pki trustpoints** [**status** | *label* [**status**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint local`	Declares the CA that your router should use and enters ca-trustpoint configuration mode. **Note** Effective with Cisco IOS Release 12.3(8)T, the **crypto pki trustpoint** command replaced the **crypto ca trustpoint** command.
Step 4	**enrollment selfsigned** **Example:** `Router(ca-trustpoint)# enrollment selfsigned`	Specifies self-signed enrollment.
Step 5	**subject-name** [*x.500-name*] **Example:** `Router(ca-trustpoint)# subject-name`	(Optional) Specifies the requested subject name to be used in the certificate request. • If the *x-500-name* argument is not specified, the FQDN, which is the default subject name, is used.
Step 6	**rsakeypair** *key-label* [*key-size* [*encryption-key-size*]] **Example:** `Router(ca-trustpoint)# rsakeypair examplekeys 1024 1024`	(Optional) Specifies which key pair to associate with the certificate. • The *key-label* argument will be generated during enrollment if it does not already exist or if the **auto-enroll regenerate** command was issued. • Specify the *key-size* argument for generating the key, and specify the *encryption-key-size* argument to request separate encryption, signature keys, and certificates. **Note** If this command is not enabled, the FQDN key pair is used.
Step 7	**crypto pki enroll** *name* **Example:** `Router(ca-trustpoint)# crypto pki enroll local`	Tells the router to generate the persistent self-signed certificate.

	Command or Action	Purpose
Step 8	**end** **Example:** `Router(ca-trustpoint)# end` **Example:** `Router(config)# end`	(Optional) Exits ca-trustpoint configuration mode and global configuration mode.
Step 9	**show crypto pki certificates** [*trustpoint-name* [**verbose**]] **Example:** `Router# show crypto pki certificates local verbose`	Displays information about your certificate, the certification authority certificate, and any registration authority certificates.
Step 10	**show crypto pki trustpoints** [**status** | *label* [**status**]] **Example:** `Router# show crypto pki trustpoints status`	Displays the trustpoints that are configured in the router.

Enabling the HTTPS Server

Perform the following task to enable the HTTPS server.

Prerequisites

To specify parameters, you must create a trustpoint and configure it. To use default values, delete any existing self-signed trustpoints. Deleting all self-signed trustpoints causes the HTTPS server to generate a persistent self-signed certificate using default values as soon as the server is enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http secure-server**
4. **end**
5. **copy system:running-config nvram: startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip http secure-server** **Example:** `Router(config)# ip http secure-server`	Enables the secure HTTP web server. **Note** A key pair (modulus 1024) and a certificate are generated.
Step 4	**end** **Example:** `Router(config)# end`	Exits global configuration mode.
Step 5	**copy system:running-config nvram: startup-config** **Example:** `Router# copy system:running-config nvram: startup-config`	Saves the self-signed certificate and the HTTPS server in enabled mode.

Configuring a Certificate Enrollment Profile for Enrollment or Reenrollment

Perform this task to configure an enrollment profile for certificate enrollment or reenrollment of a router that is already enrolled with a third-party vendor CA but wants to enroll with a Cisco IOS CA.

Enable a router that is enrolled with a third-party vendor CA to use its existing certificate to enroll with the Cisco IOS certificate server so the enrollment request is automatically granted. To enable this functionality, you must issue the **enrollment credential** command. Also, you cannot configure manual certificate enrollment.

Prerequisites

Before configuring a certificate enrollment profile for the client router that is already enrolled with a third party vendor CA so that the router can reenroll with a Cisco IOS certificate server, you should have already performed the following tasks at the client router:

- Defined a trustpoint that points to the third-party vendor CA.
- Authenticated and enrolled the client router with the third party-vendor CA.

Restrictions

- To use certificate profiles, your network must have an HTTP interface to the CA.
- If an enrollment profile is specified, an enrollment URL may not be specified in the trustpoint configuration. Although both commands are supported, only one command can be used at a time in a trustpoint.
- Because there is no standard for the HTTP commands used by various CAs, the user is required to enter the command that is appropriate to the CA that is being used.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment profile** *label*
5. **exit**
6. **crypto pki profile enrollment** *label*
7. **authentication url** *url*

   or

   authentication terminal
8. **authentication command**
9. **enrollment url** *url*

   or

   enrollment terminal
10. **enrollment credential** *label*
11. **enrollment command**
12. **parameter** *number* {**value** *value* | **prompt** *string*}
13. **exit**
14. **show crypto pki certificates**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`configure terminal` **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint Entrust`	Declares the trustpoint and a given name and enter ca-trustpoint configuration mode.
Step 4	**enrollment profile** *label* **Example:** `Router(ca-trustpoint)# enrollment profile E`	Specifies that an enrollment profile is to be used for certificate authentication and enrollment.
Step 5	**exit** **Example:** `Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode.
Step 6	**crypto pki profile enrollment** *label* **Example:** `Router(config)# crypto pki profile enrollment E`	Defines an enrollment profile and enters ca-profile-enroll configuration mode. • *label*—Name for the enrollment profile; the enrollment profile name must match the name specified in the **enrollment profile** command.
Step 7	**authentication url** *url* **Example:** `Router(ca-profile-enroll)# authentication url http://entrust:81` or **authentication terminal** **Example:** `Router(ca-profile-enroll)# authentication terminal`	Specifies the URL of the CA server to which to send certificate authentication requests. • *url*—URL of the CA server to which your router should send authentication requests. If using HTTP, the URL should read "http://CA_name," where CA_name is the host DNS name or IP address of the CA. If using TFTP, the URL should read "tftp://certserver/file_specification." (If the URL does not include a file specification, the FQDN of the router will be used.) Specifies manual cut-and-paste certificate authentication.
Step 8	**authentication command** **Example:** `Router(ca-profile-enroll)# authentication command`	(Optional) Specifies the HTTP command that is sent to the CA for authentication. This command should be used after the **authentication url** command has been entered.

	Command or Action	Purpose
Step 9	**enrollment url** *url* **Example:** `Router(ca-profile-enroll)# enrollment url http://entrust:81/cda-cgi/clientcgi.exe` or **enrollment terminal** **Example:** `Router(ca-profile-enroll)# enrollment terminal`	Specifies the URL of the CA server to which to send certificate enrollment requests via HTTP or TFTP. Specifies manual cut-and-paste certificate enrollment.
Step 10	**enrollment credential** *label* **Example:** `Router(ca-profile-enroll)# enrollment credential Entrust`	(Optional) Specifies the third-party vendor CA trustpoint that is to be enrolled with the Cisco IOS CA. **Note** This command cannot be issued if manual certificate enrollment is being used.
Step 11	**enrollment command** **Example:** `Router(ca-profile-enroll)# enrollment command`	(Optional) Specifies the HTTP command that is sent to the CA for enrollment.
Step 12	**parameter** *number* {**value** *value* | **prompt** *string*} **Example:** `Router(ca-profile-enroll)# parameter 1 value aaaa-bbbb-cccc`	(Optional) Specifies parameters for an enrollment profile. This command can be used multiple times to specify multiple values.
Step 13	**exit** **Example:** `Router(ca-profile-enroll)# exit` `Router(config)# exit`	Enter this command two times—one time to exit ca-profile-enroll configuration mode and the second time to exit global configuration mode.
Step 14	**show crypto pki certificates** **Example:** `Router# show crypto pki certificates`	(Optional) Displays information about your certificates, the certificates of the CA, and RA certificates.

What to Do Next

If you configured the router to reenroll with a Cisco IOS CA, you must now configure the Cisco IOS certificate server to accept enrollment requests only from clients already enrolled with the specified third party vendor CA trustpoint. For more information, see the module "Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment."

Configuration Examples for PKI Certificate Enrollment Requests

This section contains the following configuration examples:

- Configuring Autoenrollment: Example, page 1485
- Configuring Certificate Autoenrollment with Key Rollover: Example, page 1486
- Configuring Manual Certificate Enrollment with Key Rollover: Example, page 1486
- Creating and Verifying a Persistent Self-Signed Certificate: Example, page 1486
- Configuring Direct HTTP Enrollment: Example, page 1488

Configuring Autoenrollment: Example

The following example shows how to configure the router to autoenroll with a CA on startup and how to specify all necessary enrollment information in the configuration:

```
crypto pki trustpoint frog
 enrollment url http://frog.phoobin.com/
 subject-name OU=Spiral Dept., O=tiedye.com
 ip-address ethernet-0
 serial-number none
 usage ike
 auto-enroll regenerate
 password revokeme
 rsa-key frog 2048
!
crypto pki certificate chain frog
certificate pki 0B
30820293 3082023D A0030201 0202010B 300D0609 2A864886 F70D0101 04050030
79310B30 09060355 04061302 5553310B 30090603 55040813 02434131 15301306
0355040A 130C4369 73636F20 53797374 656D3120 301E0603 55040B13 17737562
6F726420 746F206B 6168756C 75692049 50495355 31243022 06035504 03131B79
6E692D75 31302043 65727469 66696361 7465204D 616E6167 6572301E 170D3030
30373134 32303536 32355A17 0D303130 37313430 31323834 335A3032 310E300C
06035504 0A130543 6973636F 3120301E 06092A86 4886F70D 01090216 11706B69
2D343562 2E636973 636F2E63 6F6D305C 300D0609 2A864886 F70D0101 01050003
4B003048 024100B3 0512A201 3B4243E1 378A9703 8AC5E3CE F77AF987 B5A422C4
15E947F6 70997393 70CF34D6 63A86B9C 4347A81A 0551FC02 ABA62360 01EF7DD2
6C136AEB 3C6C3902 03010001 A381F630 81F3300B 0603551D 0F040403 02052030
1C060355 1D110415 30138211 706B692D 3435622E 63697363 6F2E636F 6D301D06
03551D0E 04160414 247D9558 169B9A21 23D289CC 2DDA2A9A 4F77C616 301F0603
551D2304 18301680 14BD742C E892E819 1D551D91 683F6DB2 D8847A6C 73308185
0603551D 1F047E30 7C307AA0 3CA03AA4 38303631 0E300C06 0355040A 13054369
73636F31 24302206 03550403 131B796E 692D7531 30204365 72746966 69636174
65204D61 6E616765 72A23AA4 38303631 0E300C06 0355040A 13054369 73636F31
24302206 03550403 131B796E 692D7531 30204365 72746966 69636174 65204D61
6E616765 72300D06 092A8648 86F70D01 01040500 03410015 BC7CECF9 696697DF
E887007F 7A8DA24F 1ED5A785 C5C60452 47860061 0C18093D 08958A77 5737246B
0A25550A 25910E27 8B8B428E 32F8D948 3DD1784F 954C70
quit
```

Configuring Certificate Autoenrollment with Key Rollover: Example

The following example shows how to configure the router to autoenroll with the CA named "trustme1" on startup. In this example, the **regenerate** keyword is issued, so a new key will be generated for the certificate. The renewal percentage is configured as 90 so if the certificate has a lifetime of one year, a new certificate is requested 36.5 days before the old certificate expires. The changes made to the running configuration are saved to the NVRAM startup configuration because autoenrollment will not update NVRAM if the running configuration has been modified but not written to NVRAM.

```
crypto pki trustpoint trustme1
 enrollment url http://trustme1.company.com/
 subject-name OU=Spiral Dept., O=tiedye.com
 ip-address ethernet0
 serial-number none
 auto-enroll 90 regenerate
 password revokeme
 rsakeypair trustme1 2048
 exit
crypto pki authenticate trustme1
copy system:running-config nvram:startup-config
```

Configuring Manual Certificate Enrollment with Key Rollover: Example

The following example shows how to configure key rollover to regenerate new keys with a manual certificate enrollment from the CA named "trustme2":

```
crypto pki trustpoint trustme2
 enrollment url http://trustme2.company.com/
 subject-name OU=Spiral Dept., O=tiedye.com
 ip-address ethernet0
 serial-number none
 regenerate
 password revokeme
 rsakeypair trustme2 2048s
 exit
crypto pki authenticate trustme2
crypto pki enroll trustme2
```

Creating and Verifying a Persistent Self-Signed Certificate: Example

The following example shows how to declare and enroll a trustpoint named "local" and generate a self-signed certificate with an IP address:

```
crypto pki trustpoint local
 enrollment selfsigned
 end
configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

crypto pki enroll local
Nov 29 20:51:13.067: %SSH-5-ENABLED: SSH 1.99 has been enabled
Nov 29 20:51:13.267: %CRYPTO-6-AUTOGEN: Generated new 512 bit key pair
% Include the router serial number in the subject name? [yes/no]: yes
% Include an IP address in the subject name? [no]: yes
Enter Interface name or IP Address[]: ethernet 0
Generate Self Signed Router Certificate? [yes/no]: yes
Router Self Signed Certificate successfully created
```

Note A router can have only one self-signed certificate. If you attempt to enroll a trustpoint configured for a self-signed certificate and one already exists, you receive a notification and are asked if you want to replace it. If so, a new self-signed certificate is generated to replace the existing one.

Enabling the HTTPS Server: Example

The following example shows how to enable the HTTPS server and generate a default trustpoint because one was not previously configured:

```
configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

ip http secure-server
% Generating 1024 bit RSA keys ...[OK]
*Dec 21 19:14:15.421:%PKI-4-NOAUTOSAVE:Configuration was modified.  Issue "write memory"
to save new certificate
Router(config)#
```

Note You need to save the configuration to NVRAM if you want to keep the self-signed certificate and have the HTTPS server enabled following router reloads.

The following message also appears:

```
*Dec 21 19:14:10.441:%SSH-5-ENABLED:SSH 1.99 has been enabled
Router(config)#
```

Note Creation of the key pair used with the self-signed certificate causes the Secure Shell (SSH) server to start. This behavior cannot be suppressed. You may want to modify your access control lists (ACLs) to permit or deny SSH access to the router.

Verifying the Self-Signed Certificate Configuration: Example

The following example displays information about the self-signed certificate that you just created:

```
Router# show crypto pki certificates

Router Self-Signed Certificate
  Status: Available
  Certificate Serial Number: 01
  Certificate Usage: General Purpose
  Issuer:
    cn=IOS-Self-Signed-Certificate-3326000105
  Subject:
    Name: IOS-Self-Signed-Certificate-3326000105
    cn=IOS-Self-Signed-Certificate-3326000105
  Validity Date:
    start date: 19:14:14 GMT Dec 21 2004
    end   date: 00:00:00 GMT Jan 1 2020
  Associated Trustpoints: TP-self-signed-3326000105
```

Note The number 3326000105 above is the router's serial number and varies depending on the router's actual serial number.

The following example displays information about the key pair corresponding to the self-signed certificate:

```
Router# show crypto key mypubkey rsa

% Key pair was generated at: 19:14:10 GMT Dec 21 2004
Key name: TP-self-signed-3326000105
 Usage: General Purpose Key
 Key is not exportable.
 Key Data:
  30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00B88F70
  6BC78B6D 67D6CFF3 135C1D91 8F360292 CA44A032 5AC1A8FD 095E4865 F8C95A2B
  BFD1C2B7 E64A3804 9BBD7326 207BD456 19BAB78B D075E78E 00D2560C B09289AE
  6DECB8B0 6672FB3A 5CDAEE92 9D4C4F71 F3BCB269 214F6293 4BA8FABF 9486BCFC
  2B941BCA 550999A7 2EFE12A5 6B7B669A 2D88AB77 39B38E0E AA23CB8C B7020301 0001
% Key pair was generated at: 19:14:13 GMT Dec 21 2004
Key name: TP-self-signed-3326000105.server
 Usage: Encryption Key
 Key is not exportable.
 Key Data:
  307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00C5680E 89777B42
  463E5783 FE96EA9E F446DC7B 70499AF3 EA266651 56EE29F4 5B003D93 2FC9F81D
  8A46E12F 3FBAC2F3 046ED9DD C5F27C20 1BBA6B9B 08F16E45 C34D6337 F863D605
  34E30F0E B4921BC5 DAC9EBBA 50C54AA0 BF551BDD 88453F50 61020301 0001
Router#
```

Note The second key pair with the name TP-self-signed-3326000105.server is the SSH key pair and is generated when any key pair is created on the router and SSH starts up.

The following example displays information about the trustpoint named “local”:

```
Router# show crypto pki trustpoints

Trustpoint local:
    Subject Name:
    serialNumber=C63EBBE9+ipaddress=10.3.0.18+hostname=test.cisco.com
          Serial Number: 01
    Persistent self-signed certificate trust point
```

Configuring Direct HTTP Enrollment: Example

The following example show how to configure an enrollment profile for direct HTTP enrollment with a CA server:

```
crypto pki trustpoint Entrust
 enrollment profile E
 serial

crypto pki profile enrollment E
 authentication url http://entrust:81
 authentication command GET /certs/cacert.der
 enrollment url http://entrust:81/cda-cgi/clientcgi.exe
 enrollment command POST reference_number=$P2&authcode=$P1
 &retrievedAs=rawDER&action=getServerCert&pkcs10Request=$REQ
 parameter 1 value aaaa-bbbb-cccc
 parameter 2 value 5001
```

Additional References

The following sections provide references related to certificate enrollment for a PKI.

Related Documents

Related Topic	Document Title
Overview of PKI, including RSA keys, certificate enrollment, and CAs.	"Cisco IOS PKI Overview: Understanding and Planning a PKI" module
Secure Device Provisioning: functionality overview and configuration tasks	"Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI" module
RSA key generation and deployment	"Deploying RSA Keys Within a PKI" module
Cisco IOS certificate server overview information and configuration tasks	"Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment" module
PKI commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*, Release 12.4

Technical AssistanceAssistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Feature Information for PKI Certificate Enrollment

Table 60 ists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for a specific command was introduced, see the command reference documentation.

For information on a feature in this technology that is not documented here, see the Implementing and Managing PKI Features Roadmap.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Table 60 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 60 Feature Information for PKI Certificate Enrollment

Feature Name	Software Release	Feature Configuration Information
Certificate Autoenrollment	12.2(8)T	This feature introduces certificate autoenrollment, which allows the router to automatically request a certificate from the CA that is using the parameters in the configuration. The following sections provide information about this feature: • Automatic Certificate Enrollment • Configuring Certificate Enrollment or Autoenrollment The following commands were introduced by this feature: **auto-enroll**, **rsakeypair**, **show crypto ca timers**
Certificate Enrollment Enhancements	12.2(8)T	This feature introduces five new **crypto ca trustpoint** subcommands that provide new options for certificate requests and allow users to specify fields in the configuration instead of having to go through prompts. The following section provides information about this feature: • Configuring Certificate Enrollment or Autoenrollment The following commands were introduced by this feature: **ip-address (ca-trustpoint)**, **password (ca-trustpoint)**, **serial-number**, **subject-name**, **usage**
Direct HTTP Enrollment with CA Servers	12.3(4)T	This feature allows users to configure an enrollment profile if their CA server does not support SCEP and they do not want to use an RA as a proxy. The enrollment profile allows users to send HTTP requests directly to the CA server instead of the RA proxy. The following sections provide information about this feature: • Certificate Enrollment Profiles • Configuring a Certificate Enrollment Profile for Enrollment or Reenrollment The following commands were introduced by this feature: **authentication command**, **authentication terminal**, **authentication url**, **crypto ca profile enrollment**, **enrollment command**, **enrollment profile**, **enrollment terminal**, **enrollment url**, **parameter**

Table 60 Feature Information for PKI Certificate Enrollment (continued)

Feature Name	Software Release	Feature Configuration Information
Import of RSA Key Pair and Certificates in PEM Format	12.3(4)T	This feature allows customers to issue certificate requests and receive issued certificates in PEM-formatted files. The following section provides information about this feature: • Configuring Manual Certificate Enrollment The following commands were modified by this feature: **enrollment**, **enrollment terminal**
Key Rollover for Certificate Renewal	12.3(7)T	This feature allows the certificate renewal request to be made before the certificate expires and retains the old key and certificate until the new certificate is available. The following sections provide information about this feature: • Certificate Autoenrollment with Key Rollover • Configuring Certificate Enrollment or Autoenrollment • Configuring Manual Certificate Enrollment The following commands were introduced or modified by this feature: **auto-enroll**, **regenerate**
Manual Certificate Enrollment (TFTP Cut-and-Paste)	12.2(13)T	This feature allows users to generate a certificate request and accept CA certificates as well as the router's certificates via a TFTP server or manual cut-and-paste operations. The following sections provide information about this feature: • Supported Certificate Enrollment Methods • Configuring Manual Certificate Enrollment The following commands were introduced or modified by this feature: **crypto ca import**, **enrollment**, **enrollment terminal**
Multiple-Tier CA Hierarchy[1]	12.2(15)T	This enhancement enables users to set up a PKI in a hierarchical framework to support multiple CAs. Within a hierarchical PKI, all enrolled peers can validate the certificate of one another as long as the peers share a trusted root CA certificate or a common subordinate CA. The following section provides information about this enhancement: • Hierarchical PKI: Multiple CAs

Table 60 Feature Information for PKI Certificate Enrollment (continued)

Feature Name	Software Release	Feature Configuration Information
Persistent Self-Signed Certificates	12.3(14)T	This feature allows users the HTTPS server to generate and save a self-signed certificate in the router's startup configuration. Thus, future SSL handshakes between the client and the HTTPS server can use the same self-signed certificate without user intervention. The following sections provide information about this feature: • Supported Certificate Enrollment Methods • Configuring a Persistent Self-Signed Certificate for Enrollment via SSL The following commands were introduced or modified by this feature: **enrollment selfsigned**, **show crypto pki certificates**, **show crypto pki trustpoints**
PKI Status[1]	12.3(11)T	This enhancement added the **status** keyword to the **show crypto pki trustpoints** command, which allows you to view the current status of the trustpoint. Prior to this enhancement, you had to issue the **show crypto pki certificates** and the **show crypto pki timers** commands for the current status. The following section provides information about this enhancement: • How to Configure Certificate Enrollment for a PKI
Reenroll Using Existing Certificates	12.3(11)T	This feature allows users to reenroll a router with a Cisco IOS CA via existing certificates from a third-party vendor CA. The following section provides information about this enhancement: • Configuring a Certificate Enrollment Profile for Enrollment or Reenrollment The following commands were introduced by this feature: **enrollment credential**, **grant auto trustpoint**
Trustpoint CLI	12.2(8)T	This feature introduces the **crypto pki trustpoint** command, which adds support for trustpoint CAs.

1. This is a minor enhancement. Minor enhancements are not typically listed in Feature Navigator.

Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI

This module describes how to use Secure Device Provisioning (SDP) in a public key infrastructure (PKI). SDP is a web-based certificate enrollment interface that can be used to easily deploy PKI between two end devices, such as a Cisco IOS client and a Cisco IOS certificate server. SDP provides a solution for users deploying a large number of peer devices, including certificates and configurations.

Module History

This module was first published on May 2, 2005, and last updated on May 2, 2005.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all features. To find information about feature support and configuration, use the "Feature Information for SDP in a PKI" section on page 1518.

Contents

- Prerequisites for Setting Up SDP for Enrollment in a PKI, page 1494
- Information About Setting Up SDP for Enrollment in a PKI, page 1494
- How to Set Up SDP for a PKI, page 1499
- Configuration Examples for Setting up a PKI via SDP, page 1509
- Additional References, page 1517
- Feature Information for SDP in a PKI, page 1518

Prerequisites for Setting Up SDP for Enrollment in a PKI

The following assumptions are made before setting up SDP:

- Both the client device and the server have IP connectivity between each other.
- The administrator has a web browser that supports JavaScript.
- The administrator has enable privileges on the client device.
- Your Cisco IOS software version is Release 12.3(8)T or later.

Information About Setting Up SDP for Enrollment in a PKI

Before using SDP for certificate enrollment, you should understand the following concepts:

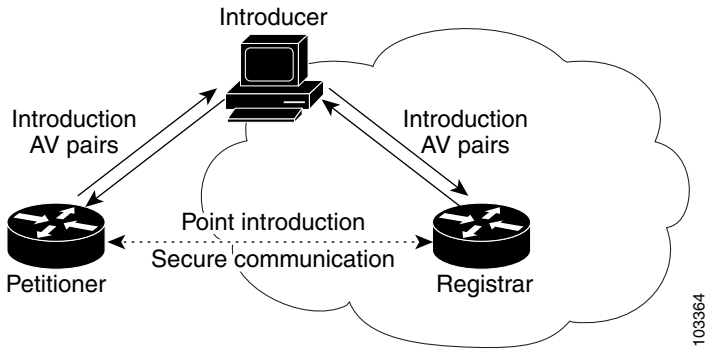
- SDP Overview, page 1494
- How SDP Works, page 1495
- How SDP Uses an External AAA Database, page 1498

SDP Overview

SDP (also referred to as Trusted Transitive Introduction [TTI]) is a communication protocol that provides a bidirectional introduction between two end entities, such as a new network device and a Virtual Private Network (VPN). SDP involves the following three entities (see Figure 100):

- Introducer—A mutually trusted device that introduces the petitioner to the registrar. The introducer can be a device user, such as a system administrator.
  - An introducer can be configured as an administrative introducer, which allows an administrator performing the introduction to supply the name for the device being introduced. The supplied device name is used as if it were the name of an introducer in the normal SDP mechanisms, preserving the existing functionality of the SDP configuration. For more information on function of the administrative introducer, see the section "Authentication and Authorization Lists for an Administrative Introducer."
- Petitioner—A new device that is joined to the secure domain.
- Registrar—A server that authorizes the petitioner. The registrar can be a certificate server.

Figure 100 Post-Introduction Secure Communication



SDP is implemented over a web browser with three phases—welcome, introduction, and completion. Each phase is shown to the user via a web page. For more information on each phase, see the section "How SDP Works."

How SDP Works

The following sections describe how SDP deploys PKI between two devices:

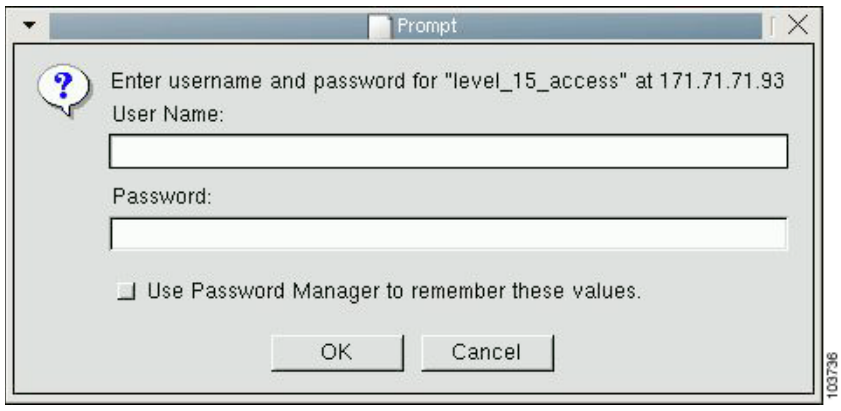
- SDP Phase One—Welcome
- SDP Phase Two—Introduction
- SDP Phase Three—Completion

The sample figures show how to introduce the local device "bubinga.cisco.com" (the petitioner) to the security domain of "walnut.cisco.com" (the registrar). (The "introducer" is referred to as the end user.)

SDP Phase One—Welcome

The welcome phase is the initial communication between the introducer and petitioner. Before the welcome page is displayed, the end user must direct his or her browser to the welcome page via the URL "http://device/ezsdd/welcome." Thereafter, the local login dialog box is displayed (see Figure 101), and the end user can log into the local device (for example, bubinga.cisco.com) via a local password. (If a username password is configured on the local device, the user must log into that device.)

Figure 101 Petitioner Local Login Dialog Box



After the password is successfully entered, the welcome web page is displayed (see Figure 102), which is initiated from the petitioner (for example, bubinga.cisco.com).

Figure 102 Sample SDP Welcome Page

After entering the URL of the registrar (for example, walnut.cisco.com) and clicking the Next button on the welcome web page, the end user logs into his or her registrar as shown in Figure 103.

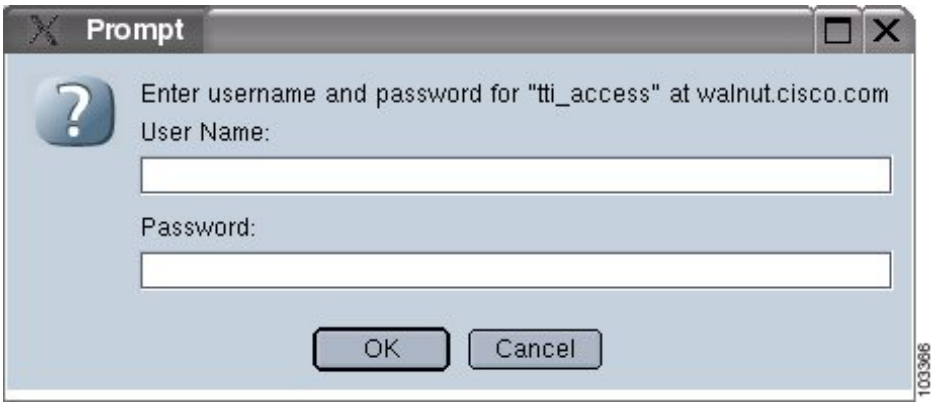
SDP Phase Two—Introduction

Before the introduction page is displayed, the end user must log into the registrar (for example, walnut.cisco.com) via a username and password per the external authentication, authorization, and accounting (AAA) database. With an external AAA database, the introducer can use an account on the database to perform the introduction without requiring knowledge of the enable password of the registrar. Without an external AAA database, the introducer may use the enable password of the registrar for authentication.

Note Using the enable password of the registrar exposes the password to end users; therefore, it is recommended that the enable password be used only for administrative testing.

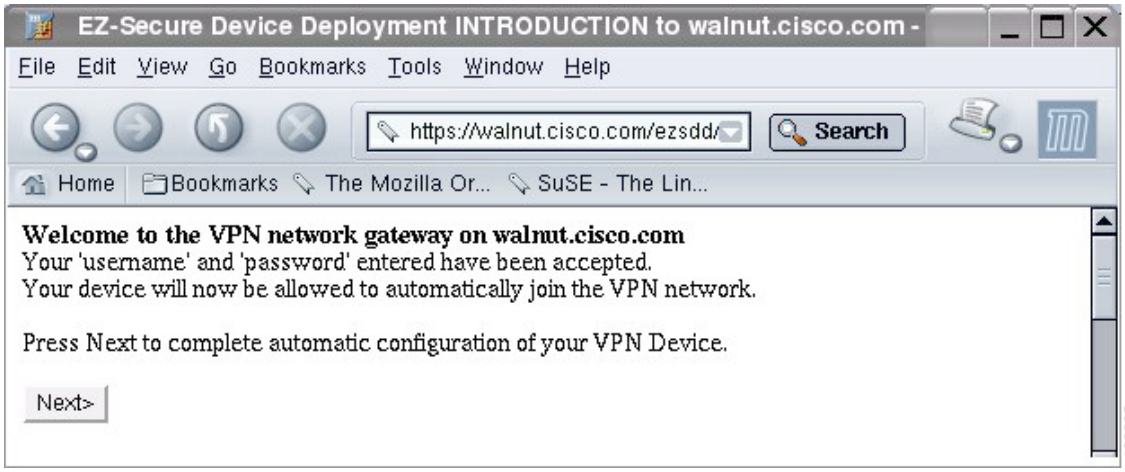
For more information on the function of the external AAA database, see the section "How SDP Uses an External AAA Database."

Figure 103 Registrar Remote Login Dialog Box



After the end user successfully enters his or her password, the introduction web page is displayed (see Figure 104).

Figure 104 Sample SDP Introduction Page

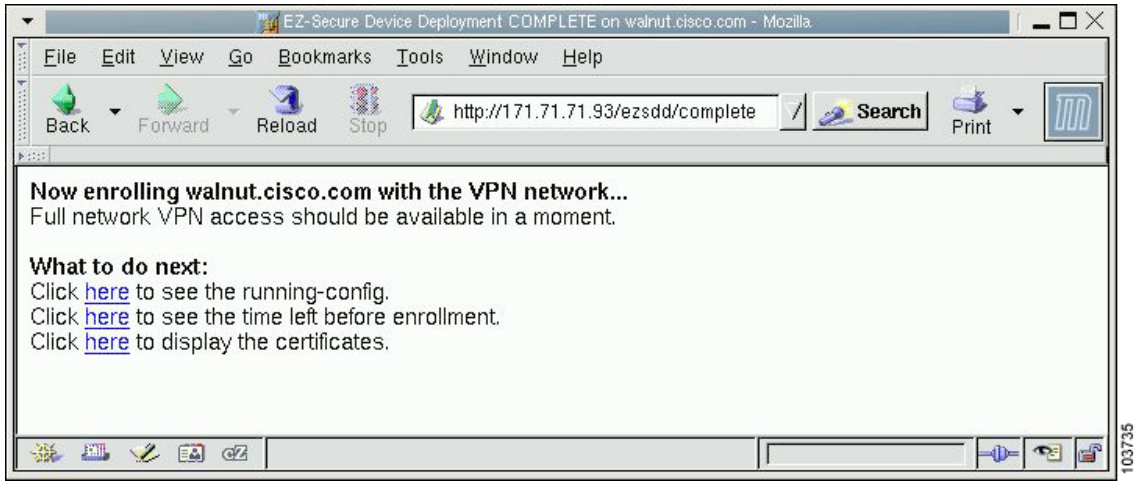


After the end user clicks the Next button on the introduction page, the end user enters the completion phase and automatically returns to his or her local device (for example, bubinga.cisco.com).

SDP Phase Three—Completion

Now that the end user has enrolled with the remote device (for example, walnut.cisco.com) and returned to his or her local device (for example, bubinga.cisco.com), the local device will display the completion page (see Figure 105).

Figure 105 Sample SDP Completion Page



After the SDP exchange is complete, the petitioner will automatically enroll with the registrar and obtain a certificate.

How SDP Uses an External AAA Database

The external AAA database is accessed twice during the SDP exchange. The first time is to authenticate the introducer; that is, when the registrar receives an introduction request via the secure HTTP (HTTPS) server, the registrar does a AAA lookup based on the introducer's username and password to authorize the request. The second time is to obtain authorization information that is applied to the configuration and certificates that are issued to the petitioner device; that is, the registrar checks the integrity of the request by verifying the request signature using the petitioner-signing certificate. The certificate subject name may be specified in the AAA database, and up to nine configuration template variables may be specified and expanded into the template configuration.

Use of a Self-Signed Certificate Versus a Certificate Issued by Another CA Server

By default, the SDP exchange results in only one certificate being issued to the petitioner device. Although just one certificate is issued, the introducer is not restricted from introducing multiple devices and thus obtaining multiple certificates. By specifying the subject name in the certificate that is issued, you can be assured that all certificates that are issued in this way are associated with the introducer. You can use PKI AAA integration to further restrict the use of these certificates. Additionally, the AAA database can be configured to accept only one authentication and authorization request per user.

Because the petitioner certificate is self-signed, it is just used to convey the public key of the petitioner. No verification or authorization check is performed on the certificate; thus, authorization is per-user based and no per-device information is used.

There are some scenarios when per-device authorization is preferred. Therefore, if the petitioner is able to use certificates issued by other certification authority (CA) servers for SDP transactions, the existing PKI can be used and authorization can be achieved over the certificate attributes.

Configuring the petitioner and the registrar for certificate-based authorization provides authorization of the specific device being deployed. Previously, introducer-to-petitioner device communication was secured only using physical security between the introducer and the petitioner device. SDP certificate-based authorization gives the registrar an opportunity to validate the current device identity before accepting the introduction.

Authentication and Authorization Lists for SDP

When you are configuring your SDP registrar, if you specify an authentication list and an authorization list, the registrar uses the specified lists for all introducer requests. The authentication list is used when authenticating the introducer (the AAA server checks for a valid account by looking at the username and password). The authorization list is used to receive the appropriate authorized fields for the certificate subject name and a list of template variables to be expanded into the Cisco IOS command-line interface (CLI) snippet that is sent back to the petitioner. The authentication and authorization lists will usually point to the same AAA server list, but it is possible to use a different database for authentication and authorization. (Storing files on different databases is not recommended.)

When a petitioner makes an introduction request, multiple queries are sent to the AAA list database on the RADIUS or TACACS+ server. The queries search for entries of the following form:

```
user Password <userpassword>
   cisco-avpair="ttti:subjectname=<<DN subjectname>>"
   cisco-avpair="tti:iosconfig#<<value>>"
   cisco-avpair="tti:iosconfig#<<value>>"
   cisco-avpair="tti:iosconfig#=<<value>>"
```

Note The existence of a valid AAA username record is enough to pass the authentication check. The "cisco-avpair=tti" information is necessary only for the authorization check.

If a subject name was received in the authorization response, the SDP registrar stores it in the enrollment database, and that "subjectname" overrides the subject name that is supplied in the subsequent certificate request (PKCS10) from the petitioner device.

The numbered "tti:iosconfig" values are expanded into the SDP Cisco IOS snippet that is sent to the petitioner. The configurations replace any numbered ($1 through $9) template variable. Because the default Cisco IOS snippet template does not include the variables $1 through $9, these variables are ignored unless you configure an external Cisco IOS snippet template. To specify an external configuration, use the **template config** command.

Note The template configuration location may include a variable "$n," which is expanded to the name with which the user is logged in.

Authentication and Authorization Lists for an Administrative Introducer

The SDP mechanisms assume a permanent relationship between the introducer and the device. As a result, the introducer username is used to define the device name.

In some SDP deployment scenarios, the introducer is an administrator doing the introduction for many devices. However, using the introducer (the administrator) name to define the device name results in multiple devices being incorrectly deployed with the same device name. Instead, an administrative introducer allows the administrator to specify the correct device name during the introduction.

More generally stated, the introducer username is used as the database record locator to determine all other information about the device including the Cisco IOS configuration template, various template variables (pulled from an AAA database and expanded into the template), and the appropriate subject name for PKI certificates issued to the device. For simplicity, this database record locator is called the user/device name.

The administrative introducer provides a device name. In that way, an administrator can provide the appropriate record locator when doing an introduction. For example, if an administrator is trying to introduce a device for username "rover," the administrator introduces the device into the PKI network and provides rover as the record locator after logging into the registrar using the administrator's own credentials. The record locator, rover, becomes the device name. All other template and PKI certificate subject name information specific to the introduction is then provided by the rover username records instead of by the administrator's record.

The registrar device uses the supplied username information with a user introducer name. The username allows the existing mechanisms for determining a user's authorization, template, and PKI certificate information to be supported without modification.

How to Set Up SDP for a PKI

This section contains the following procedures that should be followed when setting up SDP for your PKI. Note that you can configure the registrar according to only one of the registrar configuration tasks.

- Enabling the SDP Petitioner, page 1500
- Enabling the SDP Registrar and Adding AAA Lists to the Server, page 1501
- Enabling the SDP Registrar for Certificate-Based Authorization, page 1504
- Configuring an Administrative Introducer, page 1506

Enabling the SDP Petitioner

Perform this task to enable or disable the petitioner and associate a trustpoint with the SDP exchange.

You can also use this task to configure the petitioner to use a certificate and the Rivest, Shamir, and Adelman (RSA) keys associated with a specific trustpoint.

Note The petitioner is enabled by default on a Cisco device that contains a crypto image; thus, you have only to issue the **crypto provisioning petitioner** command if you have previously disabled the petitioner or if you want to use an existing trustpoint instead of the automatically generated trustpoint.

Note By default, the SDP petitioner device uses an existing certificate. If multiple certificates and one specific certificate exist, use this task to make a choice. However, this task is not necessary to enable the default behavior.

Prerequisites

- The HTTP server must be enabled via the **ip http server** command. (The HTTP server is typically enabled by default on many default Cisco IOS configurations.)
- If you are configuring the petitioner to use a certificate and RSA keys, your SDP petitioner device must have an existing manufacturer's or a third-party certificate.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto provisioning petitioner**
4. **trustpoint** *trustpoint-label*

   or

   trustpoint signing *trustpoint-label*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto provisioning petitioner** **Example:** `Router(config)# crypto provisioning petitioner`	Allows SDP petitioner device behavior to be modified and enters tti-petitioner configuration mode. **Note** Effective with Cisco IOS Release 12.3(14)T, the **crypto provisioning petitioner** command replaced the **crypto wui tti petitioner** command.
Step 4	**trustpoint** *trustpoint-label* **Example:** `Router(tti-petitioner) trustpoint mytrust` or **trustpoint signing** *trustpoint-label* **Example:** `Router(tti-petitioner) trustpoint signing mytrust`	(Optional) Specifies the trustpoint that is to be associated with the SDP exchange between the petitioner and the registrar. **Note** If this command is not issued, the *trustpoint-label* argument is automatically labeled "tti." (Optional) Specifies the trustpoint and associated certificate that are used when signing all introduction data during the SDP exchange.
Step 5	**end** **Example:** `Router(tti-petitioner) end`	(Optional) Exits tti-petitioner configuration mode.

Troubleshooting Tips

After the SDP exchange is complete, a new trustpoint-label named "tti" will exist. The trustpoint will be automatically enrolled with the certificate server (the registrar). To verify that the trustpoint is really there, use the **show running-config** command.

What to Do Next

If you set up the petitioner to use a certificate and the RSA keys associated with the specified trustpoint, you should configure the registrar as shown in the task "Enabling the SDP Registrar for Certificate-Based Authorization."

Enabling the SDP Registrar and Adding AAA Lists to the Server

Perform this task to enable the registrar and associate a certificate server with the SDP exchange.

You can also use this task if you want to add an authentication list and an authorization list to the RADIUS or TACACS+ server.

Prerequisites

Before configuring an registrar, ensure the following tasks are complete:

- Enable the HTTP server or the HTTPS server.

Note It is recommended that you issue the **ip http secure-server** command to enable the HTTPS web server. If you enable a secure server, you should issue the **ip http secure-trustpoint** command. You must disable the standard HTTP server via the **no ip http server** command (if the standard server is enabled). The specified trustpoint is a registrar local trustpoint appropriate for HTTPS communication between the registrar and the user's browser.

- Configure the Cisco IOS certificate server (via the **crypto pki server** command).

If you are configuring AAA lists, you should complete the prerequisites required for the registrar in addition to completing the following tasks:

- Add user information to the AAA server database. To configure a RADIUS or TACACS+ AAA server, see the "Configuring RADIUS" and "Configuring TACACS+" chapters of the *Cisco IOS Security Configuration Guide*.
- Configure new AAA lists. To configure AAA lists, see the following chapters in the *Cisco IOS Security Configuration Guide*: "Configuring RADIUS," "Configuring TACACS+," "Configuring Authentication," and "Configuring Authorization."

Restrictions

Cisco IOS CA Device Requirement

During the SDP process, a Cisco IOS CA certificate is automatically issued to the peer device. If an SDP registrar is configured on a third-party vendor's CA device, the SDP process will not work.

The template config Command

There are only nine Cisco IOS configuration variables. However, if you need more configuration flexibility, the **template config** command can be used to reference a configuration template that is specific to the introducer.

The **template config** command can also be used to reference a configuration Common Gateway Interface (CGI) script.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto provisioning registrar**
4. **pki-server** *label*
5. **authentication list** *list-name*
6. **authorization list** *list-name*
7. **template username** *name* [**password** *password*]
8. **template config** *url*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto provisioning registrar** **Example:** `Router(config)# crypto provisioning registrar`	Configures a device to become a registrar for the SDP exchange and enters tti-registrar configuration mode. **Note** Effective with Cisco IOS Release 12.3(14)T, the **crypto provisioning registrar** command replaced the **crypto wui tti registrar** command.
Step 4	**pki-server** *label* **Example:** `Router(tti-registrar)# pki-server mycs`	Specifies the certificate server that is to be associated with the SDP exchange between the petitioner and the registrar.
Step 5	**authentication list** *list-name* **Example:** `Router (tti-registrar)# authentication list authen-tac`	(Optional) Authenticates the introducer in an SDP exchange.
Step 6	**authorization list** *list-name* **Example:** `Router (tti-registrar)# authorization list author-rad`	(Optional) Receives the appropriate authorized fields for the certificate subject name and list of template variables to be expanded into the Cisco IOS CLI snippet that is sent back to the petitioner.
Step 7	**template username** *name* [**password** *password*] **Example:** `Router(tti-registrar)# template username ftpuser password ftppwd`	(Optional) Establishes a username and password in which to access the configuration template on the file system.
Step 8	**template config** *url* **Example:** `Router(tti-registrar)# template config ftp://server/iossnippet.txt`	(Optional) Specifies a remote URL for the Cisco IOS CLI configuration template, which is sent from the registrar to the petitioner during the SDP exchange.

Examples

To help troubleshoot the SDP transaction, you can issue the **debug crypto provisioning** command, which displays output from the petitioner and registrar devices.

The following is output for the **debug crypto provisioning** command. The output from the petitioner and registrar devices are shown below.

```
Petitioner device
! The user starts the Welcome phase.
Nov  7 03:15:48.171: CRYPTO_PROVISIONING: received welcome get request.
! The router generates a Rivest, Shamir, and Adelman (RSA) keypair for future enrollment.
Nov  7 03:15:48.279: CRYPTO_PROVISIONING: keyhash 'A506BE3B83C6F4B4A6EFCEB3D584AACA'
! The TTI transaction is completed.
Nov  7 03:16:10.607: CRYPTO_PROVISIONING: received completion post request.

Registrar device
!. During the introduction phase, the browser prompts for login information.
06:39:18: CRYPTO_PROVISIONING: received introduction post request.
06:39:18: CRYPTO_PROVISIONING: checking AAA authentication (ipsecca_script_aaalist,
ttiuser)
! This happens if the user types in the wrong username or password.
06:39:19: CRYPTO_PROVISIONING: authentication declined by AAA, or AAA server not found -
0x3
06:39:19: CRYPTO_PROVISIONING: aaa query fails!
! The user re-enters login information.
06:39:19: CRYPTO_PROVISIONING: received introduction post request.
06:39:19: CRYPTO_PROVISIONING: checking AAA authentication (ipsecca_script_aaalist,
ttiuser)
06:39:20: CRYPTO_PROVISIONING: checking AAA authorization (ipsecca_script_aaalist,
ttiuser)
! The login attempt succeeds and authorization information is retrieved from the AAA
database.
06:39:21: CRYPTO_PROVISIONING: aaa query ok!
! These attributes are inserted into the configuration template.
06:39:21: CRYPTO_PROVISIONING: building TTI av pairs from AAA attributes
06:39:21: CRYPTO_PROVISIONING: "subjectname" = "CN=jack, O=cisco, C=US"
06:39:21: CRYPTO_PROVISIONING: "$1" = "ntp server 10.3.0.1"
06:39:21: CRYPTO_PROVISIONING: "$2" = "hostname jack-vpn"
! The registrar stores this subject name and overrides the subject name in the subsequent
enrollment request.
06:39:21: CRYPTO_PROVISIONING: subjectname=CN=jack, O=cisco, C=US
! The registrar stores this key information so that it may be used to automatically grant
the subsequent enrollment request.
06:39:21: CRYPTO_PROVISIONING: key_hash=A506BE3B83C6F4B4A6EFCEB3D584AACA
```

Enabling the SDP Registrar for Certificate-Based Authorization

Perform this task to enable the SDP registrar to perform the following functions:

- Verify the petitioner-signing certificate using a specified trustpoint or any configured trustpoint.
- Initiate authorization lookups using the introducer username and the certificate name field.

Prerequisites

You must also configure the SDP petitioner to use a certificate and RSA keys associated with a specific trustpoint. To complete this task, use the trustpoint signing command as shown in the task "Enabling the SDP Petitioner."

Restrictions

Because RADIUS does not differentiate between authentication and authorization, you need to use the default password, cisco, for certificate authorization.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto provisioning registrar**
4. **authentication trustpoint** {*trustpoint-label* | **use-any**}
5. **authorization** {**login | certificate | login certificate**}
6. **authorization username** {**subjectname** *subjectname*}
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto provisioning registrar** **Example:** `Router(config)# crypto provisioning registrar`	Configures a device to become an SDP registrar and enters tti-registrar configuration mode.
Step 4	**authentication trustpoint** {*trustpoint-label*| **use-any**} **Example:** `Router(tti-registrar)# authentication trustpoint mytrust`	(Optional) Specifies the trustpoint used to authenticate the SDP petitioner device's existing certificate. • *trustpoint-label*—Specifies a specific trustpoint. • **use-any**—Specifies any configured trustpoint. **Note** If you do not use this command to specify a trustpoint, the existing petitioner certificate is not validated. (This functionality provides compatibility with self-signed petitioner certificates.)

	Command or Action	Purpose
Step 5	**authorization** {**login** | **certificate** | **login certificate**} **Example:** `Router(tti-registrar)# authorization login certificate`	(Optional) Enables AAA authorization for an introducer or a certificate. • Use the **login** keyword for authorization based on the introducer's username. • Use the **certificate** keyword for authorization based on the petitioner's certificate. • Use the **login certificate** keyword for authorization based on the introducer's username and the petitioner's certificate.
Step 6	**authorization username** {**subjectname** *subjectname*} **Example:** `Router(tti-registrar)# authorization username subjectname all`	Sets parameters for the different certificate fields that are used to build the AAA username. • The **all** keyword specifies that the entire subject name if the certificate is used as the authorization username.
Step 7	**end** **Example:** `Router(tti-registrar)# end`	(Optional) Exits tti-registrar configuration mode.

Configuring an Administrative Introducer

Perform the following task to configure an administrative introducer using administrator authentication and authorization lists.

Prerequisites

The administrative introducer must have enable privileges on the client device and administrator privileges on the server.

Restrictions

When using RADIUS, a user/device that needs to be introduced by the administrative introducer must always use cisco as its own password. TACACS+ does not have this limitation; a user/device can have any password and be introduced by the administrative introducer.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto provisioning registrar**
4. **administrator authentication list** *list-name*
5. **administrator authorization list** *list-name*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto provisioning registrar** **Example:** `Router(config)# crypto provisioning registrar`	Configures a device to become an SDP registrar and enters tti-registrar configuration mode.
Step 4	**administrator authentication list** *list-name* **Example:** `Router(tti-registrar)# administrator authentication list authen-tac`	Configures the AAA list used to authenticate an administrator during an introduction.
Step 5	**administrator authorization list** *list-name* **Example:** `Router(tti-registrar)# administrator authorization list author-tac`	Configures the AAA list used to obtain authorization information for an administrator during an introduction. Information that can be obtained includes the certificate subject name and/or the list of template variables to be expanded into the Cisco IOS CLI snippet that is sent back to the petitioner.
Step 6	**end** **Example:** `Router(tti-registrar)# end`	(Optional) Exits tti-registrar configuration mode.

Examples

The following example from the **show running-config** command allows you to verify that an administrative introducer using administrator authentication and authorization lists have been created:

```
Router# show running-config

Building configuration...

Current configuration : 2700 bytes
!
! Last configuration change at 01:22:26 GMT Fri Feb 4 2005
!
version 12.4
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname router
!
```

```
boot-start-marker
boot-end-marker
!
memory-size iomem 5
enable secret 5 $1$tpBS$PXnBDTIDXfX5pWa//1JX20
enable password lab
!
aaa new-model
!
!
!
aaa session-id common
!
resource manager
!
clock timezone GMT 0
ip subnet-zero
no ip routing
!
!
no ip dhcp use vrf connected
!
!
no ip cef
no ip domain lookup
ip domain name cisco.com
ip host router 10.3.0.6
ip host router.cisco.com 10.3.0.6
no ip ips deny-action ips-interface
!
no ftp-server write-enable
!
crypto pki server mycs
!
crypto pki trustpoint mycs
 revocation-check crl
 rsakeypair mycs
!
crypto pki trustpoint tti
 revocation-check crl
 rsakeypair tti
!
crypto pki trustpoint mic
 enrollment url http://router:80
 revocation-check crl
!
crypto pki trustpoint cat
 revocation-check crl
!
!
!
crypto pki certificate map cat 10
!
crypto pki certificate chain mycs
 certificate ca 01
crypto pki certificate chain tti
crypto pki certificate chain mic
 certificate 02
 certificate ca 01
crypto pki certificate chain cat
!
crypto provisioning registrar <---------- !SDP registrar device parameters!
 administrator authentication list authen-tac
 administrator authorization list author-tac
```

```
!
no crypto engine onboard 0
username qa privilege 15 password 0 lab
```

Configuration Examples for Setting up a PKI via SDP

This section contains the following configuration examples:

- Verifying the SDP Registrar: Example, page 1509
- Verifying the SDP Petitioner: Example, page 1512
- Adding AAA Lists to a RADIUS or TACACS+ Server: Examples, page 1515
- Configuration Template File: Example, page 1516
- Configuring the Petitioner and Registrar for Certificate-Based Authentication: Example, page 1516
- Configuring an Administrative Introducer Using Authentication and Authorization Lists: Example, page 1517

Verifying the SDP Registrar: Example

The following sample output from the **show running-config** command verifies that the certificate server "cs1" was configured and associated with the SDP exchange between the registrar and petitioner:

```
Router# show running-config

Building configuration...

Current configuration : 5902 bytes
!
! Last configuration change at 09:34:44 GMT Sat Jan 31 2004
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname pki-36a
!
boot-start-marker
boot-end-marker
!
logging buffered 32768 debugging
no logging console
enable secret 5 $1$b3jz$CKquLGjFIE3AdXA2/Rl9./
enable password lab
!
clock timezone GMT 0
no aaa new-model
ip subnet-zero
!
!
ip cef
ip domain name cisco.com
ip host msca-root
ip host yni-u10
ip host pki-36a 10.23.2.131
ip host pki-36a.cisco.com 10.23.2.131
```

```
!
!
crypto pki server cs1
 issuer-name CN = ioscs,L = Santa Cruz,C =US
 lifetime crl 336
 lifetime certificate 730
!
crypto pki trustpoint pki-36a
 enrollment url http://pki-36a:80
 ip-address FastEthernet0/0
 revocation-check none
!
crypto pki trustpoint cs1
 revocation-check crl
 rsakeypair cs1
!
!
crypto pki certificate chain pki-36a
 certificate 03
  308201D0 30820139 A0030201 02020103 300D0609 2A864886 F70D0101 04050030
  34310B30 09060355 04061302 55533114 30120603 55040713 0B205361 6E746120
  4372757A 310F300D 06035504 03130620 696F7363 73301E17 0D303430 31333130
  39333334 345A170D 30363031 33303039 33333434 5A303A31 38301606 092A8648
  86F70D01 09081309 31302E32 332E322E 32301E06 092A8648 86F70D01 09021611
  706B692D 3336612E 63697363 6F2E636F 6D305C30 0D06092A 864886F7 0D010101
  0500034B 00304802 4100AFFA 8F429618 112FAB9D 01F3352E 59DD3D2D AE67E31D
  370AC4DA 619735DF 9CF4EA13 64E4B563 C239C5F0 1578B773 07BED641 A18CA629
  191884B5 61B66ECF 4D110203 010001A3 30302E30 0B060355 1D0F0404 030205A0
  301F0603 551D2304 18301680 141DA8B1 71652961 3F7D69F0 02903AC3 2BADB137
  C6300D06 092A8648 86F70D01 01040500 03818100 67BAE186 327CED31 D642CB39
  AD585731 95868683 B950DF14 3BCB155A 2B63CFAD B34B579C 79128AD9 296922E9
  4DEDFCAF A7B5A412 AB1FC081 09951CE3 08BFFDD9 9FB1B9DA E9AA42C8 D1049268
  C524E58F 11C6BA7F C750320C 03DFB6D4 CBB3E739 C8C76359 CE939A97 B51B3F7F
  3FF;A9D82 9CFDB6CF E2503A14 36D0A236 A1CCFEAE
  quit
 certificate ca 01
  30820241 308201AA A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  34310B30 09060355 04061302 55533114 30120603 55040713 0B205361 6E746120
  4372757A 310F300D 06035504 03130620 696F7363 73301E17 0D303430 31333130
  39333132 315A170D 30373031 33303039 33313231 5A303431 0B300906 03550406
  13025553 31143012 06035504 07130B20 53616E74 61204372 757A310F 300D0603
  55040313 0620696F 73637330 819F300D 06092A86 4886F70D 01010105 0003818D
  00308189 02818100 FC0695AF 181CE90A 1B34B348 BA957178 680C8B51 07802AC3
  BF77B9C6 CB45092E 3C22292D C7D5FFC1 899185A1 FD8F37D5 C44FC206 6D1FA581
  E2264C83 1CC7453E 548C89C6 F3CD25BC 9BFFE7C5 E6653A06 62133950 78BED51B
  49128428 AB237F80 83A530EA 6F896193 F2134B54 D181F059 348AA84B 21EE6D80
  727BF668 EB004341 02030100 01A36330 61300F06 03551D13 0101FF04 05300301
  01FF300E 0603551D 0F0101FF 04040302 0186301D 0603551D 0E041604 141DA8B1
  71652961 3F7D69F0 02903AC3 2BADB137 C6301F06 03551D23 04183016 80141DA8
  B1716529 613F7D69 F002903A C32BADB1 37C6300D 06092A86 4886F70D 01010405
  00038181 00885895 A0141169 3D754EB2 E6FEC293 5BF0A80B E424AA2F A3F59765
  3463AAD1 55E71F0F B5D1A35B 9EA79DAC DDB40721 1344C01E 015BAB73 1E148E03
  9DD01431 A5E2887B 4AEC8EF4 48ACDB66 A6F9401E 8F7CA588 8A4199BB F8A437A0
  F25064E7 112805D3 074A154F 650D09B9 8FA19347 ED359EAD 4181D9ED 0C667C10
  8A7BCFB0 FB
  quit
crypto pki certificate chain cs1
 certificate ca 01
  30820241 308201AA A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  34310B30 09060355 04061302 55533114 30120603 55040713 0B205361 6E746120
  4372757A 310F300D 06035504 03130620 696F7363 73301E17 0D303430 31333130
  39333132 315A170D 30373031 33303039 33313231 5A303431 0B300906 03550406
  13025553 31143012 06035504 07130B20 53616E74 61204372 757A310F 300D0603
  55040313 0620696F 73637330 819F300D 06092A86 4886F70D 01010105 0003818D
```

```
  00308189 02818100 FC0695AF 181CE90A 1B34B348 BA957178 680C8B51 07802AC3
  BF77B9C6 CB45092E 3C22292D C7D5FFC1 899185A1 FD8F37D5 C44FC206 6D1FA581
  E2264C83 1CC7453E 548C89C6 F3CD25BC 9BFFE7C5 E6653A06 62133950 78BED51B
  49128428 AB237F80 83A530EA 6F896193 F2134B54 D181F059 348AA84B 21EE6D80
  727BF668 EB004341 02030100 01A36330 61300F06 03551D13 0101FF04 05300301
  01FF300E 0603551D 0F0101FF 04040302 0186301D 0603551D 0E041604 141DA8B1
  71652961 3F7D69F0 02903AC3 2BADB137 C6301F06 03551D23 04183016 80141DA8
  B1716529 613F7D69 F002903A C32BADB1 37C6300D 06092A86 4886F70D 01010405
  00038181 00885895 A0141169 3D754EB2 E6FEC293 5BF0A80B E424AA2F A3F59765
  3463AAD1 55E71F0F B5D1A35B 9EA79DAC DDB40721 1344C01E 015BAB73 1E148E03
  9DD01431 A5E2887B 4AEC8EF4 48ACDB66 A6F9401E 8F7CA588 8A4199BB F8A437A02;
  F25064E7 112805D3 074A154F 650D09B9 8FA19347 ED359EAD 4181D9ED 0C667C10
  8A7BCFB0 FB
  quit
!
crypto provisioning registrar
 pki-server cs1
!
!
!
crypto isakmp policy 1
 hash md5
!
!
crypto ipsec transform-set test_transformset esp-3des
!
crypto map test_cryptomap 10 ipsec-isakmp
 set peer 10.23.1.10
 set security-association lifetime seconds 1800
 set transform-set test_transformset
 match address 170
!
!
interface Loopback0
 ip address 10.23.2.131 255.255.255.255
 no ip route-cache cef
 no ip route-cache
 no ip mroute-cache
!
interface FastEthernet0/0
 ip address 10.23.2.2 255.255.255.192
 no ip route-cache cef
 no ip route-cache
 no ip mroute-cache
 duplex auto
 speed auto
 crypto map test_cryptomap
!
interface FastEthernet1/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
ip default-gateway 10.23.2.62
ip http server
no ip http secure-server
ip classless
ip route 0.0.0.0 0.0.0.0 10.23.2.62
!
!
access-list 170 permit ip host 10.23.2.2 host 10.23.1.10
dialer-list 1 protocol ip permit
!
```

```
!
control-plane
!
!
line con 0
 exec-timeout 0 0
 speed 115200
line aux 0
line vty 0 4
 password lab
 login
!
!
end
```

Verifying the SDP Petitioner: Example

After the SDP exchange is complete, the petitioner will automatically enroll with the registrar and obtain a certificate. The following sample output via the **show running-config** command shows the automatically generated configuration, which verifies that the trustpoint is really there:

```
Router# show running-config

Building configuration...

Current configuration : 4650 bytes
!
! Last configuration change at 09:34:53 GMT Sat Jan 31 2004
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname pki-36b
!
boot-start-marker
boot-end-marker
!
logging buffered 32768 debugging
no logging console
enable secret 5 $1$JYgw$060JKXgl6dERLZpU9J3gb.
enable password lab
!
clock timezone GMT 0
no aaa new-model
ip subnet-zero
!
!
ip cef
ip domain name cisco.com
ip host msca-root
ip host yni-u10
ip host pki-36a 10.23.2.131
ip host pki-36a.cisco.com 10.23.2.131
!
!
crypto pki trustpoint tti
 enrollment url http://pki-36a.cisco.com:80
 revocation-check crl
 rsakeypair tti 1024
```

```
 auto-enroll 70
!
!
crypto pki certificate chain tti
 certificate 02
  308201FC 30820165 A00302012;02020102 300D0609 2A864886 F70D0101 04050030
  34310B30 09060355 04061302 55533114 30120603 55040713 0B205361 6E746120
  4372757A 310F300D 06035504 03130620 696F7363 73301E17 0D303430 31333130
  39333333 385A170D 30363031 33303039 33333338 5A302231 20301E06 092A8648
  86F70D01 09021611 706B692D 3336622E 63697363 6F2E636F 6D30819F 300D0609
  2A864886 F70D0101 01050003 818D0030 81890281 8100E383 35584B6C 24751E2C
  F4088F06 C00BFECE 84CFF8EB 50D52044 03D14A2B 91E5A260 7D07ED24 DB599D27
  432065D9 0E459248 D7CDC15D 654E2AF6 BA27D79C 23850306 3E96C508 F311D333
  76FDDC9C A810F75C FCD10F1B 9A142F0C 338B6DB3 346D3F24 97A4B15D 0A9504E7
  1F6CB769 85E9F52B FE907AAF 63D54D66 1A715A20 D7DB0203 010001A3 30302E30
  0B060355 1D0F0404 03&#048;205A0 301F0603 551D2304 18301680 141DA8B1 71652961
  3F7D69F0 02903AC3 2BADB137 C6300D06 092A8648 86F70D01 01040500 03818100
  C5E2DA0E 4312BCF8 0396014F E18B3EE9 6C970BB7 B8FAFC61 EF849568 D546F73F
  67D2A73C 156202DC 7404A394 D6124DAF 6BACB8CF 96C3141D 109C5B0E 46F4F827
  022474ED 8B59D654 F04E31A2 C9AA1152 75A0C455 FD7EEEF5 A505A648 863EE9E6
  C361D9BD E12BBB36 16B729DF 823AD5CC 404CCE48 A4379CDC 67FF6362 0601B950
  quit
 certificate ca 01
  30820241 308201AA A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  34310B30 09060355 04061302 55533114 30120603 55040713 0B205361 6E746120
  4372757A 310F300D 06035504 03130620 696F7363 73301E17 0D303430 31333130
  39333132 315A170D 30373031 33303039 33313231 5A303431 0B300906 03550406
  13025553 31143012 06035504 07130B20 53616E74 61204372 757A310F 300D0603
  55040313 0620696F 73637330 819F300D 06092A86 4886F70D 01010105 0003818D
  00308189 02818100 FC0695AF 181CE90A 1B34B348 BA957178 680C8B51 07802AC3
  BF77B9C6 CB45092E 3C22292D C7D5FFC1 899185A1 FD8F37D5 C44FC206 6D1FA581
  E2264C83 1CC7453E 548C89C6 F3CD25BC 9BFFE7C5 E6653A06 62133950 78BED51B
  49128428 AB237F80 83A530EA 6F896193 F2134B54 D181F059 348AA84B 21EE6D80
  727BF668 EB004341 02030100 01A36330 61300F06 03551D13 0101FF04 05300301
  01FF300E 0603551D 0F0101FF 04040302 0186301D 0603551D 0E041604 141DA8B1
  71652961 3F7D69F0 02903AC3 2BADB137 C6301F06 03551D23 04183016 80141DA8
  B1716529 613F7D69 F002903A C32BADB1 37C6300D 06092A86 4886F70D 01010405
  00038181 00885895 A0141169 3D754EB2 E6FEC293 5BF0A80B E424AA2F A3F59765
  3463AAD1 55E71F0F B5D1A35B 9EA79DAC DDB40721 1344C01E 015BAB73 1E148E03
  9DD01431 A5E2887B 4AEC8EF4 48ACDB66 A6F9401E 8F7CA588 8A4199BB F8A437A0
  F25064E7 112805D3 074A154F 650D09B9 8FA19347 ED359EAD 4181D9ED 0C667C10
  8A7BCFB0 FB
  quit
!
no crypto engine accelerator
!
!
crypto isakmp policy 1
 hash md5
!
!
crypto ipsec transform-set test_transformset esp-3des
!
crypto map test_cryptomap 10 ipsec-isakmp
 set peer 10.23.2.2
 set security-association lifetime seconds 1800
 set transform-set test_transformset
 match address 170
!
!
interface Ethernet0/0
 ip address 10.23.1.10 255.255.255.192
 no ip route-cache cef
 no ip route-cache
```

```
 no ip mroute-cache
 half-duplex
 crypto map test_cryptomap
!
interface Ethernet0/1
 no ip address
 shutdown
 half-duplex
!
interface Ethernet0/2
 no ip address
 shutdown
 half-duplex
!
interface Ethernet0/3
 no ip address
 shutdown
 half-duplex
!
interface Serial1/0
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
ip default-gateway 10.23.1.62
ip http server
no ip http secure-server
ip classless
ip route 0.0.0.0 0.0.0.0 10.23.1.62
!
!
access-list 170 permit ip host 10.23.1.10 host 10.23.2.2
dialer-list 1 protocol ip permit
!
!
control-plane
!
!
line con 0
 exec-timeout 0 0
 speed 115200
line aux 0
line vty 0 4
 password lab
 login
!
!
end
```

Adding AAA Lists to a RADIUS or TACACS+ Server: Examples

This section contains the following configuration examples:

- TACACS+ AAA Server Database: Example, page 1515
- RADIUS AAA Server Database: Example, page 1515
- AAA List on a TACACS+ and a RADIUS AAA Server: Example, page 1515

TACACS+ AAA Server Database: Example

In the following example, user information has been added to a TACACS+ AAA database. The username is “jack.” The password is “cisco.” Two Cisco IOS configuration template variables are configured for “jack”: iosconfig1 and iosconfig2. The variables will replace $1 and $2 in the configuration template file. The subject name “CN=jack, O=cisco, C=US” is also configured. This subject name will replace the subject name field in the subsequent enrollment request (PKCS10) that is received from the petitioner device.

```
user = jack
    password = clear "cisco"

    service=tti
        ! The certificate server inserts the following subject name to the certificate.
        set subjectname="CN=jack, O=cisco, C=US"

        ! Up to nine template variables may be added.
        set iosconfig1="ntp server 10.3.0.1"
        set iosconfig2="hostname jack-vpn"
```

RADIUS AAA Server Database: Example

User information has been added to the RADIUS AAA server database in the following example. The username is “jack.” The password is “cisco.” Two Cisco IOS configuration template variables are configured for “jack”: iosconfig1 and iosconfig2. The variables will replace $1 and $2 in the configuration template file. The subject name “CN=jack, O=cisco, C=US” is also configured. This subject name will replace the subject name field in the subsequent enrollment request (PKCS10) that is received from the petitioner device.

```
user = jack
    password = clear "cisco"
    radius=cisco
    reply_attributes=9,1="tti:subjectname=CN=jack, O=cisco, C=US"
    ! Up to nine template variables may be added.
    9,1="tti:iosconfig1=ntp server 10.3.0.5"
    9,1="tti:iosconfig2=hostname jack-vpn"
```

AAA List on a TACACS+ and a RADIUS AAA Server: Example

The following is a configuration example showing that AAA authentication has been configured on a TACACS+ server and that AAA authorization has been configured on a RADIUS server.

Note Authentication and authorization usually point to the same server.

```
Router(config)# tacacs-server host 10.0.0.48 key cisco
Router(config)# aaa authentication login authen-tac group tacacs+
```

```
Router(config)# radius-server host 10.0.1.49 key cisco
Router(config)# aaa authorization network author-rad group radius
```

Configuration Template File: Example

The following output shows that a configuration template file has been configured and stored on a file system. Only two template variables are configured.

Note The lines called "iossnippet.txt" that are shown in this output are not included in the template file.

```
---iossnippet.txt---
$t
!
$c
! This variable will be replaced by iosconfig1, which is stored in the AAA database.
$1
! This variable will be replaced by iosconfig2, which is stored in the AAA database.
$2
!
end
---iossnippet.txt---
```

Note There are nine configuration template variables available ($1 through $9).

If a flexible configuration template file is needed, you can write your own CGI program to generate the template dynamically. An additional variable "$n" is provided for this type of usage, which will be expanded with the introducer name. For example, to use a CGI program "getconfig.cgi," which takes a username and dynamically generates the appropriate configuration snippet for that user, configure the following under the registrar:

```
Router (tti-registrar)# template config http://server/cgi-bin/getconfig.cgi?name=$n
```

You can also use a different configuration template file on the basis of the introducer name. For example, if you have multiple template files for different users, each with the username in the filename, configure the following under the registrar:

```
Router (tti-registrar)# template config tftp://server/config-$n.txt
```

Configuring the Petitioner and Registrar for Certificate-Based Authentication: Example

The following examples shows how to configure a petitioner to use the certificate issued by the trustpoint named mytrust:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)# crypto provisioning petitioner
Router(tti-petitioner)# trustpoint signing mytrust
Router(tti-petitioner)# end
```

The following example shows how to configure a registrar to verify the petitioner-signing certificate and to perform authorization lookups:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# crypto provisioning registrar
Router(tti-registrar)# authentication trustpoint mytrust
Router(tti-registrar)# authorization login certificate
Router(tti-registrar)# authorization username subjectname all
Router(tti-registrar)# end
```

Configuring an Administrative Introducer Using Authentication and Authorization Lists: Example

The following example shows how to configure an administrative introducer with the authentication list "authen-tac" and the authorization list "author-tac":

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)# crypto provisioning registrar
Router(tti-registrar)# administrator authentication list authen-tac
Router(tti-registrar)# administrator authorization list author-tac
Router(tti-registrar)# end
```

Additional References

The following sections provide references related to SDP.

Related Documents

Related Topic	Document Title
Certificate enrollment	"Configuring Certificate Enrollment for a PKI" module
Certificate server configuration	"Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment" module
PKI AAA integration concepts and configuration tasks	"Configuration Revocation and Authorization of Certificates in a PKI" module
PKI commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*, Release 12.3T

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Feature Information for SDP in a PKI

Table 61 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for a specific command was introduced, see the command reference documentation.

For information on a feature in this technology that is not documented here, see the Implementing and Managing PKI Features Roadmap.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Table 61 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 61 Feature Information for SDP in a PKI

Feature Name	Software Release	Feature Configuration Information
Administrative Secure Device Provisioning Introducer	12.3(14)T	This feature allows you to act as an administrative introducer to introduce a device into a PKI network and then provide a username as the device name for the record locator in the AAA database. The following sections provide information about this feature: • Authentication and Authorization Lists for an Administrative Introducer • Configuring an Administrative Introducer The following commands were introduced by this feature: **administrator authentication list**, **administrator authorization list**
Easy Secure Device Deployment	12.3(8)T	This feature introduces support for EzSDD, which offers a web-based enrollment interface that enables network administrators to deploy new devices in large networks. The following sections provide information about this feature: • Information About Setting Up SDP for Enrollment in a PKI • How to Set Up SDP for a PKI The following commands were introduced or modified by this feature: **crypto wui tti petitioner**, **crypto wui tti registrar**, **pki-server**, **template config**, **template username**, **trustpoint (tti-petitioner)**

Table 61 Feature Information for SDP in a PKI (continued)

Feature Name	Software Release	Feature Configuration Information
Easy Secure Device Deployment AAA Integration	12.3(8)T	This feature integrates an external AAA database, allowing the EzSDD introducer to be authenticated against a AAA database instead of having to use the enable password of the local Cisco certificate server. The following sections provide information about this feature: • How SDP Uses an External AAA Database • Enabling the SDP Registrar and Adding AAA Lists to the Server The following commands were introduced or modified by this feature: **authentication list (tti-registrar)**, **authorization list (tti-registrar)**, **debug crypto wui template config**, **template username**
Secure Device Provisioning (SDP) Certificate-Based Authorization	12.3(14)T	This feature feature allows certificates issued by other authority (CA) servers to be used for SDP introductions. The following sections provide information about this feature: • Use of a Self-Signed Certificate Versus a Certificate Issued by Another CA Server • Enabling the SDP Registrar for Certificate-Based Authorization The following commands were introduced by this feature: **administrator authentication list**, **administrator authorization list**

Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment

This module describes how to set up and manage a Cisco IOS Certificate Server (CS) for public key infrastructure (PKI) deployment. A CS embeds a simple certificate server, with limited certification authority (CA) functionality, into the Cisco IOS software. Thus, the following benefits are provided to the user:

- Easier PKI deployment by defining default behavior. The user interface is simpler because default behaviors are predefined. That is, you can leverage the scaling advantages of PKI without all of the certificate extensions that a CA provides, thereby allowing you to easily enable a basic PKI-secured network.
- Direct integration with Cisco IOS software.

Module History

This module was first published on May 2, 2005, and last updated on May 2, 2005.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all features. To find information about feature support and configuration, use the "Feature Information for the Cisco IOS Certificate Server" section on page 1556.

Contents

- Prerequisites for Configuring a Cisco IOS CS, page 1522
- Restrictions for Configuring a Cisco IOS Certificate Server, page 1522
- Information About Cisco IOS Certificate Servers, page 1522
- How to Set Up and Deploy a Cisco IOS Certificate Server, page 1526
- Configuration Examples for Using a Certificate Server, page 1546
- Where to Go Next, page 1555
- Additional References, page 1555
- Feature Information for the Cisco IOS Certificate Server, page 1556

Prerequisites for Configuring a Cisco IOS CS

Planning Your PKI Before Configuring the Certificate Server

Before configuring a Cisco IOS certificate server, it is important that you have planned for and chosen appropriate values for the settings you intend to use within your PKI (such as certificate lifetimes and certificate revocation list (CRL) lifetimes). After the settings have been configured in the certificate server and certificates have been granted, settings cannot be changed without having to reconfigure the certificate server and reenrolling the peers. For information on certificate server default settings and recommended settings, see the section "Certificate Server Default Values and Recommended Values."

Enabling an HTTP Server

The certificate server supports Simple Certificate Enrollment Protocol (SCEP) over HTTP. The HTTP server must be enabled on the router for the certificate server to use SCEP. (To enable the HTTP server, use the **ip http server** command.) The certificate server will automatically enable or disable SCEP services after the HTTP server is enabled or disabled. If the HTTP server is not enabled, only manual PKCS10 enrollment is supported.

Configuring Reliable Time Services

Time services must be running on the router because the certificate server must have reliable time knowledge. If a hardware clock is unavailable, the certificate server will depend on manually configured clock settings, such as Network Time Protocol (NTP). If there is not a hardware clock or the clock is invalid, the following message will be displayed at bootup:

```
% Time has not been set. Cannot start the Certificate server.
```

After the clock has been set, the certificate server will automatically switch to running status.

For information on manually configuring clock settings, see the section "Setting Time and Calendar Services" in the chapter "Performing Basic System Management" of the *Cisco IOS Network Management Configuration Guide*.

"crypto ca" to "crypto pki" command-line interface (CLI) change

As of Cisco IOS Release 12.3(7)T, all commands that begin as "crypto ca" have been changed to begin as "crypto pki." Although the router will still accept crypto ca commands, all output will be read back as crypto pki.

Restrictions for Configuring a Cisco IOS Certificate Server

The certificate server does not provide a mechanism for modifying the certificate request that is received from the client; that is, the certificate that is issued from the certificate server matches the requested certificate without modifications. If a specific certificate policy, such as name constraints, must be issued, the policy must be reflected in the certificate request.

Information About Cisco IOS Certificate Servers

Before setting up and deploying a certificate server in your PKI, you should understand the following concepts:

- RSA Key Pair and Certificate of the Certificate Server, page 1523
- Trustpoint of the Certificate Server, page 1524

- Certificate Revocation Lists (CRLs), page 1524
- Certificate Server Error Conditions, page 1525
- Certificate Enrollment Using a Certificate Server, page 1525

RSA Key Pair and Certificate of the Certificate Server

Note The certificate server automatically generates a 1024-bit Rivest, Shamir, and Adelman (RSA) key pair. You must manually generate an RSA key pair if you prefer a different size key pair. (For information on completing this task, see the section "Generating and Exporting a Certificate Server RSA Key Pair" later in this document.)

The certificate server will use a regular Cisco IOS RSA key pair as its CA key. This key pair must have the same name as the certificate server. If you do not generate the key pair before the certificate server is created on the router, a general-purpose key pair will be automatically generated during the configuration of the certificate server. As of Cisco IOS Release 12.3(11)T, the CA certificate and CA key can be backed up (archived) automatically just once after they are generated by the certificate server. As a result, it is not necessary to generate an exportable CA key if CA backup is desirable.

What to Do with Automatically Generated Key Pairs in Cisco IOS software Prior to Release 12.3(11)T

If the key pair is automatically generated, it will not be marked as exportable. Thus, you must manually generate the key pair as exportable if you want to back up the CA key. (For information on how to complete this task, see the section Generating and Exporting a Certificate Server RSA Key Pair.)

How the CA Certificate and CA Key Are Automatically Archived

At initial certificate server setup, you can enable the CA certificate and the CA key to be automatically archived so that they may be restored later if either the original copy or the original configuration is lost.

When the certificate server is turned on the first time, the CA certificate and CA key will be generated. If automatic archive is also enabled, the CA certificate and the CA key will be exported (archived) to the server database. The archive can be in PKCS12 or privacy-enhanced mail (PEM) format.

Note

- This CA key backup file is extremely important and should be moved immediately to another secured place.
- This archiving action occurs only one time. Only the CA key that is (1) manually generated and marked exportable or (2) automatically generated by the certificate server will be archived (this key will be marked nonexportable).
- Autoarchiving will not occur if you generate the CA key manually and mark it "nonexportable."
- In addition to the CA certificate and CA key archive file, you should also regularly back up the serial file (.ser) and the CRL file (.crl). The serial file and the CRL file are both critical for CA operation if you need to restore your certificate server.

It is not possible to manually back up a server that uses nonexportable RSA keys or manually generated nonexportable RSA keys. Although automatically generated RSA keys are marked as nonexportable, they are automatically archived once.

Trustpoint of the Certificate Server

The certificate server will also have an automatically generated trustpoint of the same name; the trustpoint will store the certificate of the certificate server. After the router detects that a trustpoint is being used to store the certificate of the certificate server, the trustpoint will be locked so that it cannot be modified. (Before configuring the certificate server, you can manually create and set up this trustpoint [using the **crypto pki trustpoint**], which allows you to specify an alternative RSA key pair [using the **rsakeypair** command]).

Note The automatically generated trustpoint and the certificate server certificate are not available for the certificate server device identity. Thus, any command-line interface (CLI) (such as the **ip http secure-trustpoint** command) that is used to specify the CA trustpoint to obtain certificates and authenticate the connecting client's certificate must point to an additional trustpoint configured on the certificate server device.

If the server is a root certificate server, it will use the RSA key pairs and several other attributes to generate a self-signed certificate. The associated CA certificate will have the following key usage extensions—Digital Signature, Certificate Sign, and CRL (certificate revocation list) Sign.

After the CA certificate is generated, attributes can be changed only if the certificate server is destroyed.

Certificate Revocation Lists (CRLs)

By default, CRLs are issued once every 168 hours (1 week). To specify a value other than the default value for the CRL, issue the **lifetime crl** command. After the CRL is issued, it is written to the specified database location as *ca-label*.crl (where *ca-label* is the name of the certificate server). It is the responsibility of the network administrator to ensure that the CRL is available from the location that is specified via the **cdp-url** command. If the **cdp-url** command is not specified, the CRL distribution point (CDP) certificate extension will not be included in the certificates that are issued by the certificate server. Thus, Cisco IOS PKI clients will automatically use SCEP to retrieve a CRL from the certificate server, which puts an additional load on the certificate server because it must provide SCEP server support for each CRL request.

Note

- The CRL will always be available via SCEP, which is enabled by default, if the HTTP server is enabled.
- If many peer devices will be checking CRLs, it is recommended that you configure an HTTP-based CDP; for example, CDP URL http://myhttpserver.company.com/mycs.crl.

The CDP URL may be changed after the certificate server is running, but existing certificates will not be reissued with the new CDP that is specified via the **cdp-url** command.

When a new CRL is issued, the certificate server uses its current cached CRL to generate a new CRL (When the certificate server is rebooted, it reloads the current CRL from the database.) A new CRL is issued after a certificate is revoked from the CLI.

Note A new CRL cannot be issued unless the current CRL is expired.

The certificate server supports only one CDP; thus, all certificates that are issued include the same CDP.

Certificate Server Error Conditions

At startup, the certificate server checks the current configuration before issuing any certificates. It reports the last known error conditions via the **show crypto pki server** command output. Example errors can include any of the following conditions:

- Storage inaccessible
- Waiting for HTTP server
- Waiting for time setting

If the certificate server experiences a critical failure at any time, such as failing to publish a CRL, the certificate server will automatically enter a disabled state. This state allows the network administrator to fix the condition; thereafter, the certificate server will return to the previous normal state.

Certificate Enrollment Using a Certificate Server

A certificate enrollment request functions as follows:

- The certificate server receives the enrollment request from an end user, and the following actions occur:
  - A request entry is created in the enrollment request database with the initial state. (See Table 62 for a complete list of certificate enrollment request states.)
  - The certificate server refers to the CLI configuration (or the default behavior any time a parameter is not specified) to determine the authorization of the request. Thereafter, the state of the enrollment request is updated in the enrollment request database.
- At each SCEP query for a response, the certificate server examines the current request and performs one of the following actions:
  - Responds to the end user with a “pending” or “denied” state.
  - Generates and signs the appropriate certificate and stores the certificate in the enrollment request database.

If the connection of the client has closed, the certificate server will wait for the client to request another certificate.

All enrollment requests transition through the certificate enrollment states that are defined in Table 62. To see current enrollment requests, use the **crypto pki server request pkcs10** command.

Table 62 Certificate Enrollment Request State Descriptions

Certificate Enrollment State	Description
authorized	The certificate server has authorized the request.
denied	The certificate server has denied the request for policy reasons.
granted	The CA core has generated the appropriate certificate for the certificate request.
initial	The request has been created by the SCEP server.

Table 62 Certificate Enrollment Request State Descriptions (continued)

Certificate Enrollment State	Description
malformed	The certificate server has determined that the request is invalid for cryptographic reasons.
pending	The enrollment request must be manually accepted by the network administrator.

SCEP Reenrollment

All SCEP requests are treated as new certificate enrollment requests, even if the request specifies a duplicate subject name or public key pair as a previous certificate request. When servicing an enrollment request, there are no extended database looups.

How to Set Up and Deploy a Cisco IOS Certificate Server

This section contains the following procedures:

- Generating and Exporting a Certificate Server RSA Key Pair, page 1526
- Configuring a Certificate Server, page 1528
- Configuring Certificate Server Functionality, page 1530
- Configuring a Proxy to Offload the Root Certificate Server, page 1533
- Maintaining, Verifying, and Troubleshooting the Certificate Server, Certificates, and the CA, page 1541

Generating and Exporting a Certificate Server RSA Key Pair

Perform this task to manually generate an RSA key pair as exportable for the certificate server. If this task is not performed, the certificate server will automatically generate a key pair, which will not be marked as exportable.

Note Automatically archiving the CA certificate and the CA key was introduced in Cisco IOS Release 12.3(11)T. As a result, this task ("Generating and Exporting a Certificate Server RSA Key Pair") is no longer necessary if the automatic archive functionality meets your backup needs.

Tip In addition to keeping the private key in a secure location, it is recommended that you regularly archive the certificate server database.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto key generate rsa** {**general-keys** | **usage-keys**} [**label** *key-label*] [**modulus** *modulus-size*] [**exportable**]

4. **crypto key export rsa** *key-label* **pem** {**terminal** | **url** *url*} {**3des** | **des**} *passphrase*
5. **crypto key import rsa** *key-label* **pem** [**usage-keys**] {**terminal** | **url** *url*} [**exportable**] *passphrase*
6. **exit**
7. **show crypto key mypubkey rsa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto key generate rsa** {**general-keys** | **usage-keys**} [**label** *key-label*] [**modulus** *modulus-size*] [**exportable**] **Example:** `Router (config)# crypto key generate rsa general-keys label mycs exportable`	Generates the RSA key pair for the certificate server. **Note** You must use the same name for the key pair (*key-label*) that you plan to use for the certificate server (via the **crypto pki server** *cs-label* command). **Note** If you manually generate the exportable RSA key pair but wait until after the CA certificate has been generated before issuing the **no shutdown** command, you can use the **crypto ca export pkcs12** command to export a PKCS12 file that contains the certificate server certificate as well as the private key.
Step 4	**crypto key export rsa** *key-label* **pem** {**terminal** | **url** *url*} {**3des** | **des**} *passphrase* **Example:** `Router (config)# crypto key export rsa mycs pem url nvram: 3des PASSWORD`	Exports the generated RSA key pair.
Step 5	**crypto key import rsa** *key-label* **pem** [**usage-keys**] {**terminal** | **url** *url*} [**exportable**] *passphrase* **Example:** `Router (config)# crypto key import rsa mycs2 pem url nvram: mycs PASSWORD`	(Optional) Changes the RSA key pair to nonexportable. **Note** If you do not want the RSA key to continue to be exportable from your certificate server, import the key back to the certificate server without the **exportable** keyword. Thus, the key cannot be exported again.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Router (config)# exit`	Exits global configuration.
Step 7	**show crypto key mypubkey rsa** **Example:** `Router# show crypto key mypubkey rsa`	Displays the RSA public keys of your router.

Configuring a Certificate Server

Perform this task to configure a Cisco IOS certificate server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http server**
4. **crypto pki server** *cs-label*
5. **no shutdown**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip http server** **Example:** `Router (config)# ip http server`	Enables the HTTP server on your system.

	Command or Action	Purpose
Step 4	**crypto pki server** *cs-label* **Example:** `Router (config)# crypto pki server server-pki`	Defines a label for the certificate server and enters certificate server configuration mode. **Note** If you manually generated an RSA key pair, the *cs-label* argument must match the name of the key pair.
Step 5	**no shutdown** **Example:** `Router (cs-server)# no shutdown`	(Optional) Enables the certificate server. **Note** Only use this command at this point if you want to use the preconfigured default functionality. That is, do not issue this command just yet if you plan to change any of the default settings as shown in the task "Configuring Certificate Server Functionality."

Examples

The following example shows how to configure the certificate server "ca":

```
Router(config)# crypto pki server ca
Router(cs-server)# no shutdown
% Once you start the server, you can no longer change some of
% the configuration.
Are you sure you want to do this? [yes/no]: yes
% Generating 1024 bit RSA keys ...[OK]

% Certificate Server enabled.
Router(cs-server)# end
!
Router# show crypto pki server

Certificate Server ca:
    Status: enabled, configured
    CA cert fingerprint: 5A856122 4051347F 55E8C246 866D0AC3
    Granting mode is: manual
    Last certificate issued serial number: 0x1
    CA certificate expiration timer: 19:44:57 GMT Oct 14 2006
    CRL NextUpdate timer: 19:45:25 GMT Oct 22 2003
    Current storage dir: nvram:
    Database Level: Complete - all issued certs written as <serialnum>.cer
```

What to Do Next

After you have configured a certificate server, you can use the preconfigured default values or specify values via the CLI for the functionality of the certificate server. If you choose to specify values other than the defaults, see the following section, "Configuring Certificate Server Functionality."

Configuring Certificate Server Functionality

After you have enabled a certificate server and are in certificate server configuration mode, use any of the steps in this task to configure basic certificate server functionality values other than the default values.

Certificate Server Default Values and Recommended Values

The default values for a certificate server are intended to address a relatively small network (of about ten devices). For example, the database settings are minimal (via the **database level minimal** command) and the certificate server handles all CRL requests via SCEP. For larger networks, it is recommended that you use either the database setting "names" or "complete" (as described in the **database level** command) for possible audit and revocation purposes. Depending on the CRL checking policy, you should also use an external CDP in a larger network.

SUMMARY STEPS

1. **database url** *root-url*
2. **database level** {**minimal** | **names** | **complete**}
3. **database username** *username* [**password** [*encr-type*] *password*]
4. **database archive** {**pkcs12** | **pem**} [**password** [*encr-type*] *password*]
5. **issuer-name** *DN-string*
6. **lifetime** {**ca-certificate** | **certificate**} *time*
7. **lifetime crl** *time*
8. **lifetime enrollment-request** *time*
9. **cdp-url** *url*
10. **no shutdown**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**database url** *root-url* **Example:** `Router (cs-server)# database url tftp://cert-svr-db.company.com`	Specifies the location where all database entries for the certificate server will be written out. If this command is not specified, all database entries will be written to NVRAM.
Step 2	**database level** {**minimal** | **names** | **complete**} **Example:** `Router (cs-server)# database level complete`	Controls what type of data is stored in the certificate enrollment database. • **minimal**—Enough information is stored only to continue issuing new certificates without conflict; the default value. • **names**—In addition to the information given in the minimal level, the serial number and subject name of each certificate. • **complete**—In addition to the information given in the minimal and names levels, each issued certificate is written to the database. **Note** The **complete** keyword produces a large amount of information; if it is issued, you should also specify an external TFTP server in which to store the data via the **database url** command.
Step 3	**database username** *username* [**password** [*encr-type*] *password*] **Example:** `Router (cs-server)# database username lucy password PASSWORD`	(Optional) Sets a username and password when a user is required to access a certificate enrollment database storage location.
Step 4	**database archive** {**pkcs12** | **pem**} [**password** [*encr-type*] *password*] **Example:** `Router (cs-server)# database archive pem`	(Optional) Sets the CA key and CA certificate archive format and password to encrypt the file. The default value is **pkcs12**, so if this subcommand is not configured, autoarchiving will still be done, and the PKCS12 format will be used. • The password is optional. If it is not configured, you will be prompted for the password when the server is turned on for the first time. **Note** It is recommended that you remove the password from the configuration after the archive is finished.
Step 5	**issuer-name** *DN-string* **Example:** `Router (cs-server)# issuer-name my-server`	(Optional) Sets the CA issuer name to the specified distinguished name (*DN-string*). The default value is as follows: **issuer-name cn=**{*cs-label*}.

	Command or Action	Purpose
Step 6	**lifetime** {**ca-certificate** | **certificate**} *time* **Example:** `Router (cs-server)# lifetime certificate 888`	(Optional) Specifies the lifetime, in days, of a CA certificate or a certificate. Valid values range from 1 day to 1825 days. The default CA certificate lifetime is 3 years; the default certificate lifetime is 1 year. The maximum certificate lifetime is 1 month less than the lifetime of the CA certificate.
Step 7	**lifetime crl** *time* **Example:** `Router (cs-server)# lifetime crl 333`	(Optional) Defines the lifetime, in hours, of the CRL that is used by the certificate server. Maximum lifetime value is 336 hours (2 weeks). The default value is 168 hours (1 week).
Step 8	**lifetime enrollment-request** *time* **Example:** `Router (cs-server)# lifetime enrollment-request 888`	(Optional) Specifies how long an enrollment request should stay in the enrollment database before being removed. Maximum lifetime is 1000 hours.
Step 9	**cdp-url** *url* **Example:** `Router (cs-server)# cdp-url http://my-cdp.company.com`	(Optional) Defines a CDP to be used in the certificates that are issued by the certificate server. URL must be an HTTP URL. **Note** Although this command is optional, it is strongly recommended for any deployment scenario.
Step 10	**no shutdown** **Example:** `Router (cs-server)# no shutdown`	Enables the certificate server. You should issue this command only after you have completely configured your certificate server.

Examples

After a certificate server has been enabled on a router, the **show crypto pki server** command displays the following output:

```
Router# show crypto pki server

    Certificate Server status:enabled, configured
    Granting mode is:manual
    Last certificate issued serial number:0x1
    CA certificate expiration timer:19:31:15 PST Nov 17 2006
    CRL NextUpdate timer:19:31:15 PST Nov 25 2003
    Current storage dir:nvram:
    Database Level:Minimum - no cert data written to storage
```

Configuring a Proxy to Offload the Root Certificate Server

This section contains the following tasks that explain how to set up a proxy—via a Registration authority (RA) mode certificate server or a subordinate certificate server—to offload the root certificate server.

- Configuring a Certificate Server to Run in RA Mode by Configuring the RA Mode Certificate Server, page 1533
- Configuring the Root Certificate Server to Delegate Enrollment Tasks to the RA Mode Certificate Server, page 1535
- Configuring a Subordinate Certificate Server, page 1536

Configuring a Certificate Server to Run in RA Mode by Configuring the RA Mode Certificate Server

After a certificate server is running as a CA, you may want to configure an RA mode certificate server on another device and delegate the task of enrollment request handling to that device. Why Configure a Certificate Server for RA Mode?

RA is the authority charged with recording or verifying some or all of the data required for the CA to issue certificates. In many cases the CA will undertake all of the RA functions itself, but where a CA operates over a wide geographical area or when there is security concern over exposing the CA at the edge of the network, it may be administratively advisable to delegate some of the tasks to an RA and leave the CA to concentrate on its primary tasks of signing certificates and CRLs.

A Cisco IOS certificate server can be configured to run in RA mode. When the RA receives a SCEP or manual enrollment request, the administrator can either reject or grant it on the basis of local policy. If the request is granted, it will be forwarded to the issuing CA, and the CA will automatically generate the certificate and return it to the RA. The client can later retrieve the granted certificate from the RA.

Restrictions for Configuring a Certificate Server for RA Mode

When the Cisco IOS certificate server is acting as an RA, the issuing CA should be a Cisco IOS certificate server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment url** *url*
5. **subject-name** *x.500-name*
6. **exit**
7. **crypto pki server** *cs-label*
8. **mode ra**
9. **no shutdown**
10. **no shutdown**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Router (config)# crypto pki trustpoint ra-server`	Declares the trustpoint that your RA mode certificate server should use and enters ca-trustpoint configuration mode.
Step 4	**enrollment url** *url* **Example:** `Router (ca-trustpoint)# enrollment url http://ca-server.company.com`	Specifies the enrollment URL of the issuing CA certificate server (root certificate server).
Step 5	**subject-name** *x.500-name* **Example:** `Router (ca-trustpoint)# subject-name cn=ioscs RA`	(Optional) Specifies the subject name the RA will use. **Note** Include "cn=ioscs RA" or "ou=ioscs RA" in the subject name so that the issuing CA certificate server can recognize the RA (see Step 7 below).
Step 6	**exit** **Example:** `Router (ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode.
Step 7	**crypto pki server** *cs-label* **Example:** `Router(config)# crypto pki server ra-server`	Enables a Cisco IOS certificate server and enters cs-server configuration mode. **Note** The certificate server must have the same name as the trustpoint that was created in Step 3 above.
Step 8	**mode ra** **Example:** `Router(cs-server)# mode ra`	Places the PKI server into RA certificate server mode.

	Command or Action	Purpose
Step 9	**no shutdown** **Example:** `Router(cs-server)# no shutdown`	Enables the certificate server. **Note** After this command is issued, the RA will automatically enroll with the root certificate server. After the RA certificate has been successfully received, you must issue the **no shutdown** command again, which reenables the certificate server.
Step 10	**no shutdown** **Example:** `Router(cs-server)# no shutdown`	Reenables the certificate server.

Configuring the Root Certificate Server to Delegate Enrollment Tasks to the RA Mode Certificate Server

Perform the following steps on the router that is running the issuing certificate server; that is, configure the root certificate server that is delegating enrollment tasks to the RA mode certificate server.

Note Granting enrollment requests for an RA is essentially the same process as granting enrollment requests for client devices—except that enrollment requests for an RA are displayed in the section "RA certificate requests" of the command output for the **crypto pki server info-requests** command.

SUMMARY STEPS

1. **enable**
2. **crypto pki server** *cs-label* **info requests**
3. **crypto pki server** *cs-label* **grant** *req-id*
4. **configure terminal**
5. **crypto pki server** *cs-label*
6. **grant ra-auto**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**crypto pki server** *cs-label* **info requests** **Example:** `Router# crypto pki server root-server info requests`	Displays the outstanding RA certificate request. **Note** This command is issued on the router that is running the issuing certificate server.

	Command or Action	Purpose
Step 3	**crypto pki server** *cs-label* **grant** *req-id* **Example:** `Router# crypto pki server root-server grant 9`	Grants the pending RA certificate request. **Note** Because the issuing certificate server will delegate the enrollment request verification task to the RA, you must pay extra attention to the RA certificate request before granting it.
Step 4	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 5	**crypto pki server** *cs-label* **Example:** `Router (config)# crypto pki server root-server`	Enables a Cisco IOS certificate server and enters cs-server configuration mode.
Step 6	**grant ra-auto** **Example:** `Router(cs-server)# grant ra-auto`	(Optional) Specifies that all enrollment requests from an RA are to be granted automatically. **Note** For the **grant ra-auto** command to work, you have to include "cn=ioscs RA" or "ou=iosc RA" in the subject name of the RA certificate. (See Step 2 above.)

Configuring a Subordinate Certificate Server

Because the root RSA key pairs are extremely important in a PKI hierarchy, it is often advantageous to keep them offline or archived. To support this requirement, PKI hierarchies allow for subordinate CAs that have been signed by the root authority. In this way, the root authority can be kept offline (except to issue occasional CRL updates), and the subordinate CA can be used during normal operation.

The subordinate certificate server provides all the same features as a root certificate server.

Note Enrollment requests that come from a subordinate certificate server must always be manually granted.

Perform this task to configure a subordinate certificate server to grant all or certain SCEP or manual certificate requests.

Restrictions

- You must be running Cisco IOS Release 12.3(14)T or later. (Versions prior to Cisco IOS software Release 12.3(14)T support only one certificate server and no hierarchy; that is, subordinate certificate servers are not supported.)
- The root certificate server should be a Cisco IOS certificate server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*

4. **enrollment url** *url*
5. **exit**
6. **crypto pki server** *cs-label*
7. **issuer name** *DN-string*
8. **mode sub-cs**
9. **no shutdown**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Router (config)# crypto pki trustpoint sub`	Declares the trustpoint that your subordinate certificate server should use and enters ca-trustpoint configuration mode.
Step 4	**enrollment url** *url* **Example:** `Router (ca-trustpoint)# enrollment url http://10.3.0.6`	Specifies the enrollment URL of the issuing CA certificate server (root certificate server).
Step 5	**exit** **Example:** `Router (ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode.
Step 6	**crypto pki server** *cs-label* **Example:** `Router(config)# crypto pki server sub`	Enables a Cisco IOS certificate server and enters cs-server configuration mode. **Note** The subordinate server must have the same name as the trustpoint that was created in Step 3 above.
Step 7	**issuer name** *DN-string* ***Example:*** `Router(cs-server)# issuer-name CN=sub CA, O=Cisco, C=us`	(Optional) Specifies the DN as the CA issuer name for the certificate server.

	Command or Action	Purpose
Step 8	**mode sub-cs** **Example:** `Router(cs-server)# mode sub-cs`	Places the PKI server into sub-certificate server mode.
Step 9	**no shutdown** **Example:** `Router(cs-server)# no shutdown`	Enable or reenables the certificate server. • If this is the first time that a subordinate certificate server is enabled, the certificate server will generate the key and obtain its signing certificate from the root certificate server.

Examples

If the certificate server fails to enable or if the certificate server has trouble handling the request that has been configured, you can use the **debug crypto pki server** command to troubleshoot your configuration as shown in the following examples (Clock Not Set and Trustpoint Not Configured):

```
Router# debug crypto pki server
```

Clock Not Set

```
Router (config)# crypto pki server sub
Router (cs-server)# mode sub-cs
Router (cs-server)# no shutdown
%Some server settings cannot be changed after CA certificate generation.
% Please enter a passphrase to protect the private key % or type Return to exit
Password:
*Jan  6 20:57:37.667: CRYPTO_CS: enter FSM: input state initial, input signal no shut

Re-enter password:
% Generating 1024 bit RSA keys ...
*Jan  6 20:57:45.303: CRYPTO_CS: starting enabling checks
*Jan  6 20:57:45.303: CRYPTO_CS: key 'sub' does not exist; generated automatically[OK]
% Time has not been set. Cannot start the Certificate server
```

Trustpoint Not Configured

```
Router (config)# crypto pki server sub
Router (cs-server)# mode sub-cs
Router (cs-server)# no shutdown
%Some server settings cannot be changed after CA certificate generation.
% Please enter a passphrase to protect the private key or type Return to exit Password:
*Jan  6 21:00:15.961: CRYPTO_CS: enter FSM: input state initial, input signal no shut. Jan
6 21:03:34.309: CRYPTO_CS: enter FSM: input state initial, input signal time set. Jan 6
21:03:34.313: CRYPTO_CS: exit FSM: new state initial.
Jan  6 21:03:34.313: CRYPTO_CS: cs config has been unlocked Re-enter password:
% Generating 1024 bit RSA keys ...
Jan  6 21:03:44.413: CRYPTO_CS: starting enabling checks
Jan  6 21:03:44.413: CRYPTO_CS: associated trust point 'sub' does not exist; generated
automatically
Jan  6 21:03:44.417: CRYPTO_CS: key 'sub' does not exist; generated automatically[OK]

Jan  6 21:04:03.993: CRYPTO_CS: nvram filesystem
Jan  6 21:04:04.077: CRYPTO_CS: serial number 0x1 written.
You must specify an enrollment URL for this CA before you can authenticate it.
% Failed to authenticate the Certificate Authority
```

If the certificate server fails to obtain its signing certificate from the root certificate server, you can use the **debug crypto pki transactions** command to troubleshoot your configuration as shown in the following example:

```
Router# debug crypto pki transactions

Jan  6 21:07:00.311: CRYPTO_CS: enter FSM: input state initial, input signal time set Jan
6 21:07:00.311: CRYPTO_CS: exit FSM: new state initial
Jan  6 21:07:00.311: CRYPTO_CS: cs config has been unlockedno sh
%Some server settings cannot be changed after CA certificate generation.
% Please enter a passphrase to protect the private key % or type Return to exit Password:
Jan  6 21:07:03.535: CRYPTO_CS: enter FSM: input state initial, input signal no shut

Re-enter password:
% Generating 1024 bit RSA keys ...
Jan  6 21:07:10.619: CRYPTO_CS: starting enabling checks
Jan  6 21:07:10.619: CRYPTO_CS: key 'sub' does not exist; generated automatically[OK]

Jan  6 21:07:20.535: %SSH-5-ENABLED: SSH 1.99 has been enabled
Jan  6 21:07:25.883: CRYPTO_CS: nvram filesystem
Jan  6 21:07:25.991: CRYPTO_CS: serial number 0x1 written.
Jan  6 21:07:27.863: CRYPTO_CS: created a new serial file.
Jan  6 21:07:27.863: CRYPTO_CS: authenticating the CA 'sub'
Jan  6 21:07:27.867: CRYPTO_PKI: Sending CA Certificate Request:
GET /cgi-bin/pkiclient.exe?operation=GetCACert&message=sub HTTP/1.0
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Cisco PKI)

Jan  6 21:07:27.867: CRYPTO_PKI: can not resolve server name/IP address
Jan 6 21:07:27.871: CRYPTO_PKI: Using unresolved IP Address 10.3.0.6 Certificate has the
following attributes:
     Fingerprint MD5: 328ACC02 52B25DB8 22F8F104 B6055B5B
     Fingerprint SHA1: 02FD799D DD40C7A8 61DC53AB 1E89A3EA 2A729EE2

% Do you accept this certificate? [yes/no]:
Jan  6 21:07:30.879: CRYPTO_PKI: http connection opened
Jan 6 21:07:30.903: CRYPTO_PKI: HTTP response header:
  HTTP/1.1 200 OK
Date: Thu, 06 Jan 2005 21:07:30 GMT
Server: cisco-IOS
Content-Type: application/x-x509-ca-cert
Expires: Thu, 06 Jan 2005 21:07:30 GMT
Last-Modified: Thu, 06 Jan 2005 21:07:30 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Accept-Ranges: none

Content-Type indicates we have received a CA certificate.

Jan  6 21:07:30.903: Received 507 bytes from server as CA certificate:
Jan  6 21:07:30.907: CRYPTO_PKI: transaction GetCACert completed
Jan  6 21:07:30.907: CRYPTO_PKI: CA certificate received.
Jan  6 21:07:30.907: CRYPTO_PKI: CA certificate received.
Jan  6 21:07:30.927: CRYPTO_PKI: crypto_pki_authenticate_tp_cert()

Jan  6 21:07:30.927: CRYPTO_PKI: trustpoint sub authentication status = 0 y Trustpoint CA
certificate accepted.%

% Certificate request sent to Certificate Authority

% Enrollment in progress...
Router (cs-server)#
Jan  6 21:07:51.772: CRYPTO_CA: certificate not found
Jan  6 21:07:51.772: CRYPTO_CA: certificate not found
Jan  6 21:07:52.460: CRYPTO_CS: Publishing 213 bytes to crl file nvram:sub.crl
```

```
Jan 6 21:07:54.348: CRYPTO_CS: enrolling the server's trustpoint 'sub'
Jan  6 21:07:54.352: CRYPTO_CS: exit FSM: new state check failed
Jan 6 21:07:54.352: CRYPTO_CS: cs config has been locked
Jan  6 21:07:54.356: CRYPTO_PKI: transaction PKCSReq completed
Jan  6 21:07:54.356: CRYPTO_PKI: status:
Jan  6 21:07:55.016: CRYPTO_PKI:  Certificate Request Fingerprint MD5: 1BA027DB 1C7860C7
EC188F65 64356C80
Jan  6 21:07:55.016: CRYPTO_PKI:  Certificate Request Fingerprint SHA1: 840DB52C E17614CB
0C7BE187 0DFC884D D32CAA75
Jan  6 21:07:56.508: CRYPTO_PKI: can not resolve server name/IP address
Jan  6 21:07:56.508: CRYPTO_PKI: Using unresolved IP Address 10.3.0.6
Jan  6 21:07:56.516: CRYPTO_PKI: http connection opened
Jan  6 21:07:59.136: CRYPTO_PKI: received msg of 776 bytes
Jan  6 21:07:59.136: CRYPTO_PKI: HTTP response header:
  HTTP/1.1 200 OK
  Date: Thu, 06 Jan 2005 21:07:57 GMT
  Server: cisco-IOS
  Content-Type: application/x-pki-message
  Expires: Thu, 06 Jan 2005 21:07:57 GMT
  Last-Modified: Thu, 06 Jan 2005 21:07:57 GMT
  Cache-Control: no-store, no-cache, must-revalidate
  Pragma: no-cache
  Accept-Ranges: none


Jan 6 21:07:59.324: The PKCS #7 message has 1 verified signers.
Jan 6 21:07:59.324: signing cert: issuer=cn=root1
Jan 6 21:07:59.324: Signed Attributes:

Jan 6 21:07:59.328: CRYPTO_PKI: status = 102: certificate request pending
Jan 6 21:08:00.788: CRYPTO_PKI: can not resolve server name/IP address
Jan 6 21:08:00.788: CRYPTO_PKI: Using unresolved IP Address 10.3.0.6
Jan 6 21:08:00.796: CRYPTO_PKI: http connection opened
Jan 6 21:08:11.804: CRYPTO_PKI: received msg of 776 bytes
Jan 6 21:08:11.804: CRYPTO_PKI: HTTP response header: HTTP/1.1 200 OK
  Date: Thu, 06 Jan 2005 21:08:01 GMT
  Server: cisco-IOS
  Content-Type: application/x-pki-message
  Expires: Thu, 06 Jan 2005 21:08:01 GMT
  Last-Modified: Thu, 06 Jan 2005 21:08:01 GMT
  Cache-Control: no-store, no-cache, must-revalidate
  Pragma: no-cache
  Accept-Ranges: none


Jan  6 21:08:11.992: The PKCS #7 message has 1 verified signers.
Jan 6 21:08:11.992: signing cert: issuer=cn=root1
Jan  6 21:08:11.996: Signed Attributes:

Jan  6 21:08:11.996: CRYPTO_PKI: status = 102: certificate request pending
Jan  6 21:08:21.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan  6 21:08:31.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan  6 21:08:41.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan  6 21:08:51.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan  6 21:09:01.996: CRYPTO_PKI: All sockets are closed for trustpoint sub.
Jan  6 21:09:11.996: CRYPTO_PKI: resend GetCertInitial, 1 Jan  6 21:09:11.996: CRYPTO_PKI:
All sockets are closed for trustpoint sub.
Jan  6 21:09:11.996: CRYPTO_PKI: resend GetCertInitial for session: 0
Jan  6 21:09:11.996: CRYPTO_PKI: can not resolve server name/IP address
Jan  6 21:09:11.996: CRYPTO_PKI: Using unresolved IP Address 10.3.0.6
Jan  6 21:09:12.024: CRYPTO_PKI: http connection opened% Exporting Certificate Server
signing certificate and keys...

Jan  6 21:09:14.784: CRYPTO_PKI: received msg of 1611 bytes
```

```
Jan  6 21:09:14.784: CRYPTO_PKI: HTTP response header:
  HTTP/1.1 200 OK
  Date: Thu, 06 Jan 2005 21:09:13 GMT
  Server: cisco-IOS
  Content-Type: application/x-pki-message
  Expires: Thu, 06 Jan 2005 21:09:13 GMT
  Last-Modified: Thu, 06 Jan 2005 21:09:13 GMT
  Cache-Control: no-store, no-cache, must-revalidate
  Pragma: no-cache
  Accept-Ranges: none


Jan  6 21:09:14.972: The PKCS #7 message has 1 verified signers.
Jan 6 21:09:14.972: signing cert: issuer=cn=root1
Jan  6 21:09:14.972: Signed Attributes:

Jan  6 21:09:14.976: CRYPTO_PKI: status = 100: certificate is granted
Jan  6 21:09:15.668: The PKCS #7 message contains 1 certs and 0 crls.
Jan  6 21:09:15.688: Newly-issued Router Cert: issuer=cn=root serial=2
Jan  6 21:09:15.688: start date: 21:08:03 GMT Jan 6 2005
Jan  6 21:09:15.688: end date: 21:08:03 GMT Jan 6 2006
Jan  6 21:09:15.688: Router date: 21:09:15 GMT Jan 6 2005
Jan  6 21:09:15.692: Received router cert from CA
Jan  6 21:09:15.740: CRYPTO_CA: certificate not found
Jan  6 21:09:15.744: CRYPTO_PKI: All enrollment requests completed for trustpoint sub. Jan
6 21:09:15.744: %PKI-6-CERTRET: Certificate received from Certificate Authority Jan  6
21:09:15.744: CRYPTO_PKI: All enrollment requests completed for trustpoint sub. Jan  6
21:09:15.744: CRYPTO_PKI: All enrollment requests completed for trustpoint sub. Jan 6
21:09:15.748: CRYPTO_CS: enter FSM: input state check failed, input signal cert configured
Jan  6 21:09:15.748: CRYPTO_CS: starting enabling checks
Jan  6 21:09:15.748: CRYPTO_CS: nvram filesystem
Jan  6 21:09:15.796: CRYPTO_CS: found existing serial file.
Jan  6 21:09:15.820: CRYPTO_CS: old router cert flag 0x4
Jan  6 21:09:15.820: CRYPTO_CS: new router cert flag 0x44
Jan  6 21:09:18.432: CRYPTO_CS: DB version 1
Jan  6 21:09:18.432: CRYPTO_CS: last issued serial number is 0x1
Jan  6 21:09:18.480: CRYPTO_CS: CRL file sub.crl exists.
Jan  6 21:09:18.480: CRYPTO_CS: Read 213 bytes from crl file sub.crl.
Jan  6 21:09:18.532: CRYPTO_CS: SCEP server started
Jan  6 21:09:18.532: CRYPTO_CS: exit FSM: new state enabled
Jan  6 21:09:18.536: CRYPTO_CS: cs config has been locked
Jan 6 21:09:18.536: CRYPTO_PKI: All enrollment requests completed for trustpoint sub.
```

If the certificate server fails to enable or if the certificate server has trouble handling the request that has been configured, you can use the **debug crypto pki server** command to troubleshoot the progress of an enrollment. This command can also be used to debug the root CA (turn it on at the root CA).

Maintaining, Verifying, and Troubleshooting the Certificate Server, Certificates, and the CA

Use the tasks in this section to help maintain, verify, and troubleshoot the certificate server, as appropriate:

- Managing the Enrollment Request Database, page 1542
- Removing Requests from the Enrollment Request Database, page 1543
- Deleting a Certificate Server, page 1544
- Verifying and Troubleshooting Certificate Server, Certificate, and CA Status, page 1545

Managing the Enrollment Request Database

SCEP supports two client authentication mechanisms—manual and preshared key. Manual enrollment requires the administrator at the CA server to specifically authorize the enrollment requests; enrollment using preshared keys allows the administrator to preauthorize enrollment requests by generating a one-time password (OTP).

Use any of the optional steps within this task to help manage the enrollment request database by performing functions such as specifying enrollment processing parameters that are to be used by SCEP and by controlling the run-time behavior or the certificate server.

SUMMARY STEPS

1. **enable**
2. **crypto pki server** *cs-label* **grant** {**all** | *req-id*}
3. **crypto pki server** *cs-label* **reject** {**all** | *req-id*}
4. **crypto pki server** *cs-label* **password generate** [*minutes*]
5. **crypto pki server** *cs-label* **revoke** *certificate-serial-number*
6. **crypto pki server** *cs-label* **request pkcs10** {*url* | **terminal**} [**pem**]
7. **crypto pki server** *cs-label* **info crl**
8. **crypto pki server** *cs-label* **info requests**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**crypto pki server** *cs-label* **grant** {**all** | *req-id*} **Example:** `Router# crypto pki server mycs grant all`	Grants all or specific SCEP requests.
Step 3	**crypto pki server** *cs-label* **reject** {**all** | *req-id*} `Router# crypto pki server mycs reject all`	Rejects all or specific SCEP requests.
Step 4	**crypto pki server** *cs-label* **password generate** [*minutes*] **Example:** `Router# crypto pki server mycs password generate 75`	Generates a OTP for SCEP requests. • *minutes*—Length of time, in minutes, that the password is valid. Valid values range from 1 to 1440 minutes. The default is 60 minutes. **Note** Only one OTP is valid at a time; if a second OTP is generated, the previous OTP is no longer valid.

	Command or Action	Purpose
Step 5	**crypto pki server** *cs-label* **revoke** *certificate-serial-number* **Example:** `Router# crypto pki server mycs revoke 3`	Revokes a certificate on the basis of its serial number. • *certificate-serial-number*—One of the following options: – A string with a leading 0x, which is treated as a hexadecimal value – A string with a leading 0 and no x, which is treated as octal – All other strings, which are treated as decimal
Step 6	**crypto pki server** *cs-label* **request pkcs10** {*url* | **terminal**} [**pem**] **Example:** `Router# crypto pki server mycs request pkcs10 terminal pem`	Manually adds either a base64-encoded or PEM-formatted PKCS10 certificate enrollment request to the request database. After the certificate is granted, it will be displayed on the console terminal using base64 encoding; if the **pem** keyword is specified, PEM headers are also added to the certificate.
Step 7	**crypto pki server** *cs-label* **info crl** **Example:** `Router# crypto pki server mycs info crl`	Displays information regarding the status of the current CRL.
Step 8	**crypto pki server** *cs-label* **info requests** **Example:** `Router# crypto pki server mycs info requests`	Displays all outstanding certificate enrollment requests.

Removing Requests from the Enrollment Request Database

After the certificate server receives an enrollment request, the server can leave the request in a pending state, reject it, or grant it. The request stays in the enrollment request database for 1 week until the client polls the certificate server for the result of the request. If the client exits and never polls the certificate server, you can remove either individual requests or all requests from the database.

Use this task to remove requests from the database and allow the server to be returned to a clean slate with respect to the keys and transaction IDs. Also, you can use this task to help troubleshoot a SCEP client that may not be behaving properly.

SUMMARY STEPS

1. **enable**
2. **crypto pki server** *cs-label* **remove** {**all** | *req-id*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**crypto pki server** *cs-label* **remove** {**all** | *req-id*} **Example:** `Router# crypto pki server mycs remove 15`	Removes enrollment requests from the enrollment request database.

Deleting a Certificate Server

Users can delete a certificate server from the PKI configuration if they no longer want it on the configuration. Typically, a subordinate certificate server or an RA is being deleted. However, users may delete a root certificate server if they are moving it to another device via the archived RSA keys.

Perform this task to delete a certificate server from your PKI configuration.

Note When a certificate server is deleted, the user is queried with the option to delete the trustpoint and key.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no crypto pki server** *cs-label*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**no crypto pki server** *cs-label* **Example:** `Router (config)# no crypto pki server mycs`	Deletes a certificate server.

Verifying and Troubleshooting Certificate Server, Certificate, and CA Status

Use any of the following optional steps to verify the status of the certificate server, the certificate, or the CA.

SUMMARY STEPS

1. **enable**
2. **show crypto pki server**
3. **show crypto pki trustpoints** [**status** | *label* [**status**]]
4. **debug crypto pki server**
5. **dir** *filesystem***:**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto pki server** **Example:** `Router# show crypto pki server`	Displays the current state and configuration of the certificate server.
Step 3	**show crypto pki trustpoints** [**status** | *label* [**status**]] **Example:** `Router# show crypto pki trustpoints status`	Displays information about your certificate, the CA certificate, and any registration authority certificates.
Step 4	**debug crypto pki server** **Example:** `Router# debug crypto pki server`	Enables debugging for a crypto PKI certificate server. • This command can be used for monitoring the progress of an enrollment and for troubleshooting if the certificate server fails to respond or if the certificate server has trouble handling the request that has been configured.
Step 5	**dir** *filesystem***:** **Example:** `Router# dir slot0:`	Displays a list of files on a file system. • This command can be used to verify the certificate server autoarchived file if the **database url** command was entered to point to a local file system. You should be able to at least see "*cs-label*.ser" and "*cs-label*.crl" files in the database.

Configuration Examples for Using a Certificate Server

This section contains the following configuration examples:

- Removing Enrollment Requests from the Enrollment Request Database: Examples, page 1546
- Autoarchiving the Certificate Server Root Keys: Examples, page 1547
- Restoring a Certificate Server from Certificate Server Backup Files: Examples, page 1549
- RA Mode Certificate Server: Examples, page 1551
- Subordinate Certificate Server: Example, page 1553

Removing Enrollment Requests from the Enrollment Request Database: Examples

The following examples show both the enrollment requests that are currently in the enrollment request database and the result after one of the enrollment requests has been removed from the database.

Enrollment Request Currently in the Enrollment Request Database

The following example shows that the **crypto pki server info requests** command has been used to display the enrollment requests that are currently in the Enrollment Request Database:

```
Router# crypto pki server myserver info requests

Enrollment Request Database:

RA certificate requests:
ReqID    State      Fingerprint                                SubjectName
--------------------------------------------------------------------------

Router certificates requests:
ReqID    State      Fingerprint                                SubjectName
--------------------------------------------------------------------------
2        pending    1B07F3021DAAB0F19F35DA25D01D8567      hostname=host1.cisco.com
1        denied     5322459D2DC70B3F8EF3D03A795CF636      hostname=host2.cisco.com
```

crypto pki server remove Command Used to Remove One Enrollment Request

The following example shows that the **crypto pki server remove** command has been used to remove Enrollment Request 1:

```
Router# crypto pki server myserver remove 1
```

Enrollment Request Database After the Removal of One Enrollment Request

The following example shows the result of the removal of Enrollment Request 1 from the Enrollment Request Database:

```
Router# crypto pki server mycs info requests

Enrollment Request Database:

RA certificate requests:
ReqID    State     Fingerprint                          SubjectName
--------------------------------------------------------------------
```

```
Router certificates requests:
ReqID    State     Fingerprint                         SubjectName
--------------------------------------------------------------------
2        pending   1B07F3021DAAB0F19F35DA25D01D8567   hostname=host1.cisco.com
```

Autoarchiving the Certificate Server Root Keys: Examples

The following output configurations and examples show what you might see if the **database archive** command has not been configured (that is, configured using the default value); if the **database archive** command has been configured to set the CA certificate and CA key archive format as PEM, without configuring a password; and if the **database archive** command has been configured to set the CA certificate and CA key archive format as PKCS12, with a password configured. The last example is sample content of a PEM-formatted archive file.

database archive Command Not Configured

Note The default is PKCS12, and the prompt for the password appears after the **no shutdown** command has been issued.

```
Router (config)# crypto pki server myserver
Router (cs-server)# no shutdown

% Generating 1024 bit RSA keys ...[OK]
% Ready to generate the CA certificate.
%Some server settings cannot be changed after CA certificate generation.

Are you sure you want to do this? [yes/no]: y
% Exporting Certificate Server signing certificate and keys...
! Note the next two lines, which are asking for a password.
% Please enter a passphrase to protect the private key.
Password:
% Certificate Server enabled.
Router (cs-server)# end

Router# dir nvram:

Directory of nvram:/

  125  -rw-        1693               <no date>  startup-config
  126  ----           5               <no date>  private-config
    1  -rw-          32               <no date>  myserver.ser
    2  -rw-         214               <no date>  myserver.crl
! Note the next line, which indicates PKCS12 format.
    3  -rw-        1499               <no date>  myserver.p12
```

database archive Command and pem Keyword Configured

Note The prompt for the password appears after the **no shutdown** command has been issued.

```
Router (config)# crypto pki server myserver
Router (cs-server)# database archive pem
Router (cs-server)# no shutdown

% Generating 1024 bit RSA keys ...[OK]
% Ready to generate the CA certificate.
%Some server settings cannot be changed after CA certificate generation.
```

```
Are you sure you want to do this? [yes/no]: y
% Exporting Certificate Server signing certificate and keys...
!Note the next two lines, which are asking for a password.
% Please enter a passphrase to protect the private key.
Password:
% Certificate Server enabled.
Router (cs-server)# end

Router# dir nvram

Directory of nvram:/

  125  -rw-        1693            <no date>  startup-config
  126  ----           5            <no date>  private-config
    1  -rw-          32            <no date>  myserver.ser
    2  -rw-         214            <no date>  myserver.crl
! Note the next line showing that the format is PEM.
    3  -rw-        1705            <no date>  myserver.pem
```

database archive Command and pkcs12 Keyword (and Password) Configured

Note When the password is entered, it will be encrypted. However, it is recommended that you remove the password from the configuration after the archive has finished.

```
Router (config)# crypto pki server myserver
Router (cs-server)# database archive pkcs12 password cisco123
Router (cs-server)# no shutdown

% Generating 1024 bit RSA keys ...[OK]
% Ready to generate the CA certificate.
% Some server settings cannot be changed after CA certificate generation.

Are you sure you want to do this? [yes/no]: y
% Exporting Certificate Server signing certificate and keys...
! Note that you are not being prompted for a password.
% Certificate Server enabled.
Router (cs-server)# end
Router# dir nvram:
Directory of nvram:/

   125   -rw-         1693             <no date>   startup-config
   126   ----            5             <no date>   private-config
     1   -rw-           32             <no date>   myserver.ser
     2   -rw-          214             <no date>   myserver.crl
! Note that the next line indicates that the format is PKCS12.
     3   -rw-         1499             <no date>   myserver.p12
```

PEM-Formatted Archive

The following sample output shows that autoarchiving has been configured in PEM file format. The archive consists of the CA certificate and the CA private key. To restore the certificate server using the backup, you would have to import the PEM-formatted CA certificate and CA key individually.

Note In addition to the CA certificate and CA key archive files, you should also back up the serial file (.ser) and the CRL file (.crl) regularly. The serial file and the CRL file are both critical for CA operation if you need to restore your certificate server.

```
Router# more nvram:mycs.pem

-----BEGIN CERTIFICATE-----
MIIB9zCCAWCgAwIBAgIBATANBgkqhkiG9w0BAQQFADAPMQ0wCwYDVQQDEwRteWNz
MB4XDTA0MDgyNzAyMzI0NloXDTA3MDgyNzAyMzI0NlowDzENMAsGA1UEAxMEbXlj
czCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA1lZpKP4nGDJHgPkpYSkix7lD
nr23aMlZ9Kz5oo/qTBxeZ8mujpjYcZ0T8AZvoOiCuDnYM1796ZwpkMgjz1aZZbL+
BtuVvllsEOfhC+u/Ol/vxfGG5xpshoz/F5J3xdg5ZZuWWuIDAUYu9+QbI5feuG04
Z/BiPIb4AmGTP4B2MM0CAwEAAaNjMGEwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8B
Af8EBAMCAYYwHwYDVR0jBBgwFoAUKi/cuK6wkz+ZswVtb06vUJboEeEwHQYDVR0O
BBYEFCov3LiusJM/mbMFbW9Or1CW6BHhMA0GCSqGSIb3DQEBBAUAA4GBAKLOmoE2
4+NeOKEXMCXG1jcohK7O2HrkFfl/vpK0+q92PTnMUFhxLOqI8pWIq5CCgC7heace
OrTv2zcUAoH4rzx3Rc2USIxkDokWWQMLujsMm/SLIeHit0G5uj//GCcbgK20MAW6
ymf7+TmblSFljWzstoUXC2hLnsJIMq/KffaD
-----END CERTIFICATE-----

!The private key is protected by the password that is
configured in "database archive pem password pwd" or that
is entered when you are prompted for the password.
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,106CE91FFD0A075E

zyiFC8rKv8Cs+IKsQG2QpsVpvDBHqZqBSM4D528bvZv7jzr6WuHj8E6zO+6G8R/A
zjsfTALo+e+ZDg7KMzbryHARvjskbqFdOMLlVIYBhCeSElKsskWB6chOuyPHJInW
JwC5YzZdZwOqcyLBP/xOYXcvjzzNfPAXZzN12VR8vWDNq/kHT+3Lplc8hY++ABMI
M+C9FB3dpNZzu5O1BZCJg46bqbkulaCCmScIDaVt0zDFZwWTSufiemmNxZBG4xS8
t5t+FEhmSfv8DAmwg4f/KVRFTm10phUArcLxQO38Al0W5YHHORdACnuzVUvHgco7
VT4XUTjO7qMhmJgFNWy1pu49fbdS2NnOn5IoiyAq5lk1KUPrz/WABWiCvLMylGnZ
kyMCWoaMtgS/vdx74BBCj09yRZJnLMlIi6SDofjCNTDHfmFEVg4LsSWCd4lP9OP8
0MqhP1D5VIx6PbMNwkWW12lpBbCCdesFRGHjZD2dOu96kHD7ItErx34CC8W04aG4
b7DLktUu6WNV6M8g3CAqJiC0V8ATlp+kvdHZVkXovgND5IU0OJpsj0HhGzKAGpOY
KTGTUekUboISjVVkI6efp1vO6temVL3Txg3KGhzWMJGrq1snghE0KnV8tkddv/9N
d/t1l+we9mrccTq50WNDnkEi/cwHI/0PKXg+NDNH3k3QGpAprsqGQmMPdqc5ut0P
86i4cF9078QwWg4Tpay3uqNH1Zz6UN0tcarVVNmDupFESUxYw10qJrrEYVRadu74
rKAU4Ey4xkAftB2kuqvr21Av/L+jne4kkGIoZYdB+p/M98pQRgkYyg==
-----END RSA PRIVATE KEY-----
```

Restoring a Certificate Server from Certificate Server Backup Files: Examples

The following example shows that restoration is from a PKCS12 archive and that the database URL is NVRAM (the default).

```
Router# copy tftp://172.71.71.71/backup.ser nvram:mycs.ser
Destination filename [mycs.ser]?

32 bytes copied in 1.320 secs (24 bytes/sec)
Router# copy tftp://172.71.71.71/backup.crl nvram:mycs.crl
Destination filename [mycs.crl]?

214 bytes copied in 1.324 secs (162 bytes/sec)
Router# configure terminal
Router (config)# crypto pki import mycs pkcs12 tftp://172.71.71.71/backup.p12 cisco123
Source filename [backup.p12]?
CRYPTO_PKI: Imported PKCS12 file successfully.
Router (config)# crypto pki server mycs
! fill in any CS configuration here
Router (cs-server)# no shutdown
% Certificate Server enabled.
Router (cs-server)# end
```

```
Router# show crypto pki server
Certificate Server mycs:
    Status: enabled
    Server's current state: enabled
    Issuer name: CN=mycs
    CA cert fingerprint: 34885330 B13EAD45 196DA461 B43E813F
    Granting mode is: manual
    Last certificate issued serial number: 0x1
    CA certificate expiration timer: 01:49:13 GMT Aug 28 2007
    CRL NextUpdate timer: 01:49:16 GMT Sep 4 2004
    Current storage dir: nvram:
    Database Level: Minimum - no cert data written to storage
```

The following example shows that restoration is from a PEM archive and that the database URL is flash:

```
Router# copy tftp://172.71.71.71/backup.ser flash:mycs.ser
Destination filename [mycs.ser]?
32 bytes copied in 1.320 secs (24 bytes/sec)
Router# copy tftp://172.71.71.71/backup.crl flash:mycs.crl
Destination filename [mycs.crl]?
214 bytes copied in 1.324 secs (162 bytes/sec)
Router# configure terminal
! Because CA cert has Digital Signature usage, you need to import using the "usage-keys"
keyword
Router (config)# crypto ca import mycs pem usage-keys terminal cisco123
% Enter PEM-formatted CA certificate.
% End with a blank line or "quit" on a line by itself.
! Paste the CA cert from .pem archive.
-----BEGIN CERTIFICATE-----
MIIB9zCCAWCgAwIBAgIBATANBgkqhkiG9w0BAQQFADAPMQ0wCwYDVQQDEwRteWNz
MB4XDTA0MDkwMjIxMDI1NloXDTA3MDkwMjIxMDI1NlowDzENMAsGA1UEAxMEbXlj
czCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAuGnnDXJbpDDQwCuKGs5Zg2rc
K7ZJauSUotTmWYQvNx+ZmWrUs5/j9Ee5FV2YonirGBQ9mc6u163kNlrIPFck062L
GpahBhNmKDgod1o2PHTnRlZpEZNDIqU2D3hACgByxPjrY4vUnccV36ewLnQnYpp8
szEu7PYTJr5dU5ltAekCAwEAAaNjMGEwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8B
Af8EBAMCAYYwHwYDVR0jBBgwFoAUaEEQwYKCQ1dm9+wLYBKRTlzxaDIwHQYDVR0O
BBYEFGhBEMGCgkNXZvfsC2ASkU5c8WgyMA0GCSqGSIb3DQEBBAUAA4GBAHyhiv2C
mH+vswkBjRA1Fzzk8ttu9s5kwqG0dXp25QRUWsGlr9nsKPNdVKt3P7p0A/KochHe
eNiygiv+hDQ3FVnzsNv983le6O5jvAPxc17RO1BbfNhqvEWMsXdnjHOcUy7XerCo
+bdPcUf/eCiZueH/BEy/SZhD7yovzn2cdzBN
-----END CERTIFICATE-----

% Enter PEM-formatted encrypted private SIGNATURE key.
% End with "quit" on a line by itself.
! Paste the CA private key from .pem archive.
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,5053DC842B04612A

1CnlF5Pqvd0zp2NLZ7iosxzTy6nDeXPpNyJpxB5q+V29IuY8Apb6TlJCU7YrsEB/
nBTK7K76DCeGPlLpcuyEI171QmkQJ2gA0QhC0LrRo09WrINVH+b4So/y7nffZkVb
p2yDpZwqoJ8cmRH94Tie0YmzBtEh6ayOud11z53qbrsCnfSEwszt1xrW1MKrFZrk
/fTy6loHzGFzl3BDj4r5gBecExwcPp74ldHO+Ld4Nc9egG8BYkeBCsZZOQNVhXLN
I0tODOs6hP915zb6OrZFYv0NK6grTBO9D8hjNZ3U79jJzsSP7UNzIYHNTzRJiAyu
i56Oy/iHvkCSNUIK6zeIJQnW4bSoM1BqrbVPwHU6QaXUqlNzZ8SDtw7ZRZ/rHuiD
RTJMPbKquAzeuBss1132OaAUJRStjPXgyZTUbc+cWb6zATNws2yijPDTR6sRHoQL
47wHMr2Yj80VZGgkCSLAkL88ACz9TfUiVFhtf16xMC2yuFl+WRk1XfF5VtWe5Zer
3Fn1DcBmlF7O86XUkiSHP4EV0cI6n5ZMzVLx0XAUtdAl1gD94y1V+6p9PcQHLyQA
pGRmj5IlSFw90aLafgCTbRbmC0ChIqHy91UFa1ub0130+yu7LsLGRlPmJ9NE61JR
bjRhlUXItRYWY7C4M3m/0wz6fmVQNSumJM08RHq6lUB3olzIgGIZlZkoaESrLG0p
qq2AENFemCPF0uhyVS2humMHjWuRr+jedfc/IMl7sLEgAdqCVCfV3RZVEaNXBud1
4QjkuTrwaTcRXVFbtrVioT/puyVUlpA7+k7w+F5TZwUV08mwvUEqDw==
-----END RSA PRIVATE KEY-----
quit
```

```
% Enter PEM-formatted SIGNATURE certificate.
% End with a blank line or "quit" on a line by itself.
! Paste the CA cert from .pem archive again.
-----BEGIN CERTIFICATE-----
MIIB9zCCAWCgAwIBAgIBATANBgkqhkiG9w0BAQQFADAPMQ0wCwYDVQQDEwRteWNz
MB4XDTA0MDkwMjIxMDI1NloXDTA3MDkwMjIxMDI1NlowDzENMAsGA1UEAxMEbXlj
czCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAuGnnDXJbpDDQwCuKGs5Zg2rc
K7ZJauSUotTmWYQvNx+ZmWrUs5/j9Ee5FV2YonirGBQ9mc6u163kNlrIPFck062L
GpahBhNmKDgod1o2PHTnRlZpEZNDIqU2D3hACgByxPjrY4vUnccV36ewLnQnYpp8
szEu7PYTJr5dU51tAekCAwEAAaNjMGEwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8B
Af8EBAMCAYYwHwYDVR0jBBgwFoAUaEEQwYKCQ1dm9+wLYBKRTlzxaDIwHQYDVR0O
BBYEFGhBEMGCgkNXZvfsC2ASkU5c8WgyMA0GCSqGSIb3DQEBBAUAA4GBAHyhiv2C
mH+vswkBjRA1Fzzk8ttu9s5kwqG0dXp25QRUWsGlr9nsKPNdVKt3P7p0A/KochHe
eNiygiv+hDQ3FVnzsNv9831e6O5jvAPxc17RO1BbfNhqvEWMsXdnjHOcUy7XerCo
+bdPcUf/eCiZueH/BEy/SZhD7yovzn2cdzBN
-----END CERTIFICATE-----

% Enter PEM-formatted encrypted private ENCRYPTION key.
% End with "quit" on a line by itself.
! Because the CA cert only has Digital Signature usage, skip the encryption part.
quit
% PEM files import succeeded.
Router (config)# crypto pki server mycs
Router (cs-server)# database url flash:
! Fill in any CS configuration here.
Router (cs-server)# no shutdown
% Certificate Server enabled.
Router (cs-server)# end

Router # show crypto pki server

Certificate Server mycs:
    Status: enabled
    Server's current state: enabled
    Issuer name: CN=mycs
    CA cert fingerprint: F04C2B75 E0243FBC 19806219 B1D77412
    Granting mode is: manual
    Last certificate issued serial number: 0x2
    CA certificate expiration timer: 21:02:55 GMT Sep 2 2007
    CRL NextUpdate timer: 21:02:58 GMT Sep 9 2004
    Current storage dir: flash:
    Database Level: Minimum - no cert data written to storage
```

RA Mode Certificate Server: Examples

The following output is typical of what you might see after having configured an RA mode certificate server:

```
Router-ra (config)# crypto pki trustpoint myra
Router-ra (ca-trustpoint)# enrollment url http://10.3.0.17
! Include "cn=ioscs RA" or "ou=ioscs RA" in the subject-name.
Router-ra (ca-trustpoint)# subject-name cn=myra, ou=ioscs RA, o=cisco, c=us
Router-ra (ca-trustpoint)# exit
Router-ra (config)# crypto pki server myra
Router-ra (cs-server)# mode ra
Router-ra (cs-server)# no shutdown
% Generating 1024 bit RSA keys ...[OK]

Certificate has the following attributes:
Fingerprint MD5: 32661452 0DDA3CE5 8723B469 09AB9E85
Fingerprint SHA1: 9785BBCD 6C67D27C C950E8D0 718C7A14 C0FE9C38
% Do you accept this certificate? [yes/no]: yes
```

```
Trustpoint CA certificate accepted.
% Ready to request the CA certificate.
%Some server settings cannot be changed after the CA certificate has been requested.

Are you sure you want to do this? [yes/no]: yes
%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this
   password to the CA administrator in order to revoke your certificate.
   For security reasons your password will not be saved in the configuration.
   Please make a note of it.

Password:
Re-enter password:

% The subject name in the certificate will include: cn=myra, ou=ioscs RA, o=cisco, c=us
% The subject name in the certificate will include: Router-ra.cisco.com
% Include the router serial number in the subject name? [yes/no]: no
% Include an IP address in the subject name? [no]: no
Request certificate from CA? [yes/no]: yes
% Certificate request sent to Certificate Authority
% The certificate request fingerprint will be displayed.
% The 'show crypto pki certificate' command will also show the fingerprint.

% Enrollment in progress...
Router-ra (cs-server)#
Sep 15 22:32:40.197: CRYPTO_PKI:  Certificate Request Fingerprint MD5: 82B41A76 AF4EC87D
AAF093CD 07747D3A
Sep 15 22:32:40.201: CRYPTO_PKI:  Certificate Request Fingerprint SHA1: 897CDF40 C6563EAA
0FED05F7 0115FD3A 4FFC5231
Sep 15 22:34:00.366: %PKI-6-CERTRET: Certificate received from Certificate Authority
Router-ra (cs-server)#
Router-ra(cs-server)# end

Router-ra# show crypto pki server

Certificate Server myra:
    Status: enabled
    Issuer name: CN=myra
    CA cert fingerprint: 32661452 0DDA3CE5 8723B469 09AB9E85
    ! Note that the certificate server is running in RA mode
    Server configured in RA mode
    RA cert fingerprint: C65F5724 0E63B3CC BE7AE016 BE0D34FE
    Granting mode is: manual
    Current storage dir: nvram:
    Database Level: Minimum - no cert data written to storage
```

The following output shows the enrollment request database of the issuing certificate server after the RA has been enabled:

Note The RA certificate request is recognized by the issuing certificate server because "ou=ioscs RA" is listed in the subject name.

```
Router-ca# crypto pki server mycs info request
Enrollment Request Database:

Subordinate CA certificate requests:
ReqID  State      Fingerprint                      SubjectName
--------------------------------------------------------------

! The request is identified as RA certificate request.
```

```
RA certificate requests:
ReqID  State      Fingerprint                      SubjectName
--------------------------------------------------------------
12     pending    88F547A407FA0C90F97CDE8900A30CB0
hostname=Router-ra.cisco.com,cn=myra,ou=ioscs RA,o=cisco,c=us

Router certificates requests:
ReqID   State     Fingerprint                      SubjectName
--------------------------------------------------------------

! Issue the RA certificate.
Router-ca# crypto pki server mycs grant 12
```

The following output shows that the issuing certificate server is configured to issue a certificate automatically if the request comes from an RA:

```
Router-ca(config)# crypto pki server mycs
Router-ca (cs-server)# grant ra-auto
% This will cause all certificate requests already authorized by known RAs to be
automatically granted.

Are you sure you want to do this? [yes/no]: yes
Router-ca (cs-server)# end
Router-ca# show crypto pki server
Certificate Server mycs:
    Status: enabled
    Server's current state: enabled
    Issuer name: CN=mycs
    CA cert fingerprint: 32661452 0DDA3CE5 8723B469 09AB9E85
    ! Note that the certificate server will issue certificate for requests from the RA.
    Granting mode is: auto for RA-authorized requests, manual otherwise
    Last certificate issued serial number: 0x2
    CA certificate expiration timer: 22:29:37 GMT Sep 15 2007
    CRL NextUpdate timer: 22:29:39 GMT Sep 22 2004
    Current storage dir: nvram:
    Database Level: Minimum - no cert data written to storage
```

Subordinate Certificate Server: Example

The following configuration and output is typical of what you might see after configuring a subordinate certificate server:

```
Router (config)# crypto pki trustpoint sub
Router (ca-trustpoint)# enrollment url http://10.3.0.6
Router (ca-trustpoint)# exit

Router (config)# crypto pki server sub
Router (cs-server)# mode sub-cs
Router (ca-server)# no shutdown
%Some server settings cannot be changed after CA certificate generation.
% Please enter a passphrase to protect the private key % or type Return to exit Password:
Jan  6 22:32:22.698: CRYPTO_CS: enter FSM: input state initial, input signal no shut

Re-enter password:
% Generating 1024 bit RSA keys ...
Jan  6 22:32:30.302: CRYPTO_CS: starting enabling checks
Jan  6 22:32:30.306: CRYPTO_CS: key 'sub' does not exist; generated automatically[OK]

Jan  6 22:32:39.810: %SSH-5-ENABLED: SSH 1.99 has been enabled
Certificate has the following attributes:
     Fingerprint MD5: 328ACC02 52B25DB8 22F8F104 B6055B5B
```

```
     Fingerprint SHA1: 02FD799D DD40C7A8 61DC53AB 1E89A3EA 2A729EE2

% Do you accept this certificate? [yes/no]:
Jan  6 22:32:44.830: CRYPTO_CS: nvram filesystem
Jan  6 22:32:44.922: CRYPTO_CS: serial number 0x1 written.
Jan  6 22:32:46.798: CRYPTO_CS: created a new serial file.
Jan  6 22:32:46.798: CRYPTO_CS: authenticating the CA 'sub'y
Trustpoint CA certificate accepted.%

% Certificate request sent to Certificate Authority

% Enrollment in progress...
Router (cs-server)#
Jan  6 22:33:30.562: CRYPTO_CS: Publishing 213 bytes to crl file nvram:sub.crl
Jan  6 22:33:32.450: CRYPTO_CS: enrolling the server's trustpoint 'sub'
Jan  6 22:33:32.454: CRYPTO_CS: exit FSM: new state check failed
Jan  6 22:33:32.454: CRYPTO_CS: cs config has been locked
Jan  6 22:33:33.118: CRYPTO_PKI:  Certificate Request Fingerprint MD5: CED89E5F 53B9C60E
> AA123413 CDDAD964
Jan  6 22:33:33.118: CRYPTO_PKI:  Certificate Request Fingerprint SHA1: 70787C76 ACD7E67F
7D2C8B23 98CB10E7 718E84B1
% Exporting Certificate Server signing certificate and keys...

Jan  6 22:34:53.839: %PKI-6-CERTRET: Certificate received from Certificate Authority
Jan  6 22:34:53.843: CRYPTO_CS: enter FSM: input state check failed, input signal cert
configured
Jan 6 22:34:53.843: CRYPTO_CS: starting enabling checks
Jan  6 22:34:53.843: CRYPTO_CS: nvram filesystem
Jan 6 22:34:53.883: CRYPTO_CS: found existing serial file.
Jan  6 22:34:53.907: CRYPTO_CS: old router cert flag 0x4
Jan  6 22:34:53.907: CRYPTO_CS: new router cert flag 0x44
Jan  6 22:34:56.511: CRYPTO_CS: DB version
Jan  6 22:34:56.511: CRYPTO_CS: last issued serial number is 0x1
Jan  6 22:34:56.551: CRYPTO_CS: CRL file sub.crl exists.
Jan  6 22:34:56.551: CRYPTO_CS: Read 213 bytes from crl file sub.crl.
Jan  6 22:34:56.603: CRYPTO_CS: SCEP server started
Jan  6 22:34:56.603: CRYPTO_CS: exit FSM: new state enabled
Jan  6 22:34:56.603: CRYPTO_CS: cs config has been locked
Jan 6 22:35:02.359: CRYPTO_CS: enter FSM: input state enabled, input signal time set
Jan  6 22:35:02.359: CRYPTO_CS: exit FSM: new state enabled
Jan  6 22:35:02.359: CRYPTO_CS: cs config has been locked
```

Root Certificate Server Differentiation: Example

When issuing certificates, the root certificate server (or parent subordinate certificate server) will differentiate the certificate request from “Sub CA,” “RA,” and peer requests, as shown in the following sample output:

```
Router# crypto pki server server1 info req

Enrollment Request Database:
RA certificate requests:
ReqID     State        Fingerprint                                SubjectName
--------------------------------------------------------------------------------
Subordinate CS certificate requests:
ReqID     State        Fingerprint                                SubjectName
--------------------------------------------------------------------------------
1     pending        CB9977AD8A73B146D3221749999B0F66 hostname=bubinga-subcs.cisco.com
RA certificate requests:
ReqID     State        Fingerprint                                 SubjectName
--------------------------------------------------------------------------------
Router certificate requests:
```

```
ReqID     State        Fingerprint                                SubjectName
--------------------------------------------------------------------------------
```

Show Output for a Subordinate Certificate Server: Example

The following **show crypto pki server** command output indicates that a subordinate certificate server has been configured:

```
Router# show crypto pki server

Certificate Server sub:
  Status: enabled
  Server's configuration is locked  (enter "shut" to unlock it)
  Issuer name: CN=sub
  CA cert fingerprint: 11B586EE 3B354F33 14A25DDD 7BD39187
  Server configured in subordinate server mode
  Upper CA cert fingerprint: 328ACC02 52B25DB8 22F8F104 B6055B5B
  Granting mode is: manual
  Last certificate issued serial number: 0x1
  CA certificate expiration timer: 22:33:44 GMT Jan 6 2006
  CRL NextUpdate timer: 22:33:29 GMT Jan 13 2005
  Current storage dir: nvram:
  Database Level: Minimum - no cert data written to storage
```

Where to Go Next

After the certificate server is successfully running, you can either begin enrolling clients via manual mechanisms (as explained in the module “Configuring Certificate Enrollment for a PKI”) or begin configuring SDP, which is a web-based enrollment interface, (as explained in the module “Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI.”)

Additional References

The following sections provide references related to Cisco IOS certificate server.

Related Documents

Related Topic	Document Title
Manual certificate enrollment	“Configuring Certificate Enrollment for a PKI” module
Web-based certificate enrollment	“Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI” module
RSA Keys in PEM formatted files	“Deploying RSA Keys Within a PKI” module

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Feature Information for the Cisco IOS Certificate Server

Table 63 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for a specific command was introduced, see the command reference documentation.

For information on a feature in this technology that is not documented here, see the Implementing and Managing PKI Features Roadmap.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Table 63 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 63 Feature Information for the Cisco IOS Certificate Server

Feature Name	Software Releases	Feature Configuration Information
Cisco IOS Certificate Server	12.3(8)T	This feature introduces support for the Cisco IOS CS, which offers users a CA that is directly integrated with Cisco IOS software to more easily deploy basic PKI networks. The following sections provide information about this feature: • Information About Cisco IOS Certificate Servers • How to Set Up and Deploy a Cisco IOS Certificate Server
The Certificate Server Registration Authority (RA) Mode enhancement	12.3(7)T	A certificate server can be configured to run in RA mode. The following section provides information about this feature: • Configuring a Certificate Server to Run in RA Mode by Configuring the RA Mode Certificate Server The following commands were introduced by this feature: **grant ra-auto**, **lifetime enrollment-requests**
The Certificate Server Auto Archive enhancement[1]	12.3(11)T	This enhancement enables the CA certificate and CA key to be backed up automatically just once after they are generated by the certificate server. As a result, it is not necessary to generate an exportable CA key if CA backup is desirable. The following sections provide information about this feature: • Certificate Enrollment Using a Certificate Server • Configuring Certificate Server Functionality The following commands were introduced by this feature: **crypto pki server remote**, **database archive**

Table 63 Feature Information for the Cisco IOS Certificate Server (continued)

Feature Name	Software Releases	Feature Configuration Information
PKI Status[1]	12.3(11)T	This enhancement provides a quick snapshot of current trustpoint status. The following section provides information about this enhancement: • Maintaining, Verifying, and Troubleshooting the Certificate Server, Certificates, and the CA The following command was modified by this enhancement: **show crypto pki trustpoints**
Subordinate Certificate Server[1]	12.3(14)T	This enhancement allows you to configure a subordinate certificate server to grant all or certain SCEP or manual certificate requests. The following section provides information about this enhancement: • Configuring a Subordinate Certificate Server The following command was introduced by this enhancement: **mode sub-cs**

1. This is a minor enhancement. Minor enhancements are not typically listed in Feature Navigator.

Storing PKI Credentials

This module explains how to store public key infrastructure (PKI) credentials, such as Rivest, Shamir, and Adelman (RSA) keys and certificates, in a location other than the default location on the router, which is NVRAM.

For example, selected Cisco platforms now support Smart card technology in a USB key form factor (also known as an Aladdin USB eToken key). eTokens provide secure configuration distribution and allow users to store Virtual Private Network (VPN) credentials for deployment.

Module History

This module was first published on May 2, 2005, and last updated on May 2, 2005.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all features. To find information about feature support and configuration, use the "Feature Information for Storing PKI Credentials" section on page 1573.

Contents

- Prerequisites for Storing PKI Credentials, page 1559
- Restrictions for Storing PKI Credentials, page 1560
- Information About Storing PKI Credentials, page 1560
- How to Configure PKI Storage, page 1561
- Configuration Examples for PKI Storage, page 1571
- Additional References, page 1573
- Feature Information for Storing PKI Credentials, page 1573

Prerequisites for Storing PKI Credentials

Before you can use an eToken, you should have the following system requirements:

- A Cisco 871 router, Cisco 1800 series, Cisco 2800 series, or a Cisco 3800 series router
- At least a Cisco IOS Release 12.3(14)T image running on any of the supported platforms
- A Cisco supported USB eToken
- A k9 image

Restrictions for Storing PKI Credentials

When using an eToken to store PKI data, the following restrictions are applicable:

- USB eToken support requires a 3DES (k9) Cisco IOS software image, which provides secure file storage.
- You cannot boot an image from an eToken. (However, you can boot a configuration an eToken.)
- USB hubs are currently not supported. Thus, the number of supported devices is limited to the number of available USB ports.

Information About Storing PKI Credentials

To use a secure eToken on your router, you should understand the following concepts:

- How a USB eToken Works, page 1560
- Benefits of USB eTokens, page 1561

How a USB eToken Works

A Smart card is a small plastic card, containing a microprocessor and memory that allows you to store and process data. An eToken is a Smart card with a USB interface. The eToken can securely store any type of file within its available storage space (32KB). Configuration files that are stored on the eToken can be encrypted and accessed only via a user PIN. The router will not load the configuration file unless the proper PIN has been configured for secure deployment of router configuration files.

After you plug the eToken into the router, you must log into the eToken; thereafter, you can change default settings, such as the user PIN (default: 1234567890) and the allowed number of failed login attempts before future logins are refused (default: 15 attempts). For more information on accessing and configuring the eToken, see the section "Accessing and Setting Up the eToken."

After you have successfully logged into the eToken, you can copy files from the router on to the eToken via the **copy** command. By default, after the eToken is removed from the router, all associated RSA keys are removed; IP Security (IPSec) tunnels are not torn down until the next Internet Key Exchange (IKE) negotiation period. (To change the default behavior and configure a specified length of time before the IPSec tunnels are torn down, issue the **crypto pki token removal timeout** command.)

For more information about the eToken by Aladdin Knowledge Systems, see the Aladdin website at http://www.aladdin.com/etoken/cisco/.

Table 64 highlights the capabilities of the USB eToken.

Table 64 Functionality Highlights for USB eTokens

Function	USB eToken
Accessibility	Used to securely store and transfer digital certificates, preshared keys, and router configurations from the eToken to the router.
Storage Size	32KB
File Types	• Typically used to store digital certificates, preshared keys, and router configurations for IPSec virtual private networks (VPNs). • eTokens cannot store Cisco IOS images.

Table 64 Functionality Highlights for USB eTokens (continued)

Function	USB eToken
Security	• Files can be encrypted and accessed only with a user PIN. • Files can also be stored in a nonsecure format.
Boot Configurations	• The router can use the configuration stored in the eToken during boot time. • The router can use the secondary configuration stored in the eToken during boot time. (A secondary configuration allows users to load their IPSec configuration.)

Benefits of USB eTokens

USB eToken support on a Cisco router provides the following application benefits:

Removable Credentials: Provide or Store VPN Credentials on an External Device for Deployment

An Aladdin eToken can use SmartCard technology to store a digital certificate and configuration for IPSec VPN deployment. This ability enhances the capability of the router to generate RSA public keys to authenticate at least one IPSec tunnel. (Because a router can initiate multiple IPSec tunnels, the eToken can contain several certificates, as appropriate.)

Storing VPN credentials on an external device reduces the threat of compromising secure data.

PIN Configuration for Secure File Deployment

An Aladdin eToken can store a configuration file that can be used for enabling encryption on the router via a user-configured PIN. (That is, no digital certificates, preshared keys, or VPNs are used.)

Touchless or Low Touch Configuration

The eToken can provide remote software configuration and provisioning with little or no human interaction. Configuration is set up as an automated process. That is, the eToken can store a bootstrap configuration that the router can use to boot from after the eToken has been inserted into the router. The bootstrap configuration connects the router to a TFTP server, which contains a configuration that completely configures the router.

How to Configure PKI Storage

This section contains the following configuration tasks:

- Setting Up and Using USB eTokens on Cisco Routers, page 1561
- Troubleshooting USB eTokens, page 1566

Setting Up and Using USB eTokens on Cisco Routers

This section contains the following procedures that allow you to configure a router to support USB eTokens:

- Storing the Configuration on an External USB eToken, page 1562
- Accessing and Setting Up the eToken, page 1562

- Setting Administrative Functions on the eToken, page 1564

Storing the Configuration on an External USB eToken

Perform this task to store the configuration file in an eToken.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **boot config usbtoken**[**0**-**9**]:*filename*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**boot config usbtoken**[**0-9**]:*filename* **Example:** `Router(config)# boot config usbtoken0:`	Specifies that the startup configuration file is stored in a secure eToken.

Accessing and Setting Up the eToken

After you have inserted the eToken into the Cisco router, you must log into the eToken as shown in the following task:

- Logging Into the eToken, page 1563 (required)

After you have logged into the eToken, you can perform administrative tasks, such as changing the user PIN and copying files from the router to the eToken, as shown in the following task:

- Setting Administrative Functions on the eToken, page 1564 (optional)

Use of RSA Keys with an eToken

- RSA keys are loaded after the eToken is successfully logged into the router.
- By default, newly generated RSA keys are stored on the most recently inserted eToken. Regenerated keys should be stored in the same location at which the original RSA key was generated.

Logging Into the eToken

Perform this task to log into an eToken automatically or manually.

Automatic Login

Automatic login allows the router to completely come back up without any user or operator intervention. The PIN is stored in the private configuration, so it is not visible in the startup or running configuration.

Note A hand-generated startup configuration can contain the automatic login command for deployment purposes, but the **copy system:running-config nvram: startup-config** command must be issued to put the hand-generated configuration in the private configuration.

Manual Login

Unlike automatic login, manual login requires that the user know the actual token PIN. However, if the user also has physical access to the eToken, he or she can use Aladdin's Windows-based utilities to copy the RSA keys and secondary configuration files from the eToken.

Manual login can be used when storing a PIN on the router is not desirable. Manual login may also be suitable for some initial deployment or hardware replacement scenarios for which the router is obtained from the local supplier or drop-shipped to the remote site. Manual login can be executed with or without privileges, and it will make files and RSA keys on the eToken available to the Cisco IOS software. If a secondary configuration file is configured, it will be executed only with the privileges of the user who is performing the login. Thus, if you want to use manual login and set up the secondary configuration on the eToken to perform anything useful, you need to enable privileges.

Manual login can also be used in recovery scenarios for which the router configuration has been lost. If the scenario contains a remote site that normally connects to the core network with a VPN, the loss of the configuration and RSA keys requires out-of-band services that the eToken can provide. The eToken can contain a boot configuration, a secondary configuration, or both, and RSA keys to authenticate the connection.

SUMMARY STEPS

1. **enable**
2. **crypto pki token** *token-name* [**admin**] **login** [*pin*]

   or

   configure terminal
3. **crypto pki token** *token-name* **user-pin** [*pin*]
4. **exit**
5. **show usbtoken**[**0-9**]**:***filename*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**crypto pki token** *token-name* [**admin**] **login** [*pin*] **Example:** `Router# crypto pki token usbtoken0 admin login 5678` or **configure terminal** **Example:** `Router# configure terminal`	Manually logs into the eToken. You must specify the **admin** keyword if later you want to change the user PIN. or Puts the router in global configuration mode, which allows you to configure automatic eToken login.
Step 3	**crypto pki token** *token-name* **user-pin** [*pin*] **Example:** `Router(config)# crypto pki token usbtoken0 user-pin 1234`	(Optional) Creates a PIN that automatically allows the router to log into the USB eToken at router startup. **Note** Do not issue this command if you have already set up manual login.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 5	**show usbtoken**[**0-9**]**:***filename* **Example:** `Router# show usbtoken0:`	(Optional) Verifies whether the USB eToken has been logged onto the router.

Setting Administrative Functions on the eToken

Perform this task to change default settings, such as the user PIN and the maximum number of failed on the eToken.

SUMMARY STEPS

1. **enable**
2. **crypto pki token** *token-name* [**admin**] **change-pin** [*pin*]
3. **configure terminal**
4. **crypto pki token** {*token-name* | **default**} **removal timeout** [*seconds*]
5. **crypto pki token** {*token-name* | **default**} **max-retries** [*number*]
6. **exit**

7. **copy usbflash**[**0**-**9**]**:***filename destination-url*
8. **show usbtoken**[**0**-**9**]**:***filename*
9. **crypto pki token** *token-name* **logout**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**crypto pki token** *token-name* [**admin**] **change-pin** [*pin*] **Example:** `Router# crypto pki token usbtoken0 admin change-pin`	(Optional) Changes the user PIN number on the USB eToken. • If the PIN is not changed, the default PIN—1234567890—will be used. **Note** After the PIN has been changed, you must reset the login failure count to zero (via the **crypto pki token max-retries** command). The maximum number of allowable login failures is set (by default) to 15.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**crypto pki token** {*token-name* | **default**} **removal timeout** [*seconds*] **Example:** `Router(config)# crypto pki token usbtoken0 removal timeout 60`	(Optional) Sets the time interval, in seconds, that the router will wait before removing the RSA keys that are stored in the eToken after the eToken has been removed from the router. **Note** If this command is not issued, all RSA keys and IPSec tunnels associated with the eToken are torn down immediately after the eToken is removed from the router.
Step 5	**crypto pki token** {*token-name* | **default**} **max-retries** [*number*] **Example:** `Router(config)# crypto pki token usbtoken0 max-retries 20`	(Optional) Sets the maximum number of consecutive failed login attempts allowed before access to the eToken is denied. • By default, the value is set at 15.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 7	**copy usbflash**[**0**-**9**]**:***filename destination-url* **Example:** `Router# copy usbflash0:`	Copies files from the router to the eToken. • *destination-url*—See the **copy** command page documentation for a list of supported options.

	Command or Action	Purpose
Step 8	**show usbtoken**[**0-9**]:*filename* **Example:** `Router# show usbtoken`	(Optional) Displays information about the USB eToken. You can use this command to verify whether the USB eToken has been logged onto the router.
Step 9	**crypto pki token** *token-name* **logout** **Example:** `Router# crypto pki token usbtoken0 logout`	Logs the router out of the USB eToken. **Note** If you want to save any data to the USB eToken, you must log back into the eToken.

Troubleshooting USB eTokens

This section contains descriptions of the following Cisco IOS commands that can be used to help troubleshoot possible problems that may arise while using a USB eToken:

- The show file systems Command, page 1566
- The show usb device Command, page 1567
- The show usb controllers Command, page 1568
- The dir Command, page 1570

The show file systems Command

Use the **show file systems** command to determine whether the router recognizes that there is a USB module plugged into a USB port. The USB module should appear on the list of file systems. If the module does not appear on the list, it can indicate any of the following problems:

- A connection problem with the USB module.
- The Cisco IOS image running on the router does not support a USB module.
- A hardware problem with the USB module itself.

SUMMARY STEPS

1. **show file systems**

DETAILED STEPS

Step 1 Sample output from the **show file systems** command showing a USB eToken appears below. The USB module listing appears in the last line of the examples.

```
Router# show file systems

File Systems:

     Size(b)     Free(b)      Type  Flags  Prefixes
           -           -    opaque     rw   archive:
           -           -    opaque     rw   system:
           -           -    opaque     rw   null:
           -           -   network     rw   tftp:
*  129880064    69414912      disk     rw   flash:#
```

```
   491512      486395      nvram     rw   nvram:
        -           -     opaque     wo   syslog:
        -           -     opaque     rw   xmodem:
        -           -     opaque     rw   ymodem:
        -           -    network     rw   rcp:
        -           -    network     rw   pram:
        -           -    network     rw   ftp:
        -           -    network     rw   http:
        -           -    network     rw   scp:
        -           -    network     rw   https:
        -           -     opaque     ro   cns:
 63158272    33037312   usbflash     rw   usbflash0:
    32768         858   usbtoken     rw   usbtoken1:
```

The show usb device Command

Use the **show usb device** command to determine if a USB eToken is supported by Cisco.

SUMMARY STEPS

1. **show usb device**

DETAILED STEPS

Step 1 The following sample output for the **show usb device** command indicates whether or not the module is supported is bold in the sample output below:

```
Router# show usb device

Host Controller:1
Address:0x11
Device Configured:YES
Device Supported:YES
Description:eToken Pro 4254
Manufacturer:AKS
Version:1.0
Serial Number:
Device Handle:0x1010000
USB Version Compliance:1.0
Class Code:0xFF
Subclass Code:0x0
Protocol:0x0
Vendor ID:0x529
Product ID:0x514
Max. Packet Size of Endpoint Zero:8
Number of Configurations:1
Speed:Low
Selected Configuration:1
Selected Interface:0

Configuration:
    Number:1
    Number of Interfaces:1
    Description:
    Attributes:None
    Max Power:60 mA

    Interface:
        Number:0
```

```
        Description:
        Class Code:255
        Subclass:0
        Protocol:0
        Number of Endpoints:0
```

The show usb controllers Command

Use the **show usb controllers** command to determine if there is a hardware problem with a USB flash module. If the **show usb controllers** command displays an error, it indicates a hardware problem in the USB module.

You can also use the **show usb controllers** command to verify that copy operations onto a USB flash module are occurring successfully. Issuing the **show usb controllers** command after performing a file copy should display successful data transfers.

SUMMARY STEPS

1. **show usb controllers**

DETAILED STEPS

Step 1 The following sample output for the **show usb controllers** command displays a working USB flash module:

```
Router# show usb controllers

Name:1362HCD
Controller ID:1
Controller Specific Information:
    Revision:0x11
    Control:0x80
    Command Status:0x0
    Hardware Interrupt Status:0x24
    Hardware Interrupt Enable:0x80000040
    Hardware Interrupt Disable:0x80000040
    Frame Interval:0x27782EDF
    Frame Remaining:0x13C1
    Frame Number:0xDA4C
    LSThreshold:0x628
    RhDescriptorA:0x19000202
    RhDescriptorB:0x0
    RhStatus:0x0
    RhPort1Status:0x100103
    RhPort2Status:0x100303
    Hardware Configuration:0x3029
    DMA Configuration:0x0
    Transfer Counter:0x1
    Interrupt:0x9
    Interrupt Enable:0x196
    Chip ID:0x3630
    Buffer Status:0x0
    Direct Address Length:0x80A00
    ATL Buffer Size:0x600
    ATL Buffer Port:0x0
    ATL Block Size:0x100
    ATL PTD Skip Map:0xFFFFFFFF
    ATL PTD Last:0x20
```

```
    ATL Current Active PTD:0x0
    ATL Threshold Count:0x1
    ATL Threshold Timeout:0xFF

Int Level:1
Transfer Completion Codes:
         Success              :920            CRC              :0
         Bit Stuff            :0              Stall            :0
         No Response          :0              Overrun          :0
         Underrun             :0              Other            :0
         Buffer Overrun       :0              Buffer Underrun :0
Transfer Errors:
         Canceled Transfers   :2              Control Timeout :0
Transfer Failures:
         Interrupt Transfer   :0              Bulk Transfer   :0
         Isochronous Transfer :0              Control Transfer:0
Transfer Successes:
         Interrupt Transfer   :0              Bulk Transfer   :26
         Isochronous Transfer :0              Control Transfer:894

USBD Failures:
         Enumeration Failures :0              No Class Driver Found:0
         Power Budget Exceeded:0

USB MSCD SCSI Class Driver Counters:
         Good Status Failures :3              Command Fail    :0
         Good Status Timed out:0              Device not Found:0
         Device Never Opened  :0              Drive Init Fail :0
         Illegal App Handle   :0              Bad API Command :0
         Invalid Unit Number  :0              Invalid Argument:0
         Application Overflow :0              Device in use   :0
         Control Pipe Stall   :0              Malloc Error    :0
         Device Stalled       :0              Bad Command Code:0
         Device Detached      :0              Unknown Error   :0
         Invalid Logic Unit Num:0

USB Aladdin Token Driver Counters:
         Token Inserted       :1              Token Removed   :0
         Send Insert Msg Fail :0              Response Txns   :434
         Dev Entry Add Fail   :0              Request Txns    :434
         Dev Entry Remove Fail:0              Request Txn Fail:0
         Response Txn Fail    :0              Command Txn Fail:0
         Txn Invalid Dev Handle:0

USB Flash File System Counters:
         Flash Disconnected   :0              Flash Connected :1
         Flash Device Fail    :0              Flash Ok        :1
         Flash startstop Fail :0              Flash FS Fail   :0

USB Secure Token File System Counters:
         Token Inserted       :1              Token Detached  :0
         Token FS success     :1              Token FS Fail   :0
         Token Max Inserted   :0              Create Talker Failures:0
         Token Event          :0              Destroy Talker Failures:0
         Watched Boolean Create Failures:0
```

The dir Command

Use the **dir** command with the **usbflash**[**0**-**9**]**:** keyword to display all files, directories, and their permission strings on the USB eToken.

SUMMARY STEPS

1. **dir**

DETAILED STEPS

Step 1 The following sample output displays directory information for the USB eToken:

```
Router# dir usbtoken1:

Directory of usbtoken1:/

    2  d---           64  Dec 22 2032 05:23:40 +00:00  1000
    5  d---         4096  Dec 22 2032 05:23:40 +00:00  1001
    8  d---            0  Dec 22 2032 05:23:40 +00:00  1002
   10  d---          512  Dec 22 2032 05:23:42 +00:00  1003
   12  d---            0  Dec 22 2032 05:23:42 +00:00  5000
   13  d---            0  Dec 22 2032 05:23:42 +00:00  6000
   14  d---            0  Dec 22 2032 05:23:42 +00:00  7000
   15  ----          940  Jun 27 1992 12:50:42 +00:00  mystartup-config
   16  ----         1423  Jun 27 1992 12:51:14 +00:00  myrunning-config

32768 bytes total (858 bytes free)
```

The following sample output displays directory information for all devices the router is aware of:

```
Router# dir all-filesystems

Directory of archive:/

No files in directory

No space information available
Directory of system:/

    2  drwx           0                    <no date>  its
  115  dr-x           0                    <no date>  lib
  144  dr-x           0                    <no date>  memory
    1  -rw-        1906                    <no date>  running-config
  114  dr-x           0                    <no date>  vfiles

No space information available
Directory of flash:/

    1  -rw-    30125020  Dec 22 2032 03:06:04 +00:00  c3825-entservicesk9-mz.123-14.T

129880064 bytes total (99753984 bytes free)
Directory of nvram:/

  476  -rw-        1947                    <no date>  startup-config
  477  ----          46                    <no date>  private-config
  478  -rw-        1947                    <no date>  underlying-config
    1  -rw-           0                    <no date>  ifIndex-table
    2  ----           4                    <no date>  rf_cold_starts
    3  ----          14                    <no date>  persistent-data
```

```
491512 bytes total (486395 bytes free)
Directory of usbflash0:/

    1  -rw-    30125020  Dec 22 2032 05:31:32 +00:00  c3825-entservicesk9-mz.123-14.T

63158272 bytes total (33033216 bytes free)
Directory of usbtoken1:/

    2  d---          64  Dec 22 2032 05:23:40 +00:00  1000
    5  d---        4096  Dec 22 2032 05:23:40 +00:00  1001
    8  d---           0  Dec 22 2032 05:23:40 +00:00  1002
   10  d---         512  Dec 22 2032 05:23:42 +00:00  1003
   12  d---           0  Dec 22 2032 05:23:42 +00:00  5000
   13  d---           0  Dec 22 2032 05:23:42 +00:00  6000
   14  d---           0  Dec 22 2032 05:23:42 +00:00  7000
   15  ----         940  Jun 27 1992 12:50:42 +00:00  mystartup-config
   16  ----        1423  Jun 27 1992 12:51:14 +00:00  myrunning-config

32768 bytes total (858 bytes free)
```

Configuration Examples for PKI Storage

This section contains the following configuration example:

- Logging Into an eToken and Saving RSA Keys to the eToken: Example, page 1571

Logging Into an eToken and Saving RSA Keys to the eToken: Example

The following configuration example shows to how log into the eToken, generate RSA keys, and store the RSA keys onto the eToken:

```
! Configure the router to automatically log into the eToken
configure terminal
 crypto pki token default user-pin 0 1234567890
! Generate RSA keys and enroll certificates with the CA.
crypto pki trustpoint IOSCA
 enrollment url http://10.23.2.2
 exit
crypto ca authenticate IOSCA
Certificate has the following attributes:
       Fingerprint MD5:23272BD4 37E3D9A4 236F7E1A F534444E
      Fingerprint SHA1:D1B4D9F8 D603249A 793B3CAF 8342E1FE 3934EB7A

% Do you accept this certificate? [yes/no]:yes
Trustpoint CA certificate accepted.
crypto pki enroll
crypto pki enroll IOSCA
%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this
   password to the CA Administrator in order to revoke your certificate.
   For security reasons your password will not be saved in the configuration.
   Please make a note of it.

Password:
Re-enter password:
```

```
% The subject name in the certificate will include:c2851-27.cisco.com
% Include the router serial number in the subject name? [yes/no]:no
% Include an IP address in the subject name? [no]:no
Request certificate from CA? [yes/no]:yes
% Certificate request sent to Certificate Authority
% The 'show crypto ca certificate IOSCA verbose' command will show the fingerprint.


*Jan 13 06:47:19.413:CRYPTO_PKI: Certificate Request Fingerprint MD5:E6DDAB1B
 0E30EFE6 54529D8A DA787DBA
*Jan 13 06:47:19.413:CRYPTO_PKI: Certificate Request Fingerprint SHA1:3B0F33B
7 57C02A10 3935042B C4B6CD3D 61039251
*Jan 13 06:47:21.021:%PKI-6-CERTRET:Certificate received from Certificate Authority
! Issue the write memory command, which will automatically save the RSA keys to the eToken
! instead of private NVRAM.
Router# write memory
Building configuration...
[OK]

*Jan 13 06:47:29.481:%CRYPTO-6-TOKENSTOREKEY:Key c2851-27.cisco.com stored on
Cryptographic Token eToken Successfully
```

The following sample output from the **show crypto key mypubkey rsa** command displays stored credentials after they are successfully loaded from the eToken. Credentials that are stored on the eToken are in the protected area. When storing the credentials on the eToken, the files are stored in a directory called /keystore. However, the key files are hidden from the command-line interface (CLI).

```
Router# show crypto key mypubkey rsa

% Key pair was generated at:06:37:26 UTC Jan 13 2005
Key name:c2851-27.cisco.com
 Usage:General Purpose Key
 Key is not exportable.
 Key Data:
  305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00E3C644 43AA7DDD
  732E0F4E 3CA0CDAB 387ABF05 EB8F22F2 2431F1AE 5D51FEE3 FCDEA934 7FBD3603
  7C977854 B8E999BF 7FC93021 7F46ABF8 A4BA2ED6 172D3D09 B5020301 0001
% Key pair was generated at:06:37:27 UTC Jan 13 2005
Key name:c2851-27.cisco.com.server
 Usage:Encryption Key
 Key is not exportable.
 Key Data:
  307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00DD96AE 4BF912EB
  2C261922 4784EF98 2E70E837 774B3778 7F7AEB2D 87F5669B BF5DDFBC F0D521A5
  56AB8FDC 9911968E DE347FB0 A514A856 B30EAFF4 D1F453E1 003CFE65 0CCC6DC7
  21FBE3AC 2F8DEA16 126754BC 1433DEF9 53266D33 E7338C95 BB020301 0001
```

Additional References

The following sections provide references related to PKI storage support.

Related Documents

Related Topic	Document Title
Connecting the USB modules to the router	*Cisco Access Router USB Flash Module and USB eToken Hardware Installation Guide*
eToken and USB flash data sheet	*USB eToken and USB Flash Features Support*
RSA keys	*Deploying RSA Keys Within a PKI*
File management (loading, copying, and rebooting files)	The section "File Management" in the *Cisco IOS Configuration Fundamentals and Network Management Configuration Guide*, Release 12.3

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Feature Information for Storing PKI Credentials

Table 65 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1) or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for a specific command was introduced, see the command reference documentation.

For information on a feature in this technology that is not documented here, see the Implementing and Managing PKI Features Roadmap.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Table 65 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 65 Feature Information for Storing PKI Credentials

Feature Name	Software Releases	Feature Configuration Information
USB Storage	12.3(14)T	This feature enables certain models of Cisco routers to support USB eTokens. USB eTokens provide secure configuration distribution and allow users to VPN credentials for deployment. The following sections provide information about this feature: • Information About Storing PKI Credentials • How to Configure PKI Storage The following commands were introduced or modified by this feature: **copy**, **crypto pki token change-pin**, **crypto pki token login**, **crypto pki token logout**, **crypto pki token max-retries**, **crypto pki token removal timeout**, **crypto pki token secondary config**, **crypto pki token user-pin**, **debug usb driver**, **dir**, **show usb controllers**, **show usb device**, **show usb driver**, **show usbtoken**

Part 6: Other Security Features

Neighbor Router Authentication: Overview and Guidelines

You can prevent your router from receiving fraudulent route updates by configuring neighbor router authentication.

This chapter describes neighbor router authentication as part of a total security plan. It describes what neighbor router authentication is, how it works, and why you should use it to increase your overall network security.

This chapter refers to neighbor router authentication as "neighbor authentication." Neighbor router authentication is also sometimes called "route authentication."

In This Chapter

This chapter describes the following topics:

- About Neighbor Authentication
- How Neighbor Authentication Works
- Key Management (Key Chains)
- Finding Neighbor Authentication Configuration Information

About Neighbor Authentication

This section contains the following sections:

- Benefits of Neighbor Authentication
- Protocols That Use Neighbor Authentication
- When to Configure Neighbor Authentication

Benefits of Neighbor Authentication

When configured, neighbor authentication occurs whenever routing updates are exchanged between neighbor routers. This authentication ensures that a router receives reliable routing information from a trusted source.

Without neighbor authentication, unauthorized or deliberately malicious routing updates could compromise the security of your network traffic. A security compromise could occur if an unfriendly party diverts or analyzes your network traffic. For example, an unauthorized router could send a fictitious routing update to convince your router to send traffic to an incorrect destination. This diverted traffic could be analyzed to learn confidential information of your organization, or merely used to disrupt your organization's ability to effectively communicate using the network.

Neighbor authentication prevents any such fraudulent route updates from being received by your router.

Protocols That Use Neighbor Authentication

Neighbor authentication can be configured for the following routing protocols:

- Border Gateway Protocol (BGP)
- DRP Server Agent
- Intermediate System-to-Intermediate System (IS-IS)
- IP Enhanced Interior Gateway Routing Protocol (EIGRP)
- Open Shortest Path First (OSPF)
- Routing Information Protocol (RIP) version 2

When to Configure Neighbor Authentication

You should configure any router for neighbor authentication if that router meets all of these conditions:

- The router uses any of the routing protocols previously mentioned.
- It is conceivable that the router might receive a false route update.
- If the router were to receive a false route update, your network might be compromised.
- If you configure a router for neighbor authentication, you also need to configure the neighbor router for neighbor authentication.

How Neighbor Authentication Works

When neighbor authentication has been configured on a router, the router authenticates the source of each routing update packet that it receives. This is accomplished by the exchange of an authenticating key (sometimes referred to as a password) that is known to both the sending and the receiving router.

There are two types of neighbor authentication used: plain text authentication and Message Digest Algorithm Version 5 (MD5) authentication. Both forms work in the same way, with the exception that MD5 sends a "message digest" instead of the authenticating key itself. The message digest is created using the key and a message, but the key itself is not sent, preventing it from being read while it is being transmitted. Plain text authentication sends the authenticating key itself over the wire.

Note Note that plain text authentication is not recommended for use as part of your security strategy. Its primary use is to avoid accidental changes to the routing infrastructure. Using MD5 authentication, however, is a recommended security practice.

Caution As with all keys, passwords, and other security secrets, it is imperative that you closely guard authenticating keys used in neighbor authentication. The security benefits of this feature are reliant upon your keeping all authenticating keys confidential. Also, when performing router management tasks via Simple Network Management Protocol (SNMP), do not ignore the risk associated with sending keys using non-encrypted SNMP.

This section includes the following sections:

- Plain Text Authentication
- MD5 Authentication

Plain Text Authentication

Each participating neighbor router must share an authenticating key. This key is specified at each router during configuration. Multiple keys can be specified with some protocols; each key must then be identified by a key number.

In general, when a routing update is sent, the following authentication sequence occurs:

Step 1 A router sends a routing update with a key and the corresponding key number to the neighbor router. In protocols that can have only one key, the key number is always zero.

Step 2 The receiving (neighbor) router checks the received key against the same key stored in its own memory.

Step 3 If the two keys match, the receiving router accepts the routing update packet. If the two keys do not match, the routing update packet is rejected.

These protocols use plain text authentication:

- DRP Server Agent
- IS-IS
- OSPF
- RIP version 2

MD5 Authentication

MD5 authentication works similarly to plain text authentication, except that the key is never sent over the wire. Instead, the router uses the MD5 algorithm to produce a “message digest” of the key (also called a “hash”). The message digest is then sent instead of the key itself. This ensures that nobody can eavesdrop on the line and learn keys during transmission.

These protocols use MD5 authentication:

- OSPF
- RIP version 2
- BGP
- IP Enhanced IGRP

Key Management (Key Chains)

You can configure key chains for these IP routing protocols:

- RIP version 2
- IP Enhanced IGRP
- DRP Server Agent

These routing protocols offer the additional function of managing keys by using key chains. When you configure a key chain, you specify a series of keys with lifetimes, and the Cisco IOS software rotates through each of these keys. This decreases the likelihood that keys will be compromised.

Each key definition within the key chain must specify a time interval for which that key will be activated (its "lifetime"). Then, during a given key's lifetime, routing update packets are sent with this activated key. Keys cannot be used during time periods for which they are not activated. Therefore, it is recommended that for a given key chain, key activation times overlap to avoid any period of time for which no key is activated. If a time period occurs during which no key is activated, neighbor authentication cannot occur, and therefore routing updates will fail.

Multiple key chains can be specified.

Note that the router needs to know the time to be able to rotate through keys in synchronization with the other participating routers, so that all routers are using the same key at the same moment. Refer to the Network Time Protocol (NTP) and calendar commands in the "Performing Basic System Management" chapter of the *Cisco IOS Configuration Fundamentals Configuration Guide* for information about configuring time at your router.

Finding Neighbor Authentication Configuration Information

To find complete configuration information for neighbor authentication, refer to the appropriate section and chapter in the *Cisco IOS IP Configuration Guide* as listed in Table 66.

Table 66 Location of Neighbor Authentication Information for Each Supported Protocol

Protocol	Chapter	Section
BGP	"Configuring BGP"	"Configuring Neighbor Options"
DRP Server Agent	"Configuring IP Services"	"Configuring a DRP Server Agent"
IP Enhanced IGRP	"Configuring IP Enhanced IGRP"	"Configuring Enhanced IGRP Route Authentication"
IS-IS	"Configuring Integrated IS-IS"	"Assigning a Password for an Interface" and "Configuring IS-IS Authentication Passwords"
OSPF	"Configuring OSPF"	"Configuring OSPF Interface Parameters" and "Configuring OSPF Area Parameters" and "Creating Virtual Links"
RIP version 2	"Configuring RIP"	"Enabling RIP Authentication"

To find complete configuration information for key chains, refer to the "Managing Authentication Keys" section in the chapter "Configuring IP Routing Protocol-Independent Features" of the *Cisco IOS IP Configuration Guide*.

Configuring IP Security Options

Cisco provides IP Security Option (IPSO) support as described in RFC 1108. Cisco's implementation is only minimally compliant with RFC 1108 because the Cisco IOS software only accepts and generates a 4-byte IPSO.

IPSO is generally used to comply with the U.S. government's Department of Defense security policy.

For a complete description of IPSO commands, refer to the chapter "IP Security Options Commands" of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Finding Additional Feature Support Information" section on page cxvii in the chapter Using Cisco IOS Software for Release 12.4.

In This Chapter

This chapter describes how to configure IPSO for both the basic and extended security options described in RFC 1108. This chapter also describes how to configure auditing for IPSO. This chapter includes the following sections:

- IPSO Configuration Task List
- IPSO Configuration Examples

IPSO Configuration Task List

This section describes the following configuration tasks:

- Configuring Basic IP Security Options
- Configuring Extended IP Security Options
- Configuring the DNSIX Audit Trail Facility

Configuring Basic IP Security Options

Cisco's basic IPSO support provides the following features:

- Defines security level on a per-interface basis
- Defines single-level or multilevel interfaces
- Provides a label for incoming packets
- Strips labels on a per-interface basis
- Reorders options to put any basic security options first

To configure basic IPSO, complete the tasks in the following sections:

- Enabling IPSO and Setting the Security Classifications
- Specifying How IP Security Options Are Processed

Enabling IPSO and Setting the Security Classifications

To enable IPSO and set security classifications on an interface, use either of the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# ip security dedicated level authority [authority...]`	Sets an interface to the requested IPSO classification and authorities.
`Router(config-if)# ip security multilevel level1 [authority1...] to level2 authority2 [authority2...]`	Sets an interface to the requested IPSO range of classifications and authorities.

Use the **no ip security** command to reset an interface to its default state.

Specifying How IP Security Options Are Processed

To specify how IP security options are processed, use any of the following optional commands in interface configuration mode:

Command	Purpose
`Router(config-if)# ip security ignore-authorities`	Enables an interface to ignore the authorities field of all incoming packets.
`Router(config-if)# ip security implicit-labelling [level authority [authority...]]`	Classifies packets that have no IPSO with an implicit security label.
`Router(config-if)# ip security extended-allowed`	Accepts packets on an interface that has an extended security option present.
`Router(config-if)# ip security ad`	Ensures that all packets leaving the router on an interface contain a basic security option.
`Router(config-if)# ip security strip`	Removes any basic security option that might be present on a packet leaving the router through an interface.

Command	Purpose
`Router(config-if)# ip security first`	Prioritizes security options on a packet.
`Router(config-if)# ip security reserved-allowed`	Treats as valid any packets that have Reserved1 through Reserved4 security levels.

Default Values for Command Keywords

To fully comply with IPSO, the default values for the minor keywords have become complex. Default value usages include the following:

- The default for all of the minor keywords is *off,* with the exception of **implicit-labelling** and **add**.
- The default value of **implicit-labelling** is *on* if the interface is "unclassified Genser;" otherwise, it is *off*.
- The default value for **add** is *on* if the interface is not "unclassified Genser;" otherwise, it is *off*.

Table 67 provides a list of all default values.

Table 67 Default Security Keyword Values

Interface Type	Level	Authority	Implicit Labeling	Add IPSO
None	None	None	On	Off
Dedicated	Unclassified	Genser	On	Off
Dedicated	Any	Any	Off	On
Multilevel	Any	Any	Off	On

The default value for any interface is "dedicated, unclassified Genser." Note that this implies implicit labeling. This might seem unusual, but it makes the system entirely transparent to packets without options. This is the setting generated when you specify the **no ip security** interface configuration command.

Configuring Extended IP Security Options

Cisco's extended IPSO support is compliant with the Department of Defense Intelligence Information System Network Security for Information Exchange (DNSIX) specification documents. Extended IPSO functionality can unconditionally accept or reject Internet traffic that contains extended security options by comparing those options to configured allowable values. This support allows DNSIX networks to use additional security information to achieve a higher level of security than that achievable with basic IPSO.

Cisco also supports a subset of the security features defined in the DNSIX version 2.1 specification. Specifically, Cisco supports DNSIX definitions of the following:

- How extended IPSO is processed
- Audit trail facility

There are two kinds of extended IPSO fields defined by the DNSIX 2.1 specification and supported by Cisco's implementation of extended IPSO—Network-level Extended Security Option (NLESO) and Auxiliary Extended Security Option (AESO) fields.

NLESO processing requires that security options be checked against configured allowable information, source, and compartment bit values, and requires that the router be capable of inserting extended security options in the IP header.

AESO is similar to NLESO, except that its contents are not checked and are assumed to be valid if its source is listed in the AESO table.

To configure extended IPSO, complete the tasks in the following sections:

- Configuring Global Default Settings
- Attaching ESOs to an Interface
- Attaching AESOs to an Interface

Configuring Global Default Settings

To configure global default setting for extended IPSO, including AESOs, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip security eso-info source compartment-size default-bit`	Configures system-wide default settings.

Attaching ESOs to an Interface

To specify the minimum and maximum sensitivity levels for an interface, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# ip security eso-min source compartment-bits`	Sets the minimum sensitivity level for an interface.
Step 2	`Router(config-if)# ip security eso-max source compartment-bits`	Sets the maximum sensitivity level for an interface.

Attaching AESOs to an Interface

To specify the extended IPSO sources that are to be treated as AESO sources, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip security aeso source compartment-bits`	Specifies AESO sources.

DNSIX version 2.1 causes slow-switching code.

See the "IPSO Configuration Examples" section at the end of this chapter.

Configuring the DNSIX Audit Trail Facility

The audit trail facility is a UDP-based protocol that generates an audit trail of IPSO security violations. This facility allows the system to report security failures on incoming and outgoing packets. The Audit Trail Facility sends DNSIX audit trail messages when a datagram is rejected because of IPSO security violations. This feature allows you to configure organization-specific security information.

The DNSIX audit trail facility consists of two protocols:

- DNSIX Message Deliver Protocol (DMDP) provides a basic message-delivery mechanism for all DNSIX elements.
- Network Audit Trail Protocol provides a buffered logging facility for applications to use to generate auditing information. This information is then passed on to DMDP.

To configure the DNSIX auditing facility, complete the tasks in the following sections:

- Enabling the DNSIX Audit Trail Facility
- Specifying Hosts to Receive Audit Trail Messages
- Specifying Transmission Parameters

Enabling the DNSIX Audit Trail Facility

To enable the DNSIX audit trail facility, use the following command in global configuration mode:

Command	Purpose
`Router(config)# dnsix-nat source ip-address`	Starts the audit writing module.

Specifying Hosts to Receive Audit Trail Messages

To define and change primary and secondary addresses of the host to receive audit messages, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dnsix-nat primary ip-address`	Specifies the primary address for the audit trail.
Step 2	`Router(config)# dnsix-nat secondary ip-address`	Specifies the secondary address for the audit trail.
Step 3	`Router(config)# dnsix-nat authorized-redirection ip-address`	Specifies the address of a collection center that is authorized to change primary and secondary addresses. Specified hosts are authorized to change the destination of audit messages.

Specifying Transmission Parameters

To specify transmission parameters, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dnsix-nat transmit-count count`	Specifies the number of records in a packet before it is sent to a collection center.
Step 2	`Router(config)# dnsix-dmdp retries count`	Specifies the number of transmit retries for DMDP.

IPSO Configuration Examples

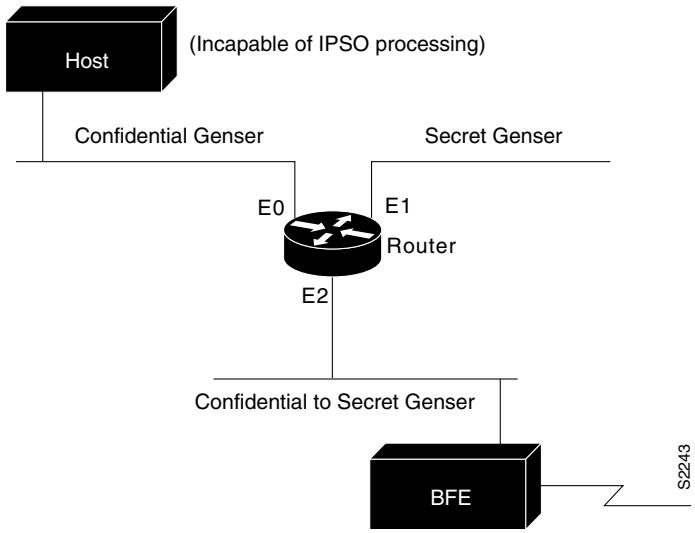
The following sections provide IPSO configuration examples:

- Example 1
- Example 2
- Example 3

Example 1

In this example, three Ethernet interfaces are presented. These interfaces are running at security levels of Confidential Genser, Secret Genser, and Confidential to Secret Genser, as shown in Figure 106.

Figure 106 IPSO Security Levels



The following commands set up interfaces for the configuration in Figure 106:

```
interface ethernet 0
 ip security dedicated confidential genser
interface ethernet 1
 ip security dedicated secret genser
interface ethernet 2
 ip security multilevel confidential genser to secret genser
```

It is possible for the setup to be much more complex.

Example 2

In the following example, there are devices on Ethernet 0 that cannot generate a security option, and so must accept packets without a security option. These hosts do not understand security options; therefore, never place one on such interfaces. Furthermore, there are hosts on the other two networks that are using the extended security option to communicate information, so you must allow these to pass through the system. Finally, there also is a host (a Blacker Front End; see the "Configuring X.25 and LABP" chapter of the *Cisco IOS Wide-Area Networking Configuration Guide* for more information about Blacker emergency mode) on Ethernet 2 that requires the security option to be the first option present, and this condition also must be specified. The new configuration follows.

```
interface ethernet 0
 ip security dedicated confidential genser
 ip security implicit-labelling
 ip security strip
interface ethernet 1
 ip security dedicated secret genser
 ip security extended-allowed
!
interface ethernet 2
 ip security multilevel confidential genser to secret genser
 ip security extended-allowed
 ip security first
```

Example 3

This example shows how to configure a Cisco router with HP-UX CMW DNSIX hosts. The following commands should be configured on each LAN interface of the router for two DNSIX hosts to communicate:

```
ip security multilevel unclassified nsa to top secret nsa
ip security extended allowed
```

DNSIX hosts do not need to know the router's IP addresses, and DNSIX hosts do not need to set up M6RHDB entries for the routers.

Unicast Reverse Path Forwarding

This part consists of the following:

- Configuring Unicast Reverse Path Forwarding

Configuring Unicast Reverse Path Forwarding

This chapter describes the Unicast Reverse Path Forwarding (Unicast RPF) feature. The Unicast RPF feature helps to mitigate problems that are caused by malformed or forged IP source addresses that are passing through a router.

For a complete description of the Unicast RPF commands in this chapter, refer to the chapter "Unicast Reverse Path Forwarding Commands" of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Finding Additional Feature Support Information" section on page cxvii in the chapter Using Cisco IOS Software for Release 12.4.

In This Chapter

This chapter has the following sections:

- About Unicast Reverse Path Forwarding
- Unicast RPF Configuration Task List
- Troubleshooting Tips
- Monitoring and Maintaining Unicast RPF
- Unicast RPF Configuration Examples

About Unicast Reverse Path Forwarding

The Unicast RPF feature helps to mitigate problems that are caused by the introduction of malformed or forged (spoofed) IP source addresses into a network by discarding IP packets that lack a verifiable IP source address. For example, a number of common types of denial-of-service (DoS) attacks, including Smurf and Tribal Flood Network (TFN), can take advantage of forged or rapidly changing source IP addresses to allow attackers to thwart efforts to locate or filter the attacks. For Internet service providers (ISPs) that provide public access, Unicast RPF deflects such attacks by forwarding only packets that have source addresses that are valid and consistent with the IP routing table. This action protects the network of the ISP, its customer, and the rest of the Internet.

This section covers the following information:

- How Unicast RPF Works
- Implementing Unicast RPF
- Restrictions
- Related Features and Technologies
- Prerequisites to Configuring Unicast RPF

How Unicast RPF Works

When Unicast RPF is enabled on an interface, the router examines all packets received as input on that interface to make sure that the source address and source interface appear in the routing table and match the interface on which the packet was received. This "look backwards" ability is available only when Cisco express forwarding (CEF) is enabled on the router, because the lookup relies on the presence of the Forwarding Information Base (FIB). CEF generates the FIB as part of its operation.

Note Unicast RPF is an input function and is applied only on the input interface of a router at the upstream end of a connection.

Unicast RPF checks to see if any packet received at a router interface arrives on the best return path (return route) to the source of the packet. Unicast RPF does this by doing a reverse lookup in the CEF table. If the packet was received from one of the best reverse path routes, the packet is forwarded as normal. If there is no reverse path route on the same interface from which the packet was received, it might mean that the source address was modified. If Unicast RPF does not find a reverse path for the packet, the packet is dropped or forwarded, depending on whether an access control list (ACL) is specified in the **ip verify unicast reverse-path** interface configuration command.

Note With Unicast RPF, all equal-cost "best" return paths are considered valid. This means that Unicast RPF works in cases where multiple return paths exist, provided that each path is equal to the others in terms of the routing cost (number of hops, weights, and so on) and as long as the route is in the FIB. Unicast RPF also functions where EIGRP variants are being used and unequal candidate paths back to the source IP address exist.

When a packet is received at the interface where Unicast RPF and ACLs have been configured, the following actions occur:

Step 1 Input ACLs configured on the inbound interface are checked.

Step 2 Unicast RPF checks to see if the packet has arrived on the best return path to the source, which it does by doing a reverse lookup in the FIB table.

Step 3 CEF table (FIB) lookup is carried out for packet forwarding.

Step 4 Output ACLs are checked on the outbound interface.

Step 5 The packet is forwarded.

This section provides information about Unicast RPF enhancements:

- Access Control Lists and Logging
- Per-Interface Statistics

Access Control Lists and Logging

If an ACL is specified in the command, then when (and only when) a packet fails the Unicast RPF check, the ACL is checked to see if the packet should be dropped (using a deny statement in the ACL) or forwarded (using a permit statement in the ACL). Whether a packet is dropped or forwarded, the packet is counted in the global IP traffic statistics for Unicast RPF drops and in the interface statistics for Unicast RPF.

If no ACL is specified in the Unicast RPF command, the router drops the forged or malformed packet immediately and no ACL logging occurs. The router and interface Unicast RPF counters are updated.

Unicast RPF events can be logged by specifying the logging option for the ACL entries used by the Unicast RPF command. Using the log information, administrators can see what source addresses are being used in the attack, the time the packets arrived at the interface, and so on.

Caution Logging requires CPU and memory resources. Logging Unicast RPF events for attacks having a high rate of forged packets can degrade the performance of the router.

Per-Interface Statistics

Each time a packet is dropped or forwarded at an interface, that information is counted two ways: globally on the router and at each interface where you have applied Unicast RPF. Global statistics on dropped packets provide information about potential attacks on the network; however, these global statistics do not help to specify which interface is the source of the attack.

Per-interface statistics allow network administrators to track two types of information about malformed packets: Unicast RPF drops and Unicast RPF suppressed drops. Statistics on the number of packets that Unicast RPF drops help to identify the interface that is the entry point of the attack. The Unicast RPF drop count tracks the number of drops at the interface. The Unicast RPF suppressed drop count tracks the number of packets that failed the Unicast RPF check but were forwarded because of the permit permission set up in the ACL. Using the drop count and suppressed drop count statistics, a network administrator can takes steps to isolate the attack at a specific interface.

Note Judicious use of ACL logging can further identify the address or addresses that are being dropped by Unicast RPF.

Figure 107 illustrates how Unicast RPF and CEF work together to validate IP source addresses by verifying packet return paths. In this example, a customer has sent a packet having a source address of 192.168.1.1 from interface FDDI 2/0/0. Unicast RPF checks the FIB to see if 192.168.1.1 has a path to FDDI 2/0/0. If there is a matching path, the packet is forwarded. If there is no matching path, the packet is dropped.

Figure 107 Unicast RPF Validating IP Source Addresses

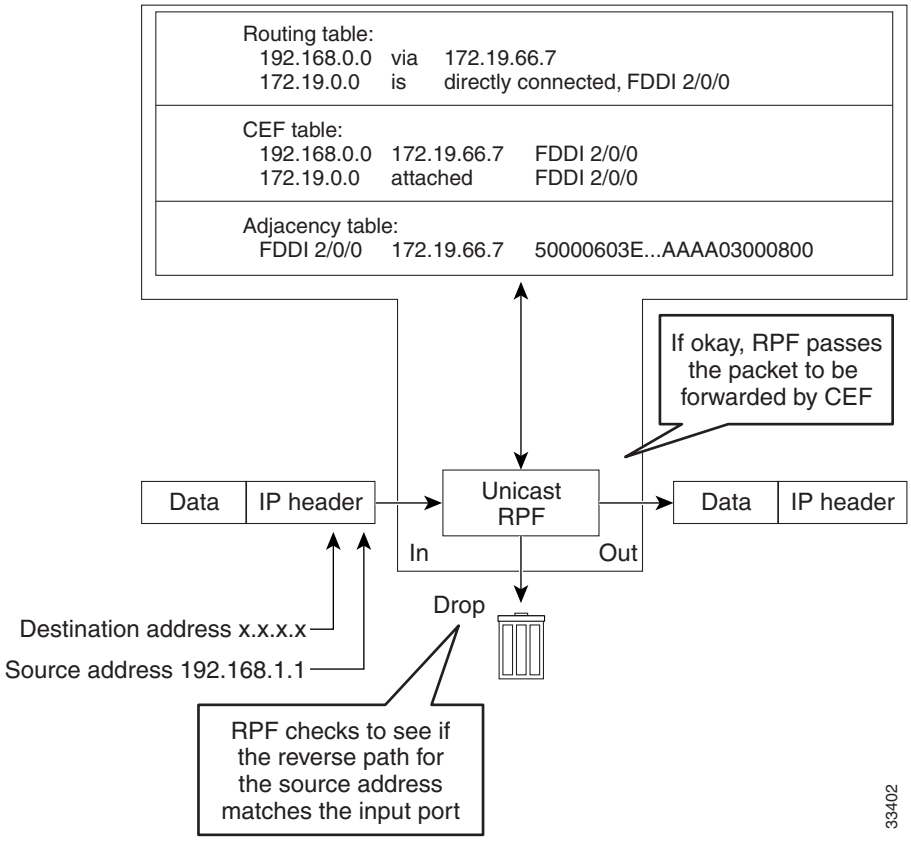
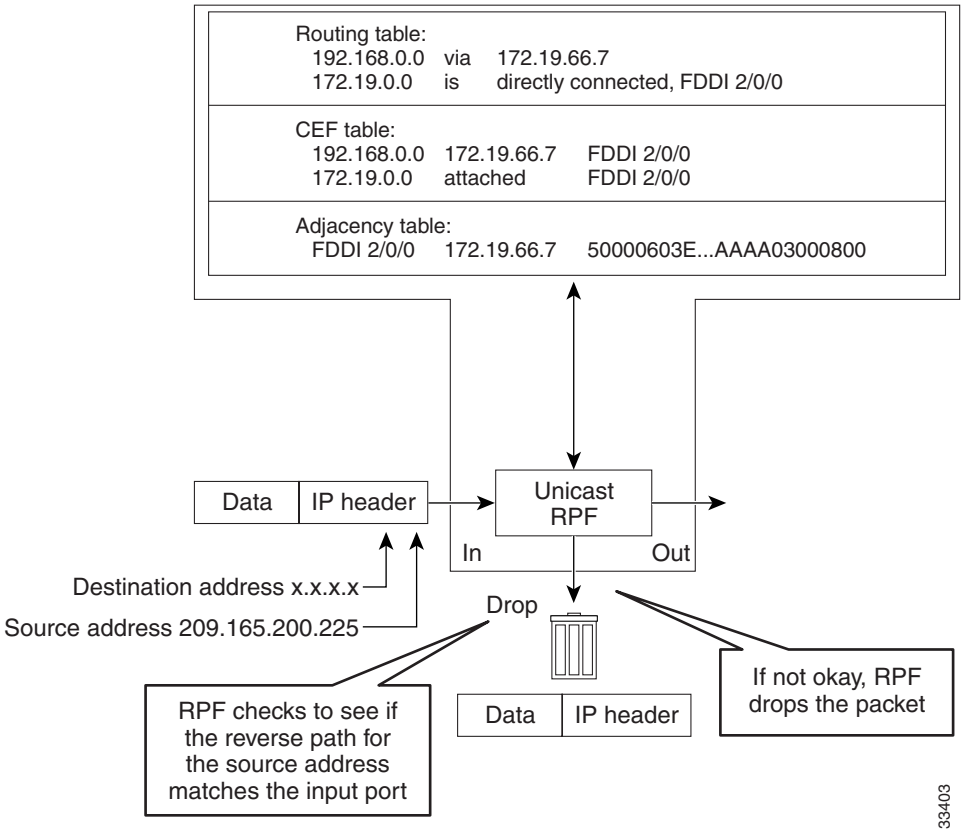


Figure 108 illustrates how Unicast RPF drops packets that fail validation. In this example, a customer has sent a packet having a source address of 209.165.200.225, which is received at interface FDDI 2/0/0. Unicast RPF checks the FIB to see if 209.165.200.225 has a return path to FDDI 2/0/0. If there is a matching path, the packet is forwarded. In this case, there is no reverse entry in the routing table that routes the customer packet back to source address 209.165.200.225 on interface FDDI 2/0/0, and so the packet is dropped.

Figure 108 Unicast RPF Dropping Packets That Fail Verification



Implementing Unicast RPF

Unicast RPF has several key implementation principles:

- The packet must be received at an interface that has the best return path (route) to the packet source (a process called *symmetric routing*). There must be a route in the FIB matching the route to the receiving interface. Adding a route in the FIB can be done via static route, network statement, or dynamic routing. (ACLs permit Unicast RPF to be used when packets are known to be arriving by specific, less optimal asymmetric input paths.)
- IP source addresses at the receiving interface must match the routing entry for the interface.
- Unicast RPF is an input function and is applied only on the input interface of a router at the upstream end of a connection.

Given these implementation principles, Unicast RPF becomes a tool that network administrators can use not only for their customers but also for their downstream network or ISP, even if the downstream network or ISP has other connections to the Internet.

Caution Using optional BGP attributes such as weight and local preference, the best path back to the source address can be modified. Modification would affect the operation of Unicast RPF.

This section provides information about the implementation of Unicast RPF:

- Security Policy and Unicast RPF
- Where to Use Unicast RPF
- Routing Table Requirements
- Where Not to Use Unicast RPF
- Unicast RPF with BOOTP and DHCP

Security Policy and Unicast RPF

Consider the following points in determining your policy for deploying Unicast RPF:

- Unicast RPF must be applied at the interface downstream from the larger portion of the network, preferably at the edges of your network.
- The further downstream you apply Unicast RPF, the finer the granularity you have in mitigating address spoofing and in identifying the sources of spoofed addresses. For example, applying Unicast RPF on an aggregation router helps mitigate attacks from many downstream networks or clients and is simple to administer, but it does not help identify the source of the attack. Applying Unicast RPF at the network access server helps limit the scope of the attack and trace the source of the attack; however, deploying Unicast RPF across many sites does add to the administration cost of operating the network.
- The more entities that deploy Unicast RPF across Internet, intranet, and extranet resources, the better the chances of mitigating large-scale network disruptions throughout the Internet community, and the better the chances of tracing the source of an attack.
- Unicast RPF will not inspect IP packets encapsulated in tunnels, such as GRE, LT2P, or PPTP. Unicast RPF must be configured at a home gateway so that Unicast RPF processes network traffic only after the tunneling and encryption layers have been stripped off the packets.

Where to Use Unicast RPF

Unicast RPF can be used in any “single-homed” environment where there is essentially only one access point out of the network; that is, one upstream connection. Networks having one access point offer the best example of symmetric routing, which means that the interface where a packet enters the network is also the best return path to the source of the IP packet. Unicast RPF is best used at the network perimeter for Internet, intranet, or extranet environments, or in ISP environments for customer network terminations.

The following sections provide a look at implementing Unicast RPF in two network environments:

- Enterprise Networks with a Single Connection to an ISP
- Network Access Server Application (Applying Unicast RPF in PSTN/ISDN PoP Aggregation Routers)

Enterprise Networks with a Single Connection to an ISP

In enterprise networks, one objective of using Unicast RPF for filtering traffic at the input interface (a process called *ingress filtering*) is for protection from malformed packets arriving from the Internet. Traditionally, local networks that have one connection to the Internet would use ACLs at the receiving interface to prevent spoofed packets from the Internet from entering their local network.

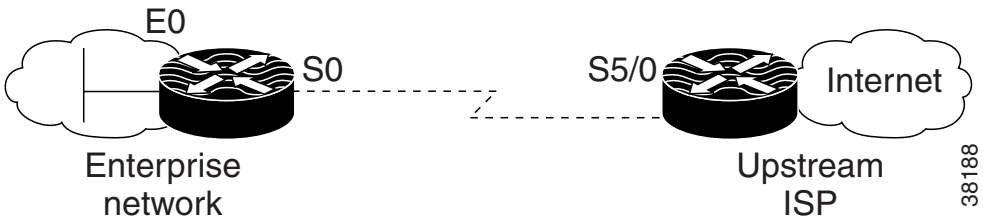
ACLs work well for many single-homed customers; however, there are trade-offs when ACLs are used as ingress filters, including two commonly referenced limitations:

- Packet per second (PPS) performance at very high packet rates
- Maintenance of the ACL (whenever there are new addresses added to the network)

Unicast RPF is one tool that addresses both of these limitations. With Unicast RPF, ingress filtering is done at CEF PPS rates. This processing speed makes a difference when the link is more than 1 Mbps. Additionally, since Unicast RPF uses the FIB, no ACL maintenance is necessary, and thus the administration overhead of traditional ACLs is reduced. The following figure and example demonstrate how Unicast RPF is configured for ingress filtering.

Figure 109 illustrates an enterprise network that has a single link to an upstream ISP. In this example, Unicast RPF is applied at interface S0 on the enterprise router for protection from malformed packets arriving from the Internet. Unicast RPF is also applied at interface S5/0 on the ISP router for protection from malformed packets arriving from the enterprise network.

Figure 109 Enterprise Network Using Unicast RPF for Ingress Filtering



Using the topography in Figure 109, a typical configuration (assuming that CEF is turned on) on the ISP router would be as follows:

```
ip cef
interface loopback 0
  description Loopback interface on Gateway Router 2
  ip address 192.168.3.1 255.255.255.255
  no ip redirects
  no ip directed-broadcast
  no ip proxy-arp
interface Serial 5/0
  description 128K HDLC link to ExampleCorp WT50314E  R5-0
  bandwidth 128
  ip unnumbered loopback 0
  ip verify unicast reverse-path
  no ip redirects
  no ip directed-broadcast
  no ip proxy-arp
ip route 192.168.10.0 255.255.252.0 Serial 5/0
```

The gateway router configuration of the enterprise network (assuming that CEF is turned on) would look similar to the following:

```
ip cef
interface Ethernet 0
 description ExampleCorp LAN
 ip address 192.168.10.1 255.255.252.0
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
```

```
interface Serial 0
 description 128K HDLC link to ExampleCorp Internet Inc WT50314E  C0
 bandwidth 128
 ip unnumbered ethernet 0
 ip verify unicast reverse-path
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
ip route 0.0.0.0 0.0.0.0 Serial 0
```

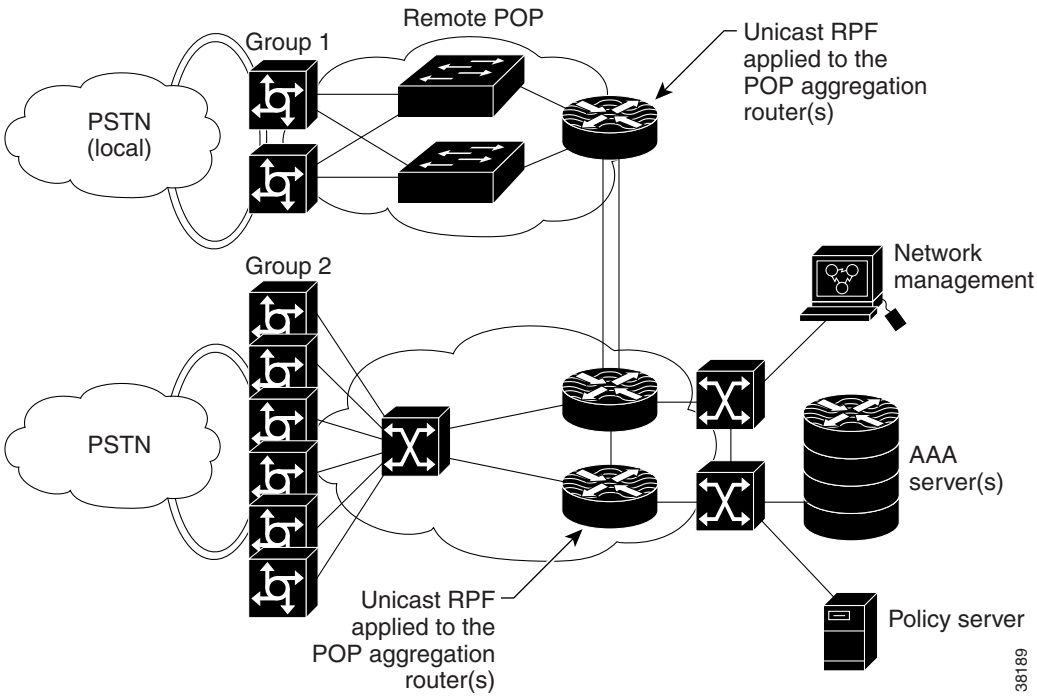
Notice that Unicast RPF works with a single default route. There are no additional routes or routing protocols. Network 192.168.10.0/22 is a connected network. Hence, packets coming from the Internet with a source address in the range 192.168.10.0/22 will be dropped by Unicast RPF.

Network Access Server Application (Applying Unicast RPF in PSTN/ISDN PoP Aggregation Routers)

Aggregation routers are ideal places to use Unicast RPF with single-homed clients. Unicast RPF works equally well on leased-line or PSTN/ISDN/xDSL customer connections into the Internet. In fact, dialup connections are reputed to be the greatest source of DoS attacks using forged IP addresses. As long as the network access server supports CEF, Unicast RPF will work. In this topology, the customer aggregation routers need not have the full Internet routing table. Aggregation routers need the routing prefixes information (IP address block); hence, information configured or redistributed in the Interior Gateway Protocol (IGP) or Internal Border Gateway Protocol (IBGP) (depending on the way that you add customer routes into your network) would be enough for Unicast RPF to do its job.

Figure 110 illustrates the application of Unicast RPF to the aggregation and access routers for an Internet service provider (ISP) point of presence (POP), with the ISP routers providing dialup customer connections. In this example, Unicast RPF is applied upstream from the customer dialup connection router on the receiving (input) interfaces of the ISP aggregation routers.

Figure 110 Unicast RPF Applied to PSTN/ISDN Customer Connections



Routing Table Requirements

To work correctly, Unicast RPF needs proper information in the CEF tables. This requirement does not mean that the router must have the entire Internet routing table. The amount of routing information needed in the CEF tables depends on where Unicast RPF is configured and what functions the router performs in the network. For example, in an ISP environment, a router that is a leased-line aggregation router for customers needs only the information based on the static routes redistributed into the IGP or IBGP (depending on which technique is used in the network). Unicast RPF would be configured on the customer interfaces—hence the requirement for minimal routing information. In another scenario, a single-homed ISP could place Unicast RPF on the gateway link to the Internet. The full Internet routing table would be required. Requiring the full routing table would help protect the ISP from external DoS attacks that use addresses that are not in the Internet routing table.

Where Not to Use Unicast RPF

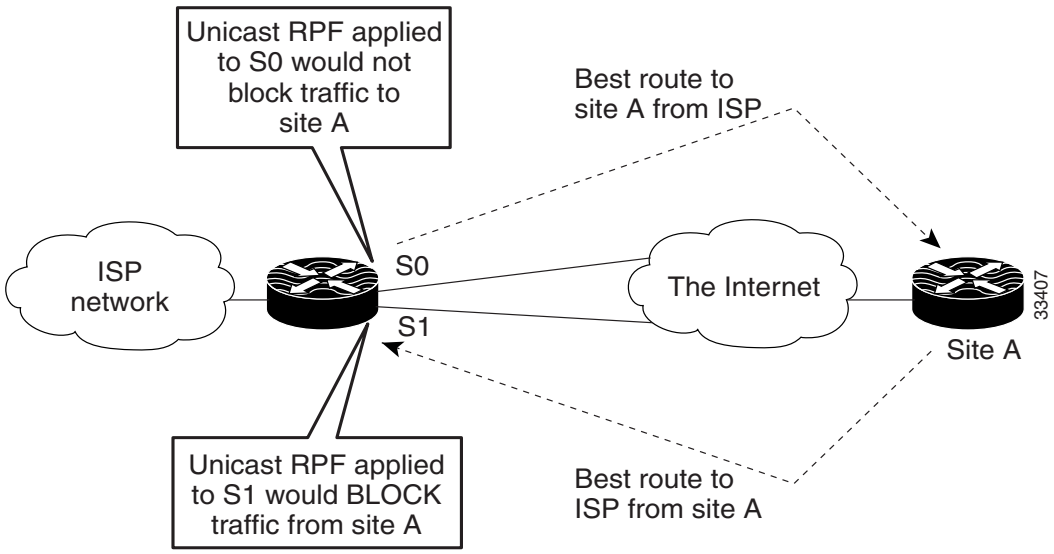
Unicast RPF should not be used on interfaces that are internal to the network. Internal interfaces are likely to have routing asymmetry (see Figure 111), meaning multiple routes to the source of a packet. Unicast RPF should be applied only where there is natural or configured symmetry. As long as administrators carefully plan which interfaces they activate Unicast RPF on, routing asymmetry is not a serious problem.

For example, routers at the edge of the network of an ISP are more likely to have symmetrical reverse paths than routers that are in the core of the ISP network. Routers that are in the core of the ISP network have no guarantee that the best forwarding path out of the router will be the path selected for packets returning to the router. Hence, it is not recommended that you apply Unicast RPF where there is a chance of asymmetric routing, unless you use ACLs to allow the router to accept incoming packets. ACLs permit

Unicast RPF to be used when packets are known to be arriving by specific, less optimal asymmetric input paths. However, it is simplest to place Unicast RPF only at the edge of a network or, for an ISP, at the customer edge of the network.

Figure 111 illustrates how Unicast RPF can block legitimate traffic in an asymmetrical routing environment.

Figure 111 Unicast RPF Blocking Traffic in an Asymmetrical Routing Environment



Unicast RPF with BOOTP and DHCP

Unicast RPF will allow packets with 0.0.0.0 source and 255.255.255.255 destination to pass so that Bootstrap Protocol (BOOTP) and Dynamic Host Configuration Protocol (DHCP) functions work properly. This enhancement was added in Cisco IOS Release 12.0 and later, but it is not in Cisco IOS Release 11.1CC.

Restrictions

There are some basic restrictions to applying Unicast RPF to multihomed clients:

- Clients should not be multihomed to the same router because multihoming defeats the purpose of building a redundant service for the client.
- Customers must ensure that the packets flowing up the link (out to the Internet) match the route advertised out the link. Otherwise, Unicast RPF filters those packets as malformed packets.
- Unicast RPF is available only for platform images that support CEF. Unicast RPF is supported in Cisco IOS Releases 11.1(17)CC and 12.0 and later. It is not available in Cisco IOS Release 11.2 or 11.3.

Related Features and Technologies

For more information about Unicast RPF-related features and technologies, review the following:

- Unicast RPF requires Cisco express forwarding (CEF) to function properly on the router. For more information about CEF, refer to the *Cisco IOS Switching Services Configuration Guide*.
- Unicast RPF can be more effective at mitigating spoofing attacks when combined with a policy of *ingress* and *egress* filtering using Cisco IOS access control lists (ACLs).
  - Ingress filtering applies filters to traffic received at a network interface from either internal or external networks. With ingress filtering, packets that arrive from other networks or the Internet and that have a source address that matches a local network, private, or broadcast address are dropped. In ISP environments, for example, ingress filtering can apply to traffic received at the router from either the client (customer) or the Internet.
  - Egress filtering applies filters to traffic exiting a network interface (the sending interface). By filtering packets on routers that connect your network to the Internet or to other networks, you can permit only packets with valid source IP addresses to leave your network.

  For more information on network filtering, refer to RFC 2267 and to the *Cisco IOS IP Configuration Guide*.
- Cisco IOS software provides additional features that can help mitigate DoS attacks:
  - Committed Access Rate (CAR). CAR allows you to enforce a bandwidth policy against network traffic that matches an access list. For example, CAR allows you to rate-limit what should be low-volume traffic, such as ICMP traffic. To find out more about CAR, refer to the *Cisco IOS Quality of Service Solutions Configuration Guide*.
  - Context-based Access Control (CBAC). CBAC selectively blocks any network traffic not originated by a protected network. CBAC uses timeout and threshold values to manage session state information, helping to determine when to drop sessions that do not become fully established. Setting timeout values for network sessions helps mitigate DoS attacks by freeing up system resources, dropping sessions after a specified amount of time. For more information on CBAC, refer to the *Cisco IOS Security Configuration Guide*.
  - TCP Intercept. The TCP Intercept feature implements software to protect TCP servers from TCP SYN-flooding attacks, which are a type of DoS attack. A SYN-flooding attack occurs when a hacker floods a server with a barrage of requests for connection. Like CBAC, the TCP Intercept feature also uses timeout and threshold values to manage session state information, helping to determine when to drop sessions that do not become fully established. For more information on TCP Intercept, refer to the *Cisco IOS Security Configuration Guide*.

Prerequisites to Configuring Unicast RPF

Prior to configuring Unicast RPF, configure ACLs:

- Configure standard or extended ACLs to mitigate transmission of invalid IP addresses (perform egress filtering). Permit only valid source addresses to leave your network and get onto the Internet. Prevent all other source addresses from leaving your network for the Internet.
- Configure standard or extended ACLs entries to drop (deny) packets that have invalid source IP addresses (perform ingress filtering). Invalid source IP addresses include the following types:
  - Reserved addresses
  - Loopback addresses
  - Private addresses (RFC 1918, *Address Allocation for Private Internets*)
  - Broadcast addresses (including multicast addresses)
  - Source addresses that fall outside the range of valid addresses associated with the protected network
- Configure standard or extended ACL entries to forward (permit) packets that fail the Unicast RPF checks to allow specific traffic from known asymmetric routed sources.
- Configure ACLs to track Unicast RPF events by adding the logging option into the ACL command. During network attacks, judicious logging of dropped or forwarded packets (suppressed drops) can provide additional information about network attacks.

Unicast RPF Configuration Task List

The following sections describe the configuration tasks for Unicast RPF. Each task in the list is identified as either optional or required.

- Configuring Unicast RPF (Required)
- Verifying Unicast RPF (Optional)

See the section "Unicast RPF Configuration Examples" at the end of this chapter.

Configuring Unicast RPF

To use Unicast RPF, you must configure the router for CEF switching or CEF distributed switching. There is no need to configure the input interface for CEF switching because Unicast RPF has been implemented as a search through the FIB using the source IP address. As long as CEF is running on the router, individual interfaces can be configured with other switching modes. Unicast RPF is an input-side function that is enabled on an interface or subinterface that supports any type of encapsulation and operates on IP packets received by the router. It is very important that CEF be turned on globally in the router—Unicast RPF will not work without CEF.

To configure Unicast RPF, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip cef` or `Router(config)# ip cef distributed`	Enables CEF or distributed CEF on the router. Distributed CEF is required for routers that use a Route Switch Processor (RSP) and Versatile Interface Processor (VIP), which includes Unicast RPF. You might want to disable CEF or distributed CEF (dCEF) on a particular interface if that interface is configured with a feature that CEF or dCEF does not support. In this case, you would enable CEF globally, but disable CEF on a specific interface using the **no ip route-cache cef** interface command, which enables all but that specific interface to use express forwarding. If you have disabled CEF or dCEF operation on an interface and want to reenable it, you can do so by using the **ip route-cache cef** command in interface configuration mode.
Step 2	`Router(config-if)# interface` *type*	Selects the input interface on which you want to apply Unicast RPF. This is the receiving interface, which allows Unicast RPF to verify the best return path before forwarding the packet on to the next destination. The interface type is specific to your router and the types of interface cards installed on the router. To display a list of available interface types, enter the **interface ?** command.
Step 3	`Router(config-if)# ip verify unicast reverse-path` *list*	Enables Unicast RPF on the interface. Use the *list* option to identify an access list. If the access list denies network access, spoofed packets are dropped at the interface. If the access list permits network access, spoofed packets are forwarded to the destination address. Forwarded packets are counted in the interface statistics. If the access list includes the logging option, information about the spoofed packets is logged to the log server. Repeat this step for each access list that you want specify.
Step 4	`Router(config-if)# exit`	Exits interface configuration mode. Repeat Steps 2 and 3 for each interface on which you want to apply Unicast RPF.

Verifying Unicast RPF

To verify that Unicast RPF is operational, use the **show cef interface** command. The following example shows that Unicast RPF is enabled at interface serial2/0/0.

```
Router-3# show cef interface serial 2/0/0

Serial2/0/0 is up (if_number 8)
 Internet address is 192.168.10.2/30
 ICMP redirects are never sent
 Per packet loadbalancing is disabled
!The next line displays Unicast RPF packet dropping information.
 IP unicast RPF check is enabled
 Inbound access list is not set
 Outbound access list is not set
 Interface is marked as point to point interface
 Packets switched to this interface on linecard are dropped to next slow path
 Hardware idb is Serial2/0/0
 Fast switching type 4, interface type 6
!The next line displays Unicast RPF packet dropping information.
 IP Distributed CEF switching enabled
 IP LES Feature Fast switching turbo vector
 IP Feature CEF switching turbo vector
 Input fast flags 0x40, Output fast flags 0x0, ifindex 7(7)
 Slot 2 Slot unit 0 VC -1
 Transmit limit accumulator 0x48001A02 (0x48001A02)
 IP MTU 1500
```

Troubleshooting Tips

If you experience problems while using Unicast RPF, check the following items.

HSRP Failure

Failure to disable Unicast RPF before disabling CEF can cause Hot Standby Router Protocol (HSRP) failure. If you want to disable CEF on the router, you must first disable Unicast RPF. To disable Unicast RPF, see the section "Monitoring and Maintaining Unicast RPF."

Dropped Boot Requests

In Cisco IOS Release 11.1(17)CC, Unicast RPF can drop BOOTP request packets that have a source address of 0.0.0.0 due to source address verification at the interface. To enable boot requests to work on the interface, you must use ACLs instead of Unicast RPF.

Monitoring and Maintaining Unicast RPF

This section describes commands used to monitor and maintain Unicast RPF.

Command	Purpose
`Router# show ip traffic`	Displays global router statistics about Unicast RPF drops and suppressed drops.
`Router# show ip interface` *type*	Displays per-interface statistics about Unicast RPF drops and suppressed drops.
`Router# show access-lists`	Displays the number of matches to a specific ACL.
`Router(config-if)# no ip verify unicast reverse-path` *list*	Disables Unicast RPF at the interface. Use the *list* option to disable Unicast RPF for a specific ACL at the interface.

Caution To disable CEF, you must first disable Unicast RPF. Failure to disable Unicast RPF before disabling CEF can cause HSRP failure. If you want to disable CEF on the router, you must first disable Unicast RPF.

Unicast RPF counts the number of packets dropped or suppressed because of malformed or forged source addresses. Unicast RPF counts dropped or forwarded packets that include the following global and per-interface information:

- Global Unicast RPF drops
- Per-interface Unicast RPF drops
- Per-interface Unicast RPF suppressed drops

The **show ip traffic** command shows the total number (global count) of dropped or suppressed packets for all interfaces on the router. The Unicast RPF drop count is included in the IP statistics section.

```
Router# show ip traffic

IP statistics:
  Rcvd:  1471590 total, 887368 local destination
         0 format errors, 0 checksum errors, 301274 bad hop count
         0 unknown protocol, 0 not a gateway
         0 security failures, 0 bad options, 0 with options
  Opts:  0 end, 0 nop, 0 basic security, 0 loose source route
         0 timestamp, 0 extended security, 0 record route
         0 stream ID, 0 strict source route, 0 alert, 0 other
  Frags: 0 reassembled, 0 timeouts, 0 couldn't reassemble
         0 fragmented, 0 couldn't fragment
  Bcast: 205233 received, 0 sent
  Mcast: 463292 received, 462118 sent
  Sent:  990158 generated, 282938 forwarded
  ! The second line below ("0 unicast RPF") displays Unicast RPF packet dropping
  information.
  Drop:  3 encapsulation failed, 0 unresolved, 0 no adjacency
         0 no route, 0 unicast RPF, 0 forced drop
```

A nonzero value for the count of dropped or suppressed packets can mean one of two things:

- Unicast RPF is dropping or suppressing packets that have a bad source address (normal operation).

- Unicast RPF is dropping or suppressing legitimate packets because the route is misconfigured to use Unicast RPF in environments where asymmetric routing exists; that is, where multiple paths can exist as the best return path for a source address.

The **show ip interface** command shows the total of dropped or suppressed packets at a specific interface. If Unicast RPF is configured to use a specific ACL, that ACL information is displayed along with the drop statistics.

```
Router> show ip interface ethernet0/1/1

   Unicast RPF ACL 197
   1 unicast RPF drop
   1 unicast RPF suppressed drop
```

The **show access-lists** command displays the number of matches found for a specific entry in a specific access list.

```
Router> show access-lists

Extended IP access list 197
     deny ip 192.168.201.0 0.0.0.63 any log-input (1 match)
     permit ip 192.168.201.64 0.0.0.63 any log-input (1 match)
     deny ip 192.168.201.128 0.0.0.63 any log-input
     permit ip 192.168.201.192 0.0.0.63 any log-input
```

Unicast RPF Configuration Examples

This section provides the following configuration examples:

- Unicast RPF on a Leased-Line Aggregation Router Example
- Unicast RPF on the Cisco AS5800 Using Dialup Ports Example
- Unicast RPF with Inbound and Outbound Filters Example
- Unicast RPF with ACLs and Logging Example

Unicast RPF on a Leased-Line Aggregation Router Example

The following commands enable Unicast RPF on a serial interface:

```
ip cef
! or "ip cef distributed" for RSP+VIP based routers
!
interface serial 5/0/0
 ip verify unicast reverse-path
```

Unicast RPF on the Cisco AS5800 Using Dialup Ports Example

The following example enables Unicast RPF on a Cisco AS5800. The **interface Group-Async** command makes it easy to apply Unicast RPF on all the dialup ports.

```
ip cef
!
interface Group-Async1
 ip verify unicast reverse-path
```

Unicast RPF with Inbound and Outbound Filters Example

The following example uses a very simple single-homed ISP to demonstrate the concepts of ingress and egress filters used in conjunction with Unicast RPF. The example illustrates an ISP-allocated classless interdomain routing (CIDR) block 209.165.202.128/28 that has both inbound and outbound filters on the upstream interface. Be aware that ISPs are usually not single-homed. Hence, provisions for asymmetrical flows (when outbound traffic goes out one link and returns via a different link) need to be designed into the filters on the border routers of the ISP.

```
ip cef distributed
!
interface Serial 5/0/0
 description Connection to Upstream ISP
 ip address 209.165.200.225 255.255.255.252
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
 ip verify unicast reverse-path
 ip access-group 111 in
 ip access-group 110 out
!
access-list 110 permit ip 209.165.202.128 0.0.0.31 any
access-list 110 deny ip any any log
access-list 111 deny ip host 0.0.0.0 any log
access-list 111 deny ip 127.0.0.0 0.255.255.255 any log
access-list 111 deny ip 10.0.0.0 0.255.255.255 any log
access-list 111 deny ip 172.16.0.0 0.15.255.255 any log
access-list 111 deny ip 192.168.0.0 0.0.255.255 any log
access-list 111 deny ip 209.165.202.128 0.0.0.31 any log
access-list 111 permit ip any any
```

Unicast RPF with ACLs and Logging Example

The following example demonstrates the use of ACLs and logging with Unicast RPF. In this example, extended ACL 197 provides entries that deny or permit network traffic for specific address ranges. Unicast RPF is configured on interface Ethernet0 to check packets arriving at that interface.

For example, packets with a source address of 192.168.201.10 arriving at interface Ethernet0 are dropped because of the deny statement in ACL 197. In this case, the ACL information is logged (the logging option is turned on for the ACL entry) and dropped packets are counted per interface and globally. Packets with a source address of 192.168.201.100 arriving at interface Ethernet0 are forwarded because of the permit statement in ACL 197. ACL information about dropped or suppressed packets is logged (logging option turned on for the ACL entry) to the log server.

```
ip cef distributed
!
int eth0/1/1
 ip address 192.168.200.1 255.255.255.0
 ip verify unicast reverse-path 197
!
int eth0/1/2
 ip address 192.168.201.1 255.255.255.0
!
access-list 197 deny   ip 192.168.201.0 0.0.0.63 any log-input
access-list 197 permit ip 192.168.201.64 0.0.0.63 any log-input
access-list 197 deny   ip 192.168.201.128 0.0.0.63 any log-input
access-list 197 permit ip 192.168.201.192 0.0.0.63 any log-input
access-list 197 deny ip host 0.0.0.0 any log
```

Secure Shell

This part consists of the following:

- Configuring Secure Shell
- Reverse SSH Enhancements
- Secure Copy
- Secure Shell Version 2 Support
- SSH Terminal-Line Access

Configuring Secure Shell

This chapter describes the Secure Shell (SSH) feature. The SSH feature consists of an application and a protocol.

For a complete description of the SSH commands in this chapter, refer to the chapter "Secure Shell Commands" of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Finding Additional Feature Support Information" section on page cxvii in the chapter Using Cisco IOS Software for Release 12.4

In This Chapter

This chapter has the following sections:

- About Secure Shell
- SSH Configuration Task List
- Troubleshooting Tips
- Monitoring and Maintaining SSH
- SSH Configuration Examples

About Secure Shell

Secure Shell (SSH) is an application and a protocol that provide a secure replacement to the Berkeley r-tools. The protocol secures the sessions using standard cryptographic mechanisms, and the application can be used similarly to the Berkeley **rexec** and **rsh** tools. There are currently two versions of SSH available: SSH Version 1 and SSH Version 2. Only SSH Version 1 is implemented in the Cisco IOS software.

Note Hereafter, unless otherwise noted, the term "SSH" will denote "SSH Version 1" only.

This rest of this section covers the following information:

- How SSH Works
- Restrictions
- Related Features and Technologies
- Prerequisites to Configuring SSH

How SSH Works

This section provides the following information about how SSH works:

- SSH Server
- SSH Integrated Client

SSH Server

The SSH Server feature enables a SSH client to make a secure, encrypted connection to a Cisco router. This connection provides functionality that is similar to that of an inbound Telnet connection. Before SSH, security was limited to Telnet security. SSH allows a strong encryption to be used with the Cisco IOS software authentication. The SSH server in Cisco IOS software will work with publicly and commercially available SSH clients.

SSH Integrated Client

The SSH Integrated Client feature is an application running over the SSH protocol to provide device authentication and encryption. The SSH client enables a Cisco router to make a secure, encrypted connection to another Cisco router or to any other device running the SSH server. This connection provides functionality that is similar to that of an outbound Telnet connection except that the connection is encrypted. With authentication and encryption, the SSH client allows for a secure communication over an insecure network.

The SSH client in the Cisco IOS software works with publicly and commercially available SSH servers. The SSH client supports the ciphers of Data Encryption Standard (DES), Triple DES (3DES), and password authentication. User authentication is performed like that in the Telnet session to the router. The user authentication mechanisms supported for SSH are RADIUS, TACACS+ and the use of locally stored user names and passwords.

Note The SSH client functionality is available only when the SSH server is enabled.

Restrictions

There following are some basic SSH restrictions:

- RSA authentication available in SSH clients is not supported in the SSH server for Cisco IOS software.
- SSH server and SSH client are supported on DES (56-bit) and 3DES (168-bit) data encryption software images only. In DES software images, DES is the only encryption algorithm available. In 3DES software images, both DES and 3DES encryption algorithms are available.
- Execution shell is the only application supported.

Related Features and Technologies

For more information about SSH-related features and technologies, review the following:

- Authentication, Authorization, and Accounting (AAA) feature. AAA is a suite of network security services that provide the primary framework through which access control can be set up on your Cisco router or access server. For more information on AAA, refer to the Authentication, Authorization, and Accounting chapters earlier in this book and the *Cisco IOS Security Command Reference*.
- IP Security (IPSec) feature. IPSec is a framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer. IPSec uses Internet Key Exchange (IKE) to handle negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPSec. IPSec can be used to protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host. For more information on IPSec, refer to the chapter "Configuring IPSec Network Security" and the *Cisco IOS Security Command Reference*.

Prerequisites to Configuring SSH

Prior to configuring SSH, perform the following tasks:

- Download the required image on your router. (The SSH server requires you to have an IPSec (DES or 3DES) encryption software image from Cisco IOS Release 12.1(1)T downloaded on your router; the SSH client requires you to have an IPSec (DES or 3DES) encryption software image from Cisco IOS Release 12.1(3)T downloaded on your router.) For more information on downloading a software image, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide*.
- Configure a host name and host domain for your router.

  To configure a host name and host domain, enter the following commands beginning in global configuration mode:

Command	Purpose
`Router(config)# hostname hostname`	Configures a host name for your router.
`Router(config)# ip domain-name domainname`	Configures a host domain for your router.

- Generate an RSA key pair for your router, which automatically enables SSH.

To generate an RSA key pair, enter the following global configuration command:

Command	Purpose
`Router(config)# crypto key generate rsa`	Enables the SSH server for local and remote authentication on the router. The recommended minimum modulus size is 1024 bits. **Note** To delete the RSA key-pair, use the **crypto key zeroize rsa** global configuration command. Once you delete the RSA key-pair, you automatically disable the SSH server.

- Configure user authentication for local or remote access. You can configure authentication with or without AAA. For more information, refer to the "Authentication, Authorization, and Accounting (AAA)" chapters earlier in the book.

SSH Configuration Task List

The following sections describe the configuration tasks for SSH. Each task in the list is identified as either optional or required.

- Configuring SSH Server (Required)
- Verifying SSH (Optional)

See the section "SSH Configuration Examples" at the end of this chapter.

Configuring SSH Server

Note The SSH client feature runs in user EXEC mode and has no specific configuration on the router.

Note The SSH commands are optional and are disabled when the SSH server is disabled.

To enable and configure a Cisco Router for SSH, you can configure SSH parameters. If you do not configure SSH parameters, the the default values will be used.

To configure SSH server, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip ssh {[timeout seconds] | [authentication-retries integer]}`	(Required) Configures SSH control variables on your router. • You can specify the timeout in seconds, not to exceed 120 seconds. The default is 120. This setting applies to the SSH negotiation phase. Once the EXEC session starts, the standard timeouts configured for the vty apply. By default, there are 5 vtys defined (0–4), therefore 5 terminal sessions are possible. After the SSH executes a shell, the vty timeout starts. The vty timeout defaults to 10 minutes. • You can also specify the number of authentication retries, not to exceed 5 authentication retries. The default is 3.

Verifying SSH

To verify that the SSH server is enabled and view the version and configuration data for your SSH connection, use the **show ip ssh** command. The following example shows that SSH is enabled:

```
Router# show ip ssh

SSH Enabled - version 1.5
Authentication timeout: 120 secs; Authentication retries: 3
```

The following example shows that SSH is disabled:

```
Router# show ip ssh

%SSH has not been enabled
```

To verify the status of your SSH server connections, use the **show ssh** command. The following example shows the SSH server connections on the router when SSH is enabled:

```
Router# show ssh
Connection      Version     EncryptionStateUsername
   01.5 3DESSession Startedguest
```

The following example shows that SSH is disabled:

```
Router# show ssh

%No SSH server connections running.
```

Troubleshooting Tips

- If your SSH configuration commands are rejected as illegal commands, you have not successfully generated a RSA key pair for your router. Make sure you have specified a host name and domain. Then use the **crypto key generate rsa** command to generate a RSA key pair and enable the SSH server.
- When configuring the RSA key pair, you might encounter the following error messages:
  - `No hostname specified`

    You must configure a host name for the router using the **hostname** global configuration command. For more information, see "Prerequisites to Configuring SSH."
  - `No domain specified`

    You must configure a host domain for the router using the **ip domain-name** global configuration command. For more information, see "Prerequisites to Configuring SSH."
- The number of allowable SSH connections is limited to the maximum number of vtys configured for the router. Each SSH connection will use a vty resource.
- SSH uses either local security or the security protocol that is configured through AAA on your router for user authentication. When configuring AAA, you must ensure that the console is not running under AAA by applying a keyword in the global configuration mode to disable AAA on the console.

Monitoring and Maintaining SSH

To monitor and maintain your SSH connections, use the following commands in user EXEC mode:

Command	Purpose
`Router# show ip ssh`	Displays the version and configuration data for SSH.
`Router# show ssh`	Displays the status of SSH server connections.

SSH Configuration Examples

This section provides the following configuration examples, which are output from the **show running configuration** EXEC command on a Cisco 7200, Cisco 7500, and Cisco 12000.

- SSH on a Cisco 7200 Series Router Example
- SSH on a Cisco 7500 Series Router Example
- SSH on a Cisco 1200 Gigabit Switch Router Example

Note The **crypto key generate rsa** command is not displayed in the **show running configuration** output.

SSH on a Cisco 7200 Series Router Example

In the following example, SSH is configured on a Cisco 7200 with a timeout that is not to exceed 60 seconds, and no more than 2 authentication retries. Also, before configuring the SSH server feature on the router, TACACS+ is specified as the method of authentication.

```
aaa new-model
aaa authentication login default tacacs+
aaa authentication login aaa7200kw none
enable password enable7200pw

username mcisco password 0 maryspw
username jcisco password 0 johnspw
ip subnet-zero
no ip domain-lookup
ip domain-name cisco.com
! Enter the ssh commands.
ip ssh time-out 60
ip ssh authentication-retries 2

controller E1 2/0

controller E1 2/1

interface Ethernet1/0
ip address 192.168.110.2 255.255.255.0 secondary
ip address 192.168.109.2 255.255.255.0
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
no keepalive
no cdp enable

interface Ethernet1/1
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
shutdown
no cdp enable

interface Ethernet1/2
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
shutdown
no cdp enable

no ip classless
ip route 192.168.1.0 255.255.255.0 10.1.10.1
ip route 192.168.9.0 255.255.255.0 10.1.1.1
ip route 192.168.10.0 255.255.255.0 10.1.1.1

map-list atm
ip 10.1.10.1 atm-vc 7 broadcast
no cdp run
```

```
tacacs-server host 192.168.109.216 port 9000
tacacs-server key cisco
radius-server host 192.168.109.216 auth-port 1650 acct-port 1651
radius-server key cisco

line con 0
exec-timeout 0 0
login authentication aaa7200kw
transport input none
line aux 0
line vty 0 4
password enable7200pw

end
```

SSH on a Cisco 7500 Series Router Example

In the following example, SSH is configured on a Cisco 7500 with a timeout that is not to exceed 60 seconds and no more than 5 authentication retries. Before the SSH Server feature is configured on the router, RADIUS is specified as the method of authentication.

```
aaa new-model
aaa authentication login default radius
aaa authentication login aaa7500kw none
enable password enable7500pw

username mcisco password 0 maryspw
username jcisco password 0 johnspw
ip subnet-zero
no ip cef
no ip domain-lookup
ip domain-name cisco.com
! Enter ssh commands.
ip ssh time-out 60
ip ssh authentication-retries 5

controller E1 3/0
channel-group 0 timeslots 1

controller E1 3/1
channel-group 0 timeslots 1
channel-group 1 timeslots 2

interface Ethernet0/0/0
no ip address
no ip directed-broadcast
no ip route-cache distributed
shutdown

interface Ethernet0/0/1
no ip address
no ip directed-broadcast
no ip route-cache distributed
shutdown
interface Ethernet0/0/2
no ip address
no ip directed-broadcast
no ip route-cache distributed
shutdown
```

```
interface Ethernet0/0/3
no ip address
no ip directed-broadcast
no ip route-cache distributed
shutdown

interface Ethernet1/0
ip address 192.168.110.2 255.255.255.0 secondary
ip address 192.168.109.2 255.255.255.0
no ip directed-broadcast
no ip route-cache
no ip mroute-cache

interface Ethernet1/1
ip address 192.168.109.2 255.255.255.0
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
shutdown

interface Ethernet1/2
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache

interface Ethernet1/3
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
shutdown

interface Ethernet1/4
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
shutdown
interface Ethernet1/5
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
shutdown

interface Serial2/0
ip address 10.1.1.2 255.0.0.0
no ip directed-broadcast
encapsulation ppp
no ip route-cache
no ip mroute-cache

ip classless
ip route 192.168.9.0 255.255.255.0 10.1.1.1
ip route 192.168.10.0 255.255.255.0 10.1.1.1
```

```
tacacs-server host 192.168.109.216 port 9000
tacacs-server key cisco
radius-server host 192.168.109.216 auth-port 1650 acct-port 1651
radius-server key cisco

line con 0
exec-timeout 0 0
login authentication aaa7500kw
transport input none
line aux 0
transport input all
line vty 0 4

end
```

SSH on a Cisco 1200 Gigabit Switch Router Example

In the following example, SSH is configured on a Cisco 12000 with a timeout that is not to exceed 60 seconds and no more than 2 authentication retries. Before the SSH Server feature is configured on the router, TACACS+ is specified as the method of authentication.

```
aaa new-model
aaa authentication login default tacacs+ local
aaa authentication login aaa12000kw local
enable password enable12000pw

username mcisco password 0 maryspw
username jcisco password 0 johnspw
redundancy
main-cpu
    auto-sync startup-config
ip subnet-zero
no ip domain-lookup
ip domain-name cisco.com
! Enter ssh commands.
ip ssh time-out 60
ip ssh authentication-retries 2

interface ATM0/0
no ip address
no ip directed-broadcast
no ip route-cache cef
shutdown

interface POS1/0
ip address 10.100.100.2 255.255.255.0
no ip directed-broadcast
encapsulation ppp
no ip route-cache cef
no keepalive
crc 16
no cdp enable

interface POS1/1
no ip address
no ip directed-broadcast
no ip route-cache cef
shutdown
crc 32
```

```
interface POS1/2
no ip address
no ip directed-broadcast
no ip route-cache cef
shutdown
crc 32

interface POS1/3
no ip address
no ip directed-broadcast
no ip route-cache cef
shutdown
crc 32

interface POS2/0
ip address 10.1.1.1 255.255.255.0
no ip directed-broadcast
encapsulation ppp
no ip route-cache cef
crc 16

interface Ethernet0
ip address 172.17.110.91 255.255.255.224
no ip directed-broadcast

router ospf 1
network 0.0.0.0 255.255.255.255 area 0.0.0.0

ip classless
ip route 0.0.0.0 0.0.0.0 172.17.110.65

logging trap debugging
tacacs-server host 172.17.116.138
tacacs-server key cisco

radius-server host 172.17.116.138 auth-port 1650 acct-port 1651
radius-server key cisco

line con 0
exec-timeout 0 0
login authentication aaa12000kw
transport input none
line aux 0
line vty 0 4

no scheduler max-task-time
no exception linecard slot 0 sqe-registers
no exception linecard slot 1 sqe-registers
no exception linecard slot 2 sqe-registers
no exception linecard slot 3 sqe-registers
no exception linecard slot 4 sqe-registers
no exception linecard slot 5 sqe-registers
no exception linecard slot 6 sqe-registers
end
```

Reverse SSH Enhancements

The Reverse SSH Enhancements feature provides an alternative method of configuring reverse Secure Shell (SSH). Using this feature, you can configure reverse SSH without having to list separate lines for every terminal or auxiliary line on which SSH has to be enabled. This feature also eliminates the rotary-group limitation. This feature is supported for SSH Version 1 and SSH Version 2.

Feature History for Reverse SSH Enhancements

Release	Modification
12.3(11)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for Reverse SSH Enhancements, page 1625
- Restrictions for Reverse SSH Enhancements, page 1626
- Information About Reverse SSH Enhancements, page 1626
- How to Configure Reverse SSH Enhancements, page 1626
- Configuration Examples for Reverse SSH Enhancements, page 1631
- Additional References, page 1632
- Command Reference, page 1633

Prerequisites for Reverse SSH Enhancements

- SSH must be enabled.
- The SSH client and server must be running the same version of SSH.

Restrictions for Reverse SSH Enhancements

- The **-l** keyword and *userid* **:**{*number*} {*ip-address*} delimiter and arguments are mandatory when configuring the alternative method of Reverse SSH for console access.

Information About Reverse SSH Enhancements

To configure Reverse SSH Enhancements, you should understand the following concepts:

- Reverse Telnet, page 1626
- Reverse SSH, page 1626

Reverse Telnet

Cisco IOS software has for quite some time included a feature called Reverse Telnet, whereby you can telnet to a certain port range and connect to terminal or auxiliary lines. Reverse telnetting has often been used to connect a Cisco IOS router that has many terminal lines to the consoles of other Cisco IOS routers or to other devices. Telnetting makes it easy to reach the router console from anywhere simply by telnetting to the terminal server on a specific line. This telnetting approach can be used to configure a router even if all network connnectivity to that router is disconnected. Reverse telnetting also allows modems that are attached to Cisco IOS routers to be used for dial-out (usually with a rotary device).

Reverse SSH

Reverse telnetting can be accomplished using SSH. Unlike reverse telnetting, SSH provides for secure connections. The Reverse SSH Enhancements feature provides you with a simplified method of configuring SSH. Using this feature, you no longer have to configure a separate line for every terminal or auxiliary line on which you want to enable SSH. The previous method of configuring reverse SSH limited the number of ports that can be accessed to 100. The Reverse SSH Enhancements feature removes the port number limitation. For information on the alternative method of configuring reverse SSH, see the section "How to Configure Reverse SSH Enhancements."

How to Configure Reverse SSH Enhancements

This section contains the following procedures:

- Configuring Reverse SSH for Console Access, page 1626
- Configuring Reverse SSH for Modem Access, page 1628
- Troubleshooting Reverse SSH on the Client, page 1630
- Troubleshooting Reverse SSH on the Server, page 1630

Configuring Reverse SSH for Console Access

To configure reverse SSH console access on the SSH server, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line** *line-number* [*ending-line-number*]
4. **no exec**
5. **login authentication** *listname*
6. **transport input ssh**
7. **exit**
8. **exit**
9. **ssh -l** *userid*:{*number*} {*ip-address*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**line** *line-number* [*ending-line-number*] **Example:** `Router# line 1 3`	Identifies a line for configuration and enters line configuration mode.
Step 4	**no exec** **Example:** `Router (config-line)# no exec`	Disables EXEC processing on a line.
Step 5	**login authentication** *listname* **Example:** `Router (config-line)#login authentication default`	Defines a login authentication mechanism for the lines. **Note** The authentication method must use a username and password.
Step 6	**transport input ssh** **Example:** `Router (config-line)# transport input ssh`	Defines which protocols to use to connect to a specific line of the router. • The **ssh** keyword must be used for the Reverse SSH Enhancements feature.
Step 7	**exit** **Example:** `Router (config-line)# exit`	Exits line configuration mode.

	Command or Action	Purpose
Step 8	**exit** **Example:** `Router (config)# exit`	Exits global configuration mode.
Step 9	**ssh -l** *userid*:{*number*} {*ip-address*} **Example:** `Router# ssh -l lab:1 router.example.com`	Specifies the user ID to use when logging in on the remote networking device that is running the SSH server. • *userid*—User ID. • **:**—Signifies that a port number and terminal IP address will follow the *userid* argument. • *number*—Terminal or auxiliary line number. • *ip-address*—Terminal server IP address. **Note** The *userid* argument and **:rotary**{*number*}{*ip-address*} delimiter and arguments are mandatory when configuring the alternative method of Reverse SSH for modem access.

Configuring Reverse SSH for Modem Access

To configure Reverse SSH for modem access, perform the steps shown in the "SUMMARY STEPS" section below.

In this configuration, reverse SSH is being configured on a modem used for dial-out lines. To get any of the dial-out modems, you can use any SSH client and start a SSH session as shown (in Step 10) to get to the next available modem from the rotary device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line** *line-number* [*ending-line-number*]
4. **no exec**
5. **login authentication** *listname*
6. **rotary** *group*
7. **transport input ssh**
8. **exit**
9. **exit**
10. **ssh -l** *userid*:**rotary**{*number*} {*ip-address*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**line** *line-number* [*ending-line-number*] **Example:** `Router# line 1 200`	Identifies a line for configuration and enters line configuration mode.
Step 4	**no exec** **Example:** `Router (config-line)# no exec`	Disables EXEC processing on a line.
Step 5	**login authentication** *listname* **Example:** `Router (config-line)# login authentication default`	Defines a login authentication mechanism for the lines. **Note** The authentication method must use a username and password.
Step 6	**rotary** *group* **Example:** `Router (config-line)# rotary 1`	Defines a group of lines consisting of one or more virtual terminal lines or one auxiliary port line.
Step 7	**transport input ssh** **Example:** `Router (config-line)# transport input ssh`	Defines which protocols to use to connect to a specific line of the router. • The **ssh** keyword must be used for the Reverse SSH Enhancements feature.
Step 8	**exit** **Example:** `Router (config-line)# exit`	Exits line configuration mode.
Step 9	**exit** **Example:** `Router (config)# exit`	Exits global configuration mode.

	Command or Action	Purpose
Step 10	**ssh -l** *userid*:**rotary**{*number*} {*ip-address*} **Example:** `Router# ssh -l lab:rotary1 router.example.com`	Specifies the user ID to use when logging in on the remote networking device that is running the SSH server. • *userid*—User ID. • **:**—Signifies that a port number and terminal IP address will follow the *userid* argument. • *number*—Terminal or auxiliary line number. • *ip-address*—Terminal server IP address. **Note** The *userid* argument and **:rotary**{*number*}{*ip-address*} delimiter and arguments are mandatory when configuring the alternative method of Reverse SSH for modem access.

Troubleshooting Reverse SSH on the Client

To troubleshoot the reverse SSH configuration on the client (remote device), perform the following steps.

SUMMARY STEPS

1. **enable**
2. **debug ip ssh client**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug ip ssh client** **Example:** `Router# debug ip ssh client`	Displays debugging messages for the SSH client.

Troubleshooting Reverse SSH on the Server

To troubleshoot the reverse SSH configuration on the terminal server, perform the following steps. The steps may be configured in any order or independent of one another.

SUMMARY STEPS

1. **enable**
2. **debug ip ssh**
3. **show ssh**
4. **show line**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug ip ssh** **Example:** `Router# debug ip ssh`	Displays debugging messages for the SSH server.
Step 3	**show ssh** **Example:** `Router# show ssh`	Displays the status of the SSH server connections.
Step 4	**show line** **Example:** `Router# show line`	Displays parameters of a terminal line.

Configuration Examples for Reverse SSH Enhancements

This section includes the following configuration examples:

- Reverse SSH Console Access: Example, page 1631
- Reverse SSH Modem Access: Example, page 1632

Reverse SSH Console Access: Example

The following configuration example shows that reverse SSH has been configured for console access for terminal lines 1 through 3:

Terminal Server Configuration

```
line 1 3
   no exec
   login authentication default
   transport input ssh
```

Client Configuration

The following commands configured on the SSH client will form the reverse SSH session with lines 1, 2, and 3, respectively:

```
ssh -l lab:1 router.example.com
ssh -l lab:2 router.example.com
ssh -l lab:3 router.example.com
```

Reverse SSH Modem Access: Example

The following configuration example shows that dial-out lines 1 through 200 have been grouped under rotary group 1 for modem access:

```
line 1 200
   no exec
   login authentication default
   rotary 1
   transport input ssh
   exit
```

The following command shows that reverse SSH will connect to the first free line in the rotary group:

```
ssh -l lab:rotary1 router.example.com
```

Additional References

The following sections provide references related to Reverse SSH Enhancements.

Related Documents

Related Topic	Document Title
Configuring Secure Shell	• "Configuring Secure Shell" chapter of *Cisco IOS Security Configuration Guide*, Release 12.3 • "Secure Shell Version 2 Support" feature guide, Release 12.3(7)T • *SSH Terminal-Line Access*, Release 12.2(2)T
Cisco IOS commands	*Cisco IOS Commands Master List*, Release 12.3T
Security commands	*Cisco IOS Security Command Reference*, Release 12.3T

Standards

Standards	Title
No new or modified standards are supported by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following modified command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **ssh**

Secure Copy

The Secure Copy (SCP) feature provides a secure and authenticated method for copying router configuration or router image files. SCP relies on Secure Shell (SSH), an application and a protocol that provide a secure replacement for the Berkeley r-tools.

Feature History for Secure Copy

Release	Modification
12.2(2)T	This feature was introduced.
12.0(21)S	This feature was integrated into Cisco IOS 12.0(21)S.
12.2(25)S	This feature was integrated into Cisco IOS 12.2(25)S.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for Secure Copy, page 1635
- Information About Secure Copy, page 1636
- How to Configure SCP, page 1636
- Configuration Examples for Secure Copy, page 1638
- Additional References, page 1639
- Command Reference, page 1640
- Glossary, page 1641

Prerequisites for Secure Copy

- Before enabling SCP, you must correctly configure SSH, authentication, and authorization on the router.

- Because SCP relies on SSH for its secure transport, the router must have an Rivest, Shamir, and Adelman (RSA) key pair.

Information About Secure Copy

To configure Secure Copy feature, you should understand the following concepts.

- How SCP Works, page 1636

How SCP Works

The behavior of SCP is similar to that of remote copy (rcp), which comes from the Berkeley r-tools suite, except that SCP relies on SSH for security. In addition, SCP requires that authentication, authorization, and accounting (AAA) authorization be configured so the router can determine whether the user has the correct privilege level.

SCP allows a user who has appropriate authorization to copy any file that exists in the Cisco IOS File System (IFS) to and from a router by using the **copy** command. An authorized administrator may also perform this action from a workstation.

How to Configure SCP

This section contains the following procedures:

- Configuring SCP, page 1636
- Verifying SCP, page 1637
- Troubleshooting SCP, page 1638

Configuring SCP

To enable and configure a Cisco router for SCP server-side functionality, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication login** {**default** | *list-name*} *method1* [*method2...*]
5. **aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} [*method1* [*method2...*]]
6. **username** *name* [**privilege** *level*] {**password** *encryption-type encrypted-password*}
7. **ip scp server enable**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router (config)# aaa new-model`	Sets AAA authentication at login.
Step 4	**aaa authentication login** {**default** | *list-name*} *method1* [*method2...*] **Example:** `Router (config)# aaa authentication login default group tacacs+`	Enables the AAA access control system.
Step 5	**aaa authorization** {**network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} [*method1* [*method2...*]] **Example:** `Router (config)# aaa authorization exec default group tacacs+`	Sets parameters that restrict user access to a network. **Note** The **exec** keyword runs authorization to determine if the user is allowed to run an EXEC shell; therefore, you must use it when you configure SCP.
Step 6	**username** *name* [**privilege** *level*] {**password** *encryption-type encrypted-password*} **Example:** `Router (config)# username superuser privilege 2 password 0 superpassword`	Establishes a username-based authentication system. **Note** You may skip this step if a network-based authentication mechanism—such as TACACS+ or RADIUS—has been configured.
Step 7	**ip scp server enable** **Example:** `Router (config)# ip scp server enable`	Enables SCP server-side functionality.

Verifying SCP

To verify SCP server-side functionality, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`show running-config` **Example:** `Router# show running-config`	Verifies the SCP server-side functionality.

Troubleshooting SCP

To troubleshoot SCP authentication problems, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **debug ip scp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`debug ip scp` **Example:** `Router# debug ip scp`	Troubleshoots SCP authentication problems.

Configuration Examples for Secure Copy

This section provides the following configuration examples:

- SCP Server-Side Configuration Using Local Authentication: Example, page 1639
- SCP Server-Side Configuration Using Network-Based Authentication: Example, page 1639

SCP Server-Side Configuration Using Local Authentication: Example

The following example shows how to configure the server-side functionality of SCP. This example uses a locally defined username and password.

```
! AAA authentication and authorization must be configured properly for SCP to work.
aaa new-model
aaa authentication login default local
aaa authorization exec default local
username tiger privilege 15 password 0 lab
! SSH must be configured and functioning properly.
ip ssh time-out 120
ip ssh authentication-retries 3
ip scp server enable
```

SCP Server-Side Configuration Using Network-Based Authentication: Example

The following example shows how to configure the server-side functionality of SCP using a network-based authentication mechanism:

```
! AAA authentication and authorization must be configured properly for SCP to work.
aaa new-model
aaa authentication login default group tacacs+
aaa authorization exec default group tacacs+
! SSH must be configured and functioning properly.
ip ssh time-out 120
ip ssh authentication-retries 3
ip scp server enable
```

Additional References

The following sections provide references related to Secure Copy.

Related Documents

Related Topic	Document Title
Secure Shell	• *Secure Shell Version 1 Support* • *Secure Shell Version 2 Support*
Authentication and authorization commands	*Cisco IOS Security Command Reference*, Release 12.3 T
Configuring authentication and authorization	"Authentication, Authorization, and Accounting (AAA)" section of Cisco IOS Security Configuration Guide, Release 12.3

Standards

Standards	Title
No new or modified standards are supported by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **debug ip scp**
- **ip scp server enable**

Glossary

AAA—authentication, authorization, and accounting. Framework of security services that provide the method for identifying users (authentication), for remote access control (authorization), and for collecting and sending security server information used for billing, auditing, and reporting (accounting).

rcp—remote copy. Relying on Remote Shell (Berkeley r-tools suite) for security, rcp copies files, such as router images and startup configurations, to and from routers.

SCP—secure copy. Relying on SSH for security, SCP support allows the secure and authenticated copying of anything that exists in the Cisco IOS File Systems. SCP is derived from rcp.

SSH—Secure Shell. Application and a protocol that provide a secure replacement for the Berkeley r-tools. The protocol secures the sessions using standard cryptographic mechanisms, and the application can be used similarly to the Berkeley rexec and rsh tools. SSH Version 1 is implemented in the Cisco IOS software.

Note Refer to *Internetworking Terms and Acronyms* for terms not included in this glossary.

Secure Shell Version 2 Support

The Secure Shell Version 2 Support feature allows you to configure Secure Shell (SSH) Version 2 (SSH Version 1 support was implemented in an earlier Cisco IOS software release). SSH runs on top of a reliable transport layer and provides strong authentication and encryption capabilities. Currently, the only reliable transport that is defined for SSH is TCP. SSH provides a means to securely access and securely execute commands on another computer over a network. The Secure Copy Protocol (SCP) feature that is provided with SSH also allows for the secure transfer of files.

Feature History for Secure Shell Version 2 Support

Release	Modification
12.3(4)T	This feature was introduced.
12.3(2)XE	This feature was incorporated into Cisco IOS Release 12.3(2)XE.
12.3(7)T	Support was added for the SSH Version 2 client.
12.2(25)S	This feature was integrated into Cisco IOS Release 12.2(25)S.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for Secure Shell Version 2 Support, page 1644
- Restrictions for Secure Shell Version 2 Support, page 1644
- Information About Secure Shell Version 2 Support, page 1644
- How to Configure Secure Shell Version 2 Support, page 1645
- Configuration Examples for Secure Shell Version 2 Support, page 1654
- Where to Go Next, page 1654
- Additional References, page 1655
- Command Reference, page 1656

Prerequisites for Secure Shell Version 2 Support

Prior to configuring SSH, perform the following task:

- Download the required image on your router. The SSH server requires you to have a k9 (Triple Data Encryption Standard [3DES]) software image from Cisco IOS Release 12.3(4)T or 12.2(25)S downloaded on your router.

Note The SSH Version 2 server is supported in Cisco IOS Release 12.3(4)T and 12.2(25)S; the SSH Version 2 client is supported beginning with Cisco IOS Release 12.3(7)T. (The SSH client runs both the SSH Version 1 and Version 2 protocol and is supported in both k8 and k9 images in Cisco IOS Release 12.3(4)T.)

For more information on downloading a software image, refer to *Cisco IOS Configuration Fundamentals and Network Management Configuration Guide*.

Restrictions for Secure Shell Version 2 Support

- Rivest, Shamir, and Adelman (RSA) user authentication is not supported in the SSH server or SSH client for Cisco IOS software.
- SSH servers and SSH clients are supported in 3DES software images.
- Execution Shell, remote command execution, and Secure Copy Protocol (SCP) are the only applications supported.
- Compression is not supported.
- The RSA key-pair size must be greater than or equal to 768.

Information About Secure Shell Version 2 Support

To configure SSH Version 2, you should understand the following concept:

- Secure Shell Version 2, page 1644

Secure Shell Version 2

The Secure Shell Version 2 Support feature allows you to configure SSH Version 2.

The configuration for the SSH Version 2 server is similar to the configuration for SSH Version 1. The command **ip ssh version** has been introduced so that you may define which version of SSH that you want to configure. If you do not configure this command, SSH by default runs in compatibility mode; that is, both SSH Version 1 and SSH Version 2 connections are honored.

Note SSH Version 1 is a protocol that has never been defined in a standard. If you do not want your router to fall back to the undefined protocol (Version 1), you should use the **ip ssh version** command and specify Version 2.

The **ip ssh rsa keypair-name** command was also introduced in Cisco IOS Release 12.3(4)T so that you can enable a SSH connection using RSA keys that you have configured. Previously, SSH was linked to the first RSA keys that were generated (that is, SSH was enabled when the first RSA key pair was generated). The behavior still exists, but by using the **ip ssh rsa keypair-name** command, you can overcome that behavior. If you configure the **ip ssh rsa keypair-name** command with a key-pair name, SSH is enabled if the key pair exists, or SSH will be enabled if the key pair is generated later. If you use this command to enable SSH, you are not forced to configure a host name and a domain name, which was required in SSH Version 1 of the Cisco IOS software.

How to Configure Secure Shell Version 2 Support

This section contains the following procedures:

- Configuring a Router for SSH Version 2 Using a Host Name and Domain Name, page 1645 (required)
- Configuring a Router for SSH Version 2 Using RSA Key Pairs, page 1646 (optional)
- Starting an Encrypted Session with a Remote Device, page 1648 (optional)
- Verifying the Status of the Secure Shell Connection Using the show ssh Command, page 1649 (optional)
- Verifying the Secure Shell Status Using the show ip ssh Command, page 1650 (optional)
- Monitoring and Maintaining Secure Shell Version 2, page 1651 (optional)

Configuring a Router for SSH Version 2 Using a Host Name and Domain Name

To configure your router for SSH Version 2 using a host name and domain name, perform the following steps. You may also configure SSH Version 2 by using the RSA key pair configuration (See Configuring a Router for SSH Version 2 Using RSA Key Pairs).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **hostname** *hostname*
4. **ip domain-name** *name*
5. **crypto key generate rsa**
6. **ip ssh** [**timeout** *seconds* | **authentication-retries** *integer*]
7. **ip ssh version** [**1** | *2*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**hostname** *hostname* **Example:** `Router (config)# hostname cisco 7200`	Configures a host name for your router.
Step 4	**ip domain-name** *name* **Example:** `Router (config)# ip domain-name cisco.com`	Configures a domain name for your router.
Step 5	**crypto key generate rsa** **Example:** `Router (config)# crypto key generate rsa`	Enables the SSH server for local and remote authentication.
Step 6	**ip ssh** [**timeout** *seconds* | **authentication-retries** *integer*] **Example:** `Router (config)# ip ssh timeout 120`	(Optional) Configures SSH control variables on your router.
Step 7	**ip ssh version** [**1** | **2**] **Example:** `Router (config)# ip ssh version 1`	(Optional) Specifies the version of SSH to be run on your router.

Configuring a Router for SSH Version 2 Using RSA Key Pairs

To enable SSH Version 2 without configuring a host name or domain name, perform the following steps. SSH Version 2 will be enabled if the key pair that you configure already exists or if it is generated later. You may also configure SSH Version 2 by using the host name and domain name configuration (See Configuring a Router for SSH Version 2 Using a Host Name and Domain Name).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ssh rsa keypair-name** *keypair-name*

4. **crypto key generate rsa usage-keys label** *key-label* **modulus** *modulus-size*
5. **ip ssh** [**timeout** *seconds* | **authentication-retries** *integer*]
6. **ip ssh version** [**1** | **2**]

DETAILED STEPS

Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip ssh rsa keypair-name** *keypair-name* **Example:** `Router (config)# ip ssh rsa keypair-name sshkeys`	Specifies which RSA keypair to use for SSH usage. **Note** A Cisco IOS router can have many RSA key pairs.
Step 4	**crypto key generate rsa usage-keys label** *key-label* **modulus** *modulus-size* **Example:** `Router (config)# crypto key generate rsa usage-keys label sshkeys modulus 768`	Enables the SSH server for local and remote authentication on the router. For SSH Version 2, the modulus size must be at least 768 bits. **Note** To delete the RSA key-pair, use the **crypto key zeroize rsa** command. After you have deleted the RSAcommand, you automatically disable the SSH server.
Step 5	**ip ssh** [**timeout** *seconds* | **authentication-retries** *integer*] **Example:** `Router (config)# ip ssh timeout 120`	Configures SSH control variables on your router.
Step 6	**ip ssh version** [**1** | **2**] **Example:** `Router (config)# ip ssh version 1`	Specifies the version of SSH to be run on a router.

Starting an Encrypted Session with a Remote Device

To start an encrypted session with a remote networking device, perform the following step. (You do not have to enable your router. SSH can be run in disabled mode.)

Note The device you wish to connect with must support a SSH server that has an encryption algorithm that is supported in Cisco IOS software.

SUMMARY STEPS

1. **ssh** [**-v** {**1** | **2**}] [**-c** {**3des** | **aes128-cbc** | **aes192-cbc** | **aes256-cbc**}] [**-m** {**hmac-md5** | **hmac-md5-96** | **hmac-sha1** | **hmac-sha1-96**}] [**l** *userid*] [**-o numberofpasswordprompts** *n*] [**-p** *port-num*] {*ip-addr* | *hostname*} [*command*]

DETAILED STEPS

Step 1	**ssh** [**-v** {**1** | **2**}] [**-c** {**3des** | **aes128-cbc** | **aes192-cbc** | **aes256-cbc**}] [**-m** {**hmac-md5** | **hmac-md5-96** | **hmac-sha1** | **hmac-sha1-96**}] [**-o numberofpasswordprompts** *n*] [**-p** *port-num*] {*ip-addr* | *hostname*} [*command*] **Example:** `Router# ssh -v 2 -c aes256-cbc -m hmac-sha1-96 -l user2 10.76.82.24` Or The above example adheres to the SSH Version 2 conventions. A more natural and common way to start a session is by linking the username with the hostname. For example, the following configuration example provides an end result that is identical to that of the above example: `Router# ssh -v 2 -c aes256-cbc -m hmac-sha1-96 user2@10.76.82.24`	Starts an encrypted session with a remote networking device.

Troubleshooting Tips

The **ip ssh version** command can be used for troubleshooting your SSH configuration. By changing versions, you can determine which SSH version has a problem.

Verifying the Status of the Secure Shell Connection Using the show ssh Command

To display the status of the SSH connection on your router, use the **show ssh** command.

SUMMARY STEPS

1. **enable**
2. **show ssh**

DETAILED STEPS

Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ssh** **Example:** `Router# show ssh`	Displays the status of SSH server connections.

Examples

The following output examples from the **show ssh** command display status about various SSH Version 1 and Version 2 connections.

Version 1 and Version 2 Connections

```
-------------------------------------------------------------------------
Router# show ssh

Connection      Version Encryption      State                   Username
 0              1.5     3DES            Session started         lab
Connection Version Mode Encryption  Hmac                State
Username
1          2.0     IN   aes128-cbc  hmac-md5     Session started       lab
1          2.0     OUT  aes128-cbc  hmac-md5     Session started       lab
-------------------------------------------------------------------------
```

Version 2 Connection with No Version 1

```
-------------------------------------------------------------------------
Router# show ssh

Connection Version Mode Encryption  Hmac                State
Username
1          2.0     IN   aes128-cbc  hmac-md5     Session started       lab
1          2.0     OUT  aes128-cbc  hmac-md5     Session started       lab
%No SSHv1 server connections running.
-------------------------------------------------------------------------
```

Version 1 Connection with No Version 2

```
------------------------------------------------------------------------
Router# show ssh

Connection      Version Encryption      State                   Username
 0              1.5     3DES            Session started         lab
%No SSHv2 server connections running.
------------------------------------------------------------------------
```

Verifying the Secure Shell Status Using the show ip ssh Command

To verify your SSH configuration, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show ip ssh**

DETAILED STEPS

Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip ssh** **Example:** `Router# show ip ssh`	Displays the version and configuration data for SSH.

Examples

The following examples from the **show ip ssh** command display the version of SSH that is enabled, the authentication timeout values, and the number of authentication retries.

Version 1 and Version 2 Connections

```
------------------------------------------------------------------------
router# show ip ssh

3d06h: %SYS-5-CONFIG_I: Configured from console by consoleh
SSH Enabled - version 1.99
Authentication timeout: 120 secs; Authentication retries: 3
------------------------------------------------------------------------
```

Version 2 Connection with No Version 1

```
------------------------------------------------------------------------
Router# show ip ssh

SSH Enabled - version 2.0
Authentication timeout: 120 secs; Authentication retries: 3
------------------------------------------------------------------------
```

Version 1 Connection with No Version 2

```
-----------------------------------------------------------------------
Router# show ip ssh

3d06h: %SYS-5-CONFIG_I: Configured from console by console
SSH Enabled - version 1.5
Authentication timeout: 120 secs; Authentication retries: 3
-----------------------------------------------------------------------
```

Monitoring and Maintaining Secure Shell Version 2

To display debug messages about the SSH connections, use the **debug ip ssh** command.

SUMMARY STEPS

1. **enable**
2. **debug ip ssh**

DETAILED STEPS

Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug ip ssh** **Example:** `Router# debug ip ssh`	Displays debugging messages for SSH.

Example

The following output from the **debug ip ssh** command shows that the digit 2 keyword has been assigned, signifying that it is an SSH Version 2 connection.

```
Router# debug ip ssh

00:33:55: SSH1: starting SSH control process
00:33:55: SSH1: sent protocol version id SSH-1.99-Cisco-1.25
00:33:55: SSH1: protocol version id is - SSH-2.0-OpenSSH_2.5.2p2
00:33:55: SSH2 1: send: len 280 (includes padlen 4)
00:33:55: SSH2 1: SSH2_MSG_KEXINIT sent
00:33:55: SSH2 1: ssh_receive: 536 bytes received
00:33:55: SSH2 1: input: packet len 632
00:33:55: SSH2 1: partial packet 8, need 624, maclen 0
00:33:55: SSH2 1: ssh_receive: 96 bytes received
00:33:55: SSH2 1: partial packet 8, need 624, maclen 0
00:33:55: SSH2 1: input: padlen 11
00:33:55: SSH2 1: received packet type 20
00:33:55: SSH2 1: SSH2_MSG_KEXINIT received
00:33:55: SSH2: kex: client->server aes128-cbc hmac-md5 none
00:33:55: SSH2: kex: server->client aes128-cbc hmac-md5 none
00:33:55: SSH2 1: expecting SSH2_MSG_KEXDH_INIT
00:33:55: SSH2 1: ssh_receive: 144 bytes received
```

```
00:33:55: SSH2 1: input: packet len 144
00:33:55: SSH2 1: partial packet 8, need 136, maclen 0
00:33:55: SSH2 1: input: padlen 5
00:33:55: SSH2 1: received packet type 30
00:33:55: SSH2 1: SSH2_MSG_KEXDH_INIT received
00:33:55: SSH2 1: signature length 111
00:33:55: SSH2 1: send: len 384 (includes padlen 7)
00:33:55: SSH2: kex_derive_keys complete
00:33:55: SSH2 1: send: len 16 (includes padlen 10)
00:33:55: SSH2 1: newkeys: mode 1
00:33:55: SSH2 1: SSH2_MSG_NEWKEYS sent
00:33:55: SSH2 1: waiting for SSH2_MSG_NEWKEYS
00:33:55: SSH2 1: ssh_receive: 16 bytes received
00:33:55: SSH2 1: input: packet len 16
00:33:55: SSH2 1: partial packet 8, need 8, maclen 0
00:33:55: SSH2 1: input: padlen 10
00:33:55: SSH2 1: newkeys: mode 0
00:33:55: SSH2 1: received packet type 2100:33:55: SSH2 1: SSH2_MSG_NEWKEYS received
00:33:56: SSH2 1: ssh_receive: 48 bytes received
00:33:56: SSH2 1: input: packet len 32
00:33:56: SSH2 1: partial packet 16, need 16, maclen 16
00:33:56: SSH2 1: MAC #3 ok
00:33:56: SSH2 1: input: padlen 10
00:33:56: SSH2 1: received packet type 5
00:33:56: SSH2 1: send: len 32 (includes padlen 10)
00:33:56: SSH2 1: done calc MAC out #3
00:33:56: SSH2 1: ssh_receive: 64 bytes received
00:33:56: SSH2 1: input: packet len 48
00:33:56: SSH2 1: partial packet 16, need 32, maclen 16
00:33:56: SSH2 1: MAC #4 ok
00:33:56: SSH2 1: input: padlen 9
00:33:56: SSH2 1: received packet type 50
00:33:56: SSH2 1: send: len 32 (includes padlen 13)
00:33:56: SSH2 1: done calc MAC out #4
00:34:04: SSH2 1: ssh_receive: 160 bytes received
00:34:04: SSH2 1: input: packet len 64
00:34:04: SSH2 1: partial packet 16, need 48, maclen 16
00:34:04: SSH2 1: MAC #5 ok
00:34:04: SSH2 1: input: padlen 13
00:34:04: SSH2 1: received packet type 50
00:34:04: SSH2 1: send: len 16 (includes padlen 10)
00:34:04: SSH2 1: done calc MAC out #5
00:34:04: SSH2 1: authentication successful for lab
00:34:04: SSH2 1: input: packet len 64
00:34:04: SSH2 1: partial packet 16, need 48, maclen 16
00:34:04: SSH2 1: MAC #6 ok
00:34:04: SSH2 1: input: padlen 6
00:34:04: SSH2 1: received packet type 2
00:34:04: SSH2 1: ssh_receive: 64 bytes received
00:34:04: SSH2 1: input: packet len 48
00:34:04: SSH2 1: partial packet 16, need 32, maclen 16
00:34:04: SSH2 1: MAC #7 ok
00:34:04: SSH2 1: input: padlen 19
00:34:04: SSH2 1: received packet type 90
00:34:04: SSH2 1: channel open request
00:34:04: SSH2 1: send: len 32 (includes padlen 10)
00:34:04: SSH2 1: done calc MAC out #6
00:34:04: SSH2 1: ssh_receive: 192 bytes received
00:34:04: SSH2 1: input: packet len 64
00:34:04: SSH2 1: partial packet 16, need 48, maclen 16
00:34:04: SSH2 1: MAC #8 ok
00:34:04: SSH2 1: input: padlen 13
00:34:04: SSH2 1: received packet type 98
00:34:04: SSH2 1: pty-req request
```

```
00:34:04: SSH2 1: setting TTY - requested: height 24, width 80; set: height 24,
width 80
00:34:04: SSH2 1: input: packet len 96
00:34:04: SSH2 1: partial packet 16, need 80, maclen 16
00:34:04: SSH2 1: MAC #9 ok
00:34:04: SSH2 1: input: padlen 11
00:34:04: SSH2 1: received packet type 98
00:34:04: SSH2 1: x11-req request
00:34:04: SSH2 1: ssh_receive: 48 bytes received
00:34:04: SSH2 1: input: packet len 32
00:34:04: SSH2 1: partial packet 16, need 16, maclen 16
00:34:04: SSH2 1: MAC #10 ok
00:34:04: SSH2 1: input: padlen 12
00:34:04: SSH2 1: received packet type 98
00:34:04: SSH2 1: shell request
00:34:04: SSH2 1: shell message received
00:34:04: SSH2 1: starting shell for vty
00:34:04: SSH2 1: send: len 48 (includes padlen 18)
00:34:04: SSH2 1: done calc MAC out #7
00:34:07: SSH2 1: ssh_receive: 48 bytes received
00:34:07: SSH2 1: input: packet len 32
00:34:07: SSH2 1: partial packet 16, need 16, maclen 16
00:34:07: SSH2 1: MAC #11 ok
00:34:07: SSH2 1: input: padlen 17
00:34:07: SSH2 1: received packet type 94
00:34:07: SSH2 1: send: len 32 (includes padlen 17)
00:34:07: SSH2 1: done calc MAC out #8
00:34:07: SSH2 1: ssh_receive: 48 bytes received
00:34:07: SSH2 1: input: packet len 32
00:34:07: SSH2 1: partial packet 16, need 16, maclen 16
00:34:07: SSH2 1: MAC #12 ok
00:34:07: SSH2 1: input: padlen 17
00:34:07: SSH2 1: received packet type 94
00:34:07: SSH2 1: send: len 32 (includes padlen 17)
00:34:07: SSH2 1: done calc MAC out #9
00:34:07: SSH2 1: ssh_receive: 48 bytes received
00:34:07: SSH2 1: input: packet len 32
00:34:07: SSH2 1: partial packet 16, need 16, maclen 16
00:34:07: SSH2 1: MAC #13 ok
00:34:07: SSH2 1: input: padlen 17
00:34:07: SSH2 1: received packet type 94
00:34:07: SSH2 1: send: len 32 (includes padlen 17)
00:34:07: SSH2 1: done calc MAC out #10
00:34:08: SSH2 1: ssh_receive: 48 bytes received
00:34:08: SSH2 1: input: packet len 32
00:34:08: SSH2 1: partial packet 16, need 16, maclen 16
00:34:08: SSH2 1: MAC #14 ok
00:34:08: SSH2 1: input: padlen 17
00:34:08: SSH2 1: received packet type 94
00:34:08: SSH2 1: send: len 32 (includes padlen 17)
00:34:08: SSH2 1: done calc MAC out #11
00:34:08: SSH2 1: ssh_receive: 48 bytes received
00:34:08: SSH2 1: input: packet len 32
00:34:08: SSH2 1: partial packet 16, need 16, maclen 16
00:34:08: SSH2 1: MAC #15 ok
00:34:08: SSH2 1: input: padlen 17
00:34:08: SSH2 1: received packet type 94
00:34:08: SSH2 1: send: len 32 (includes padlen 16)
00:34:08: SSH2 1: done calc MAC out #12
00:34:08: SSH2 1: send: len 48 (includes padlen 18)
00:34:08: SSH2 1: done calc MAC out #13
00:34:08: SSH2 1: send: len 16 (includes padlen 6)
00:34:08: SSH2 1: done calc MAC out #14
00:34:08: SSH2 1: send: len 16 (includes padlen 6)
```

```
00:34:08: SSH2 1: done calc MAC out #15
00:34:08: SSH1: Session terminated normally
```

Configuration Examples for Secure Shell Version 2 Support

This section provides the following configuration examples:

- Configuring Secure Shell Version 1: Example, page 1654
- Configuring Secure Shell Version 2: Example, page 1654
- Configuring Secure Shell Versions 1 and 2: Example, page 1654
- Starting an Encrypted Session with a Remote Device: Example, page 1654

Configuring Secure Shell Version 1: Example

```
Router# configure terminal
Router (config)# ip ssh version 1
c7200-25-2013(config)# end
```

Configuring Secure Shell Version 2: Example

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip ssh version 2
Router(config)# end
```

Configuring Secure Shell Versions 1 and 2: Example

```
Router# configure terminal
Router (config)# no ip ssh version
Router (config)# end
```

Starting an Encrypted Session with a Remote Device: Example

```
Router# ssh -v 2 -c aes256-cbc -m hmac-sha1-160 -l shaship 10.76.82.24
```

Where to Go Next

You have to use a SSH remote device that supports SSH Version 2, and you have to connect to a Cisco IOS router.

Additional References

The following sections provide references related to Secure Shell Version 2.

Related Documents

Related Topic	Document Title
Configuring Secure Shell	"Configuring Secure Shell" chapter of *Cisco IOS Security Configuration Guide*
AAA	"Authentication, Authorization, and Accounting (AAA)" section of *Cisco IOS Security Configuration Guide*
IPSec	"IP Security and Encryption" section of *Cisco IOS Security configuration Guide*
IOS configuration fundamentals	*Cisco IOS Configuration Fundamentals and Network Management Configuration Guide* and *Cisco IOS Configuration Fundamentals and Network Management Command Reference*
Security commands	*Cisco IOS Security Command Reference*, Release 12.3 T
Downloading a Cisco software image	*Cisco IOS Configuration Fundamentals and Network Management Configuration Guide*
Configuring a host name and host domain	"Configuring Secure Shell" chapter in the *Cisco IOS Security Configuration Guide*

Standards

Standards	Title
Internet Engineering Task Force (IETF) Secure Shell Version 2 Draft Standards	Internet Engineering Task Force website

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **ip ssh rsa keypair-name**
- **ip ssh version**
- **ssh**

SSH Terminal-Line Access

This feature module describes the SSH Terminal-Line Access feature and includes the following sections:

- Feature Overview, page 1657
- Supported Platforms, page 1658
- Supported Standards, MIBs, and RFCs, page 1659
- Prerequisites, page 1659
- Configuration Tasks, page 1659
- Configuration Examples, page 1661
- Command Reference, page 1662

Feature Overview

Cisco IOS supports reverse Telnet, which allows users to Telnet through the router—via a certain port range—to connect them to tty (asynchronous) lines. Reverse Telnet has allowed users to connect to the console ports of remote devices that do not natively support Telnet. However, this method has provided very little security because all Telnet traffic goes over the network in the clear. The SSH Terminal-Line Access feature replaces reverse Telnet with secure shell (SSH). This feature may be configured to use encryption to access devices on the tty lines, which provide users with connections that support strong privacy and session integrity.

SSH is an application and a protocol that provide secure replacement for the suite of Berkeley r-tools such as rsh, rlogin and rcp. (Cisco IOS supports rlogin.) The protocol secures the sessions using standard cryptographic mechanisms, and the application can be used similarly to the Berkeley rexec and rsh tools. There are currently two versions of SSH available: SSH Version 1 and SSH Version 2. Only SSH Version 1 is implemented in the Cisco IOS software.

The SSH Terminal-Line Access feature enables users to configure their router with secure access and perform the following tasks:

- Connect to a router that has multiple terminal lines connected to consoles or serial ports of other routers, switches, or devices.
- Simplify connectivity to a router from anywhere by securely connecting to the terminal server on a specific line.
- Allow modems attached to routers to be used for dial-out securely.
- Require authentication to each of the lines through a locally defined username and password, TACACS+, or RADIUS.

Benefits

The SSH Terminal-Line Access feature provides users secure access to tty lines.

Restrictions

Console Server Requirement

To configure secure console server access, you must define each line in its own rotary and configure SSH to use SSH over the network when users wish to access each of those devices.

Memory and Performance Impact

Replacing reverse Telnet with SSH may reduce the performance of available tty lines due to the addition of encryption and decryption processing above the vty processing. (Any cryptographic mechanism uses more memory than a regular access.)

Related Documents

The following documents provide information related to the SSH Terminal-Line Access feature:

- *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2
- *Cisco IOS Dial Technologies Command Reference*, Release 12.2
- *Cisco IOS Security Command Reference*, Release 12.2
- *Cisco IOS Security Configuration Guide*, Release 12.2

For more information on SSH, such as the details of the protocol, go to the SSH website at http://www.ssh.com/.

Supported Platforms

This feature is supported on the following platforms:

- Cisco 1700 Series
- Cisco 2600 Series
- Cisco 3620
- Cisco 3640
- Cisco 3660
- Cisco 4500
- Cisco 12000 Series

This feature is supported on all platforms that support SSH.

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

No new or modified MIBs are supported by this feature.

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml.

RFCs

No new or modified RFCs are supported by this feature.

Prerequisites

Download the required image on your router. The SSH server requires you to have an IPSec (DES or 3DES) encryption software image from Cisco IOS Release 12.1(1)T downloaded on your router; the SSH client requires you to have an IPSec (DES or 3DES) encryption software image from Cisco IOS Release 12.1(3)T downloaded on your router. For more information on downloading a software image, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.2.

The SSH server requires the use of a username and password, which must be defined through the use of a local username and password, TACACS+, or RADIUS.

Note The SSH Terminal-Line Access feature is available on any image that contains SSH.

Configuration Tasks

See the following section for configuration tasks for the SSH Terminal-Line Access feature:

- Configuring SSH Terminal-Line Access

Configuring SSH Terminal-Line Access

Note SSH must already be configured on the router.

To configure a Cisco router to support reverse secure Telnet, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# line line-number [ending-line-number]`	Identifies a line for configuration and enters line configuration mode. **Note** For router console configurations, each line must be defined in its own rotary, and SSH must be configured to listen in on each rotary. **Note** An authentication method requiring a username and password must be configured for each line. This may be done through the use of a local username and password stored on the router, through the use of TACACS+, or through the use of RADIUS. Neither Line passwords nor the enable password are sufficient to be used with SSH.
Step 2	`Router(config-line)# no exec`	Disables exec processing on each of the lines.
Step 3	`Router(config-line)# login {local | authentication listname}`	Defines a login authentication mechanism for the lines. **Note** The authentication method must utilize a username and password.
Step 4	`Router(config-line)# rotary group`	Defines a group of lines consisting or one or more lines. **Note** All rotaries used must be defined, and each defined rotary must be used when SSH is enabled.
Step 5	`Router(config-line)# transport input {all | ssh}`	Defines which protocols to use to connect to a specific line of the router.
Step 6	`Router(config-line)# exit`	Exits line configuration mode.
Step 7	`Router(config)# ip ssh port portnum rotary group`	Enables secure network access to the tty lines. Use this command to connect the *portnum* argument with the rotary *group* argument, which is associated with a line or group of lines. **Note** The *group* argument must correspond with the **rotary** *group* number chosen in Step 4.

Verifying SSH Terminal-Line Access

To verify that this functionality is working, you can connect to a router using an SSH client.

Configuration Examples

This section provides the following configuration examples:

- SSH Terminal-Line Access Configuration Example
- SSH Terminal-Line Access for a Console (Serial Line) Ports Configuration Example

SSH Terminal-Line Access Configuration Example

The following example shows how to configure the SSH Terminal-Line Access feature on a modem used for dial-out on lines 1 through 200. To get any of the dial-out modems, use any SSH client and start a SSH session to port 2000 of the router to get to the next available modem from the rotary.

```
line 1 200
   no exec
   login authentication default
   rotary 1
   transport input ssh
   exit
ip ssh port 2000 rotary 1
```

SSH Terminal-Line Access for a Console (Serial Line) Ports Configuration Example

The following example shows how to configure the SSH Terminal-Line Access feature to access the console or serial line interface of various devices. For this type of access, each line is put into its own rotary, and each rotary is used for a single port. In this example, lines 1 through 3 are used; the port (line) mappings of the configuration are shown in Table 68.

Table 68 Port (line) Configuration Mappings

Line Number	SSH Port Number
1	2001
2	2002
3	2003

```
line 1
   no exec
   login authentication default
   rotary 1
   transport input ssh
line 2
   no exec
   login authentication default
   rotary 2
   transport input ssh
```

```
line 3
   no exec
   login authentication default
   rotary 3
   transport input ssh

ip ssh port 2001 rotary 1 3
```

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **ip ssh port**

802.1X Authentication Services

This part consists of the following:

- Remote Site IEEE 802.1X Local Authentication Service
- VPN Access Control Using 802.1X Authentication

Remote Site IEEE 802.1X Local Authentication Service

The Remote Site IEEE 802.1X Local Authentication Service feature provides the ability to configure an access point or wireless-aware router to act as a local RADIUS server. Configuring local authentication service provides a backup authentication service in the event of a WAN link or server failure.

Feature History for the Remote Site IEEE 802.1X Local Authentication Service Feature

Release	Modification
12.2(11)JA	This feature was introduced on the Cisco IOS Release 12.2(11)JA on Cisco Aironet access points.
12.3(11)T	This feature was integrated into the Cisco IOS Release 12.3(11)T on the Cisco 2600XM, Cisco 2691, Cisco 2811, Cisco 2821, Cisco 2851, Cisco 3700 series, and Cisco 3800 series routers.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for Configuring Remote Site IEEE 802.1X Local Authentication Service, page 1666
- Restrictions for Configuring Remote Site IEEE 802.1X Local Authentication Service, page 1666
- Information About Configuring Remote Site IEEE 802.1x Local Authentication Service, page 1666
- How to Configure Remote Site IEEE 802.1X Local Authentication Service, page 1668
- Monitoring and Maintaining 802.1X Local Authentication Service, page 1674
- Configuration Examples for Remote Site IEEE 802.1X Local Authentication Service, page 1674
- Additional References, page 1678
- Command Reference, page 1679

Prerequisites for Configuring Remote Site IEEE 802.1X Local Authentication Service

Follow these guidelines when you configure an access point or wireless-aware router as a local authentication server:

- To prevent performance degradation, configure local authentication service on an access point or a wireless-aware router that does not have a high CPU load.
- Physically secure the access point or router to protect its configuration.

Restrictions for Configuring Remote Site IEEE 802.1X Local Authentication Service

The following are restrictions of the local authentication service feature:

- The local authentication server does not synchronize its database with the main RADIUS servers. It is necessary to manually configure the local authentication server with client usernames and passwords.
- LEAP is the only supported authentication protocol.
- Although multiple local authentication servers can exist on one network, only one authentication server can be configured on any single device.

Information About Configuring Remote Site IEEE 802.1x Local Authentication Service

On typical wireless LANs that use 802.1X authentication, access points and wireless-aware routers rely on remote site RADIUS servers to authenticate client devices. This authentication traffic must cross a WAN link. If the WAN link fails, or if the access points and routers cannot reach the RADIUS servers, then the client devices cannot access the wireless network even if their requirements for access are strictly local.

To provide for local authentication service or backup authentication service in the event of a WAN link or server failure, you can configure an access point or wireless-aware router to act as a local RADIUS server. The access point or wireless-aware router can authenticate Light Extensible Authentication Protocol (LEAP)–enabled wireless client devices and allow them to join your network.

Because the local authentication device does not synchronize its database with the main RADIUS servers. You must configure the local authentication server with client usernames and passwords. The local authentication server also permits you to specify a VLAN and a list of service set identifiers (SSIDs) that a client is allowed to use.

Table 69 shows the maximum number of clients that can be configured on a local authentication server.

Table 69 Maximum Number of Clients That Can be Configured on a Local Authentication Server

Local Authentication Server	Maximum Number of Clients
Cisco Aironet Access Point 1100 and Cisco Aironet Access Point 1200	50
Cisco 2610XM, Cisco 2611XM routers	50
Cisco 2620XM, Cisco 2621XM routers	50
Cisco 2650XM, Cisco 2651XM routers	50
Cisco 2691 routers	100
Cisco 2811 routers	100
Cisco 2821 routers	100
Cisco 2851 routers	200
Cisco 3725 routers	250
Cisco 3745 routers	500
Cisco 3825 routers	500
Cisco 3845 routers	1000

Note Users that are associated to the local authentication server might notice a drop in performance during authentication of client devices. However, if your wireless LAN contains only one access point, you can configure that device as both the 802.1X authenticator and the local authentication server.

You configure access points and routers to use the local authentication server when they cannot reach the main servers or when a RADIUS server is not available.

The access points and wireless-aware routers stop using the local authentication server automatically when the link to the main servers is restored.

If your local authentication server also serves client devices, you must enter the local authentication server access point or router as a network access server (NAS). When a LEAP client associates to the local authentication server access point, the access point uses itself to authenticate the client.

Caution The access point or wireless-aware router that you use as an authentication server contains detailed authentication information about your wireless LAN, so you should secure it physically to protect its configuration.

How to Configure Remote Site IEEE 802.1X Local Authentication Service

This section contains the following procedures:

- Configuring the Local Authentication Server, page 1668 (required)
- Configuring User Groups on the Local Authentication Server, page 1669 (optional)
- Creating the User List on the Local Authentication Server, page 1670 (required)
- Saving the Configuration on the Local Authentication Server, page 1671 (optional)
- Configuring Access Points or Routers to Use the Local Authentication Server, page 1671 (required)

Configuring the Local Authentication Server

Perform this task to configure the access point as a local authentication server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **radius-server local**
5. **nas** *ip-address* **key** *shared-key*

DETAILED STEPS

	Command	Purpose
Step 1	`Router> enable`	Enables privileged EXEC mode.
Step 2	`Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# aaa new-model`	Enables AAA.

	Command	Purpose
Step 4	`Router(config)# radius-server local`	Enables the access point or router as a local authentication server and enters configuration mode for the authentication server.
Step 5	`Router(config-radsrv)# nas` *ip-address* **key** *shared-key*	Adds an access point or wireless domain services (WDS) device to the list of units that use the local authentication server. Enter the IP address of the access point or WDS device, and the shared key used to authenticate communication between the local authentication server and other access points. You must enter this shared key on the WDS devices that use the local authentication server. Each access point and candidate WDS that uses the local authentication server is a network access server (NAS). If an access point is the local authentication server that also serves client devices, you must enter the local authentication server access point as a NAS. **Note** Leading spaces in the key string are ignored, but spaces within and at the end of the key are used. If you use spaces in your key, do not enclose the key in quotation marks unless the quotation marks are part of the key. Repeat this step to add each access point and candidate WDS device that uses the local authentication server.

Configuring User Groups on the Local Authentication Server

Perform this optional task (beginning in local RADIUS server configuration mode) to configure user groups on the local authentication server.

Note If you do not wish to configure user groups on the local authentication server, skip this task and go to the "Creating the User List on the Local Authentication Server" section on page 1670.

SUMMARY STEPS

1. **group** *group-name*
2. **vlan** *vlan*
3. **ssid** *ssid*
4. **reauthentication time** *seconds*
5. **block count** *count* **time** {*seconds* | **infinite**}
6. **exit**

DETAILED STEPS

	Command	Purpose
Step 1	`Router(config-radsrv)# group group-name`	Enters user group configuration mode and configures a user group to which you can assign shared settings.
Step 2	`Router(config-radsrv-group)# vlan vlan`	(Optional) Specifies a VLAN to be used by members of the user group. The access point moves group members into that VLAN, overriding other VLAN assignments. You can assign only one VLAN to the group.
Step 3	`Router(config-radsrv-group)# ssid ssid`	(Optional) Enters up to 20 service set identifiers (SSIDs) to limit members of the user group to those SSIDs. The access point checks whether the client's SSID matches an SSID in the list. If the SSID does not match, the client is disassociated.
Step 4	`Router(config-radsrv-group)# reauthentication time seconds`	(Optional) Configures the number of seconds after which access points should reauthenticate members of the group. The reauthentication provides users with a new encryption key. The default setting is 0, which means that group members are never required to reauthenticate.
Step 5	`Router(config-radsrv-group)# block count count time {seconds | infinite}`	(Optional) To help protect against password-guessing attacks, you can lock out group members for a length of time after a set number of incorrect passwords. • Count—The number of failed passwords that triggers a lockout of the username. • Time—The number of seconds that the lockout should last. If you enter **infinite**, an administrator must manually unblock the locked username. For more information, see the "Unblocking Usernames" section on page 1670.
Step 6	`Router(config-radsrv-group)# exit`	Returns to authenticator configuration mode.

Unblocking Usernames

You can unblock usernames before the lockout time expires or when the lockout time is set to infinite. To unblock a locked username, enter the following command in privileged EXEC mode on the local authentication server.

```
Router# clear radius local-server user username
```

Creating the User List on the Local Authentication Server

Perform the required task described in the following paragraphs to create a user list on the local authentication server and to configure the users that are allowed to authenticate using the local authentication server.

Note If you do not wish to configure users on the local authentication server, skip this task and go to the "Saving the Configuration on the Local Authentication Server" section on page 1671.

You must enter a username and password for each user. If you know only the NT hash value of the password, which you can often find in the authentication server database, you can enter the NT hash as a string of hexadecimal digits.

To add the user to a user group, enter the group name. If you do not specify a group, the user is not assigned to a specific VLAN and is never forced to reauthenticate.

Beginning in local RADIUS server configuration mode, enter the **user** command for each username:

```
Router(config-radsrv)# user username {password | nthash} password [group group-name]
```

Saving the Configuration on the Local Authentication Server

Perform this optional task to save the current configuration.

SUMMARY STEPS

1. **end**
2. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	`Router(config-radsrv)# end`	Returns to privileged EXEC mode.
Step 2	`Router# copy running-config startup-config`	Saves your entries in the configuration file.

Configuring Access Points or Routers to Use the Local Authentication Server

Perform this required task to add the local authentication server to the list of servers on the client access point or wireless-aware router.

Note If your local authentication server access point also serves client devices, you must configure the local authentication server to use itself to authenticate client devices.

On the wireless devices that use the local authentication server, use the **radius-server host** command in privileged EXEC mode to enter the local authentication server as a RADIUS server. The order in which the devices attempt to use the servers matches the order in which you enter the servers in the device configuration. If you are configuring the device to use a RADIUS server for the first time, enter the main RADIUS servers first, and enter the local authentication server last.

Note You must enter **1812** as the authentication port and **1813** as the accounting port. The local authentication server listens on User Datagram Protocol (UDP) port 1813 for RADIUS accounting packets. It discards the accounting packets but sends acknowledge packets back to the RADIUS clients to prevent the clients from reacting as though the server is down.

Use the **radius-server deadtime** command in global configuration mode to set an interval during which the access point or router does not attempt to use servers that do not respond, thus avoiding the wait for a request to time out before trying the next configured server. A server marked as dead is skipped by additional requests for the duration of minutes that you specify, up to 1440 (24 hours).

To remove the local authentication server from the access point or router configuration, use the **no radius-server host** command in global configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **radius-server host** {*hostname* | *ip-address*} [**auth-port** *port-number*] [**acct-port** *port-number*] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string*]
5. **aaa group server** {**radius** | **tacacs+**} *group-name*
6. **server ip-address auth-port 1812 acct-port 1813**
7. **aaa authentication login** *named-authentication-list*
8. **end**
9. **show running-config**
10. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	`Router> enable`	Enables privileged EXEC mode.
Step 2	`Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# aaa new-model`	Enables authentication, authorization, and accounting (AAA). This step must be configured before the rest of the AAA configuration steps.

	Command	Purpose
Step 4	`Router(config)# radius-server host {hostname | ip-address} [auth-port port-number] [acct-port port-number] [timeout seconds] [retransmit retries] [key string]`	Specifies the IP address or hostname of the remote RADIUS server host. • (Optional) For **auth-port** *port-number*, specify the UDP destination port for authentication requests. • (Optional) For **acct-port** *port-number*, specify the UDP destination port for accounting requests. • (Optional) For **timeout** *seconds*, specify the time interval that the access point waits for the RADIUS server to reply before retransmitting. The range is 1 to 1000. This setting overrides the setting made using the **radius-server timeout** command in global configuration mode. If no timeout is set with the **radius-server host** command, the setting made using the **radius-server timeout** command is used. • (Optional) For **retransmit** *retries*, specify the number of times that a RADIUS request is re-sent to a server if that server is not responding or is responding slowly. The range is 1 to 1000. If no retransmit value is set using the **radius-server host** command, the setting made using the **radius-server retransmit** command in global configuration command mode is used. • (Optional) For **key** *string*, specify the authentication and encryption key used between the access point and the RADIUS daemon running on the RADIUS server. **Note** The key is a text string that must match the encryption key used on the RADIUS server. Always configure the key as the last item in the **radius-server host** command. Leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in your key, do not enclose the key in quotation marks unless the quotation marks are part of the key. To configure the access point to recognize more than one host entry associated with a single IP address, enter this command as many times as necessary, making sure to use a different UDP port number for each host. The access point software searches for hosts in the order in which you specify them. Set the timeout, retransmit, and encryption key values to use with the specific RADIUS host.
Step 5	`aaa group server {radius | tacacs+} group-name`	Defines the AAA server-group with a group name.
Step 6	`Router(config-sg-radius)# server ip-address auth-port 1812 acct-port 1813`	Defines the AAA server IP address, authentication port, and accounting port.
Step 7	`Router(config)# aaa authentication login named-authentication-list`	Creates an authentication method list for the server group.
Step 8	`Router(config)# end`	Returns to privileged EXEC mode.
Step 9	`Router# show running-config`	Displays the current configuration for your verification.
Step 10	`Router# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Verifying the Configuration for Local Authentication Service

Use the **show running-config** command in global configuration mode to verify the current configuration for local authentication service.

SUMMARY STEPS

1. **enable**
2. **show running-config**

DETAILED STEPS

	Command	Purpose
Step 1	`Router> enable`	Enables privileged EXEC mode.
Step 2	`Router# show running-config`	Displays the current access point operating configuration

Monitoring and Maintaining 802.1X Local Authentication Service

To view statistics collected by the local authentication server, enter the following command in privileged EXEC mode:

```
Router# show radius local-server statistics
```

To reset local authentication server statistics to zero, enter the following command in privileged EXEC mode:

```
Router# clear radius local-server statistics
```

Configuration Examples for Remote Site IEEE 802.1X Local Authentication Service

This section provides the following configuration examples:

- Setting Up a Local Authentication Server: Example
- Setting Up Two Main Servers and a Local Authentication Server: Example
- Displaying Local Authentication Server Configuration: Example
- Displaying Local Authentication Server Statistics: Example

Setting Up a Local Authentication Server: Example

This example shows how to set up a local authentication server used by three access points with three user groups and several users:

```
AP# configure terminal
AP(config)# aaa new-model
```

```
AP(config)# aaa group server radius RADIUS_SERVER_GROUP
AP(config-sg-radius)# server 10.0.0.1 auth-port 1812 acct-port 1813
AP(config)# aaa authentication login RADIUS_METHOD_LIST
AP(config)# radius-server host 10.0.0.1 auth-port 1812 acct-port 1813 key 110337
AP(config)# radius-server local
AP(config-radsrv)# nas 10.91.6.159 key 110337
AP(config-radsrv)# nas 10.91.6.162 key 110337
AP(config-radsrv)# nas 10.91.6.181 key 110337
AP(config-radsrv)# group clerks
AP(config-radsrv-group)# vlan 87
AP(config-radsrv-group)# ssid batman
AP(config-radsrv-group)# ssid robin
AP(config-radsrv-group)# reauthentication time 1800
AP(config-radsrv-group)# block count 2 time 600
AP(config-radsrv-group)# group cashiers
AP(config-radsrv-group)# vlan 97
AP(config-radsrv-group)# ssid deer
AP(config-radsrv-group)# ssid antelope
AP(config-radsrv-group)# ssid elk
AP(config-radsrv-group)# reauthentication time 1800
AP(config-radsrv-group)# block count 2 time 600
AP(config-radsrv-group)# group managers
AP(config-radsrv-group)# vlan 77
AP(config-radsrv-group)# ssid mouse
AP(config-radsrv-group)# ssid chipmunk
AP(config-radsrv-group)# reauthentication time 1800
AP(config-radsrv-group)# block count 2 time 600
AP(config-radsrv-group)# exit
AP(config-radsrv)# user jsmith password twain74 group clerks
AP(config-radsrv)# user stpatrick password snake100 group clerks
AP(config-radsrv)# user nick password uptown group clerks
AP(config-radsrv)# user sam password rover32 group cashiers
AP(config-radsrv)# user patsy password crowder group cashiers
AP(config-radsrv)# user carl password 272165 group managers
AP(config-radsrv)# user vic password lid178 group managers
AP(config-radsrv)# end
```

Setting Up Two Main Servers and a Local Authentication Server: Example

This example shows how to set up two main servers and a local authentication server with a server deadtime of 10 minutes:

```
Router(config)# aaa new-model
Router(config)# aaa group server radius RADIUS_SERVER_GROUP
Router(config-sg-radius)# server 172.20.0.1 auth-port 1000 acct-port 1001
Router(config-sg-radius)# server 172.10.0.1 auth-port 1645 acct-port 1646
Router(config-sg-radius)# server 10.91.6.151 auth-port 1812 acct-port 1813
Router(config)# radius-server host 172.20.0.1 auth-port 1000 acct-port 1001 key 77654
Router(config)# radius-server host 172.10.0.1 auth-port 1645 acct-port 1646 key 77654
Router(config)# radius-server host 10.91.6.151 auth-port 1812 acct-port 1813 key 110337
Router(config)# radius-server deadtime 10
```

In this example, if the WAN link to the main servers fails, the access point or wireless-aware router completes these steps when a LEAP-enabled client device associates:

1. It tries the first server, times out multiple times, and marks the first server as dead.
2. It tries the second server, times out multiple times, and marks the second server as dead.
3. It tries and succeeds using the local authentication server.

If another client device needs to authenticate during the 10-minute deadtime interval, the access point skips the first two servers and tries the local authentication server first. After the deadtime interval, the access point tries to use the main servers for authentication. When setting a deadtime, you must balance the need to skip dead servers with the need to check the WAN link and begin using the main servers again as soon as possible.

Each time an access point or wireless-aware router tries to use the main servers while they are down, the client device that is trying to authenticate might report an authentication timeout. The client device retries and succeeds when the main servers time out and the access point or wireless-aware router tries the local authentication server. You can extend the timeout value on Cisco client devices to accommodate expected server timeouts.

Displaying Local Authentication Server Configuration: Example

The following is sample output for configuration of a local authentication server on the Cisco 2621 router.

```
2621-1# show run
Building configuration...

Current configuration : 2954 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname 2621-1
!
!
aaa new-model
!
!
aaa group server radius RADIUS_LEAP_GROUP
 server 10.0.0.1 auth-port 1812 acct-port 1813
!
aaa authentication login AUTH_LEAP group RADIUS_LEAP_GROUP
aaa session-id common
ip subnet-zero
!
!
ip dhcp pool 2621-dhcp-pool
   network 10.0.0.0 255.0.0.0
!
!
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet1/0
 no ip address
```

```
!
interface FastEthernet1/1
 switchport mode trunk
 no ip address
!
interface FastEthernet1/2
 no ip address
 shutdown
!
interface FastEthernet1/3
 no ip address
 shutdown
!
interface FastEthernet1/4
 no ip address
 shutdown
!
interface FastEthernet1/5
 no ip address
!
!
interface GigabitEthernet1/0
 no ip address
 shutdown
!
interface Vlan1
 ip address 10.0.0.1 255.0.0.0
!
ip classless
!
ip http server
no ip http secure-server
!
!
!
radius-server local
  nas 10.0.0.1 key 0 cisco
  user ap-1 nthash 7 101B2A415547345A5F257908017065100641524253257 20D7D04075D523D4F780A
  user ap-5 nthash 7 144231535C540C7A77096016074B51332753030D0877705A264F450A09720A7307
  user user1 nthash 7 1350344A5B5C227B78057B10107A452232515402097C77002B544B45087D0E7200
!
radius-server host 10.0.0.1 auth-port 1812 acct-port 1813
radius-server key cisco
!
wlccp authentication-server infrastructure AUTH_LEAP
wlccp authentication-server client leap AUTH_LEAP
wlccp wds priority 255 interface Vlan1
!
line con 0
line aux 0
line vty 0 4
!
!
!
end
```

Displaying Local Authentication Server Statistics: Example

The following is sample output for configuration for the **show radius local-server statistics** command:

```
router-2621-1# show radius local-server statistics
Successes              : 11262       Unknown usernames      : 0
Client blocks          : 0           Invalid passwords      : 8
Unknown NAS            : 0           Invalid packet from NAS: 0

NAS : 10.0.0.1
Successes              : 11262       Unknown usernames      : 0
Client blocks          : 0           Invalid passwords      : 8
Corrupted packet       : 0           Unknown RADIUS message : 0
No username attribute  : 0           Missing auth attribute : 0
Shared key mismatch    : 0           Invalid state attribute: 0
Unknown EAP message    : 0           Unknown EAP auth type  : 0

Maximum number of configurable users: 50, current user count: 11
Username                  Successes  Failures  Blocks
vayu-ap-1                      2235         0       0
vayu-ap-2                      2235         0       0
vayu-ap-3                      2246         0       0
vayu-ap-4                      2247         0       0
vayu-ap-5                      2247         0       0
vayu-11                           3         0       0
vayu-12                           5         0       0
vayu-13                           5         0       0
vayu-14                          30         0       0
vayu-15                           3         0       0
scm-test                          1         8       0

router-2621-1#
```

The first section shows cumulative statistics from the local authentication server. The second section shows statistics for each access point (NAS) that is authorized to use the local authentication server. The third section shows statistics for individual users. If a user is blocked and the lockout time is set to infinite, *Blocked* appears at the end of the line of statistics for that user. If the lockout time is not set to infinite, *Unblocked in x seconds* appears at the end of the statistics line for that user.

Additional References

The following sections provide references related to Remote Site IEEE 802.1X Local Authentication Service.

Related Documents

Related Topic	Document Title
Comprehensive set of software configuration commands	*Cisco IOS Software Configuration Guide for Cisco Aironet Access Points*
Configuration commands for wireless roaming	*Configuring Fast Secure Roaming*

MIBs

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

This feature uses no new or modified commands. To see the command pages for the commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

VPN Access Control Using 802.1X Authentication

The home access router provides connectivity to the corporate network via a Virtual Private Network (VPN) tunnel through the Internet. In the home LAN, apart from the employee, other members of the household may also be using the same access router. The VPN Access Control Using 802.1X Authentication feature allows enterprise employees to access their enterprise networks from home while allowing other household members to access only the Internet. The feature uses the Institute of Electrical and Electronics Engineers (IEEE) 802.1X protocol framework to achieve the VPN access control. The authenticated employee has access to the VPN tunnel and others (unauthenticated users on the same LAN) have access only to the Internet.

Feature History for VPN Access Control Using 802.1X Authentication

Release	Modification
12.3(2)XA	This feature was introduced.
12.3(4)T	This feature was integrated into Cisco IOS Release 12.3(4)T, and the following platform support was added: Cisco 1751, Cisco 2610XM – Cisco 2611XM, Cisco 2620XM – Cisco 2621XM, Cisco 2650XM – Cisco 2651XM, Cisco 2691, Cisco 3640, Cisco 3640A, and Cisco 3660.
12.3(11)T	802.1X supplicant support was added.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for VPN Access Control Using 802.1X Authentication, page 1682
- Restrictions for VPN Access Control Using 802.1X Authentication, page 1682
- Information About VPN Access Control Using 802.1X Authentication, page 1682
- How to Configure VPN Access Control Using 802.1X Authentication, page 1685
- Configuration Examples for VPN Access Control Using 802.1X Authentication, page 1703
- Additional References, page 1710
- Additional References, page 1710

- Command Reference, page 1711
- Glossary, page 1713

Prerequisites for VPN Access Control Using 802.1X Authentication

- The PCs connecting behind the router should have 802.1X clients running on them.
- You should know how to configure authentication, authorization, and accounting (AAA) and RADIUS.
- You should be familiar with IP Security (IPSec).
- You should be familiar with Dynamic Host Configuration Protocol (DHCP).
- You should know how to configure user lists on a Cisco access control server (ACS).

Restrictions for VPN Access Control Using 802.1X Authentication

- Easy VPN is not supported.
- VLAN interfaces are currently not supported.
- If there is a switch located between the router and the supplicant (client PC), the Extensible Authentication Protocol over LAN (EAPOL) frames will not reach the router because the switch discards them.

Information About VPN Access Control Using 802.1X Authentication

To configure the VPN Access Control Using 802.1X Authentication feature, you should understand the following concepts:

- How VPN Control Using 802.1X Authentication Works, page 1682
- 802.1X Supplicant Support, page 1684
- Authentication Using Passwords and MD5, page 1684

How VPN Control Using 802.1X Authentication Works

The home access router provides connectivity to the corporate network via a VPN tunnel through the Internet. In the home LAN, both authenticated (employee) and unauthenticated (other household members) users exist, and both have access to the corporate VPN tunnel. Currently there is no existing mechanism to prevent the unauthenticated user from accessing the VPN tunnel.

To distinguish between the users, the VPN Access Control Using 802.1X Authentication feature uses the IEEE 802.1X protocol that allows end hosts to send user credentials on Layer 2 of the network operating system. Unauthenticated traffic users will be allowed to pass through the Internet but will be blocked from accessing the corporate VPN tunnel. The VPN Access Control Using 802.1X feature expands the scope of the 802.1X standard to authenticate devices rather than ports, meaning that multiple devices can be independently authenticated for any given port. This feature separates traffic from authenticated and unauthenticated users so that separate access policies can be applied.

When an 802.1X-capable host starts up, it will initiate the authentication phase by sending the EAPOL-Start 802.1X protocol data unit (PDU) to the reserved IEEE multicast MAC address (01-80-C2-00-00-03) with the Ethernet type or length set to 0x888E.

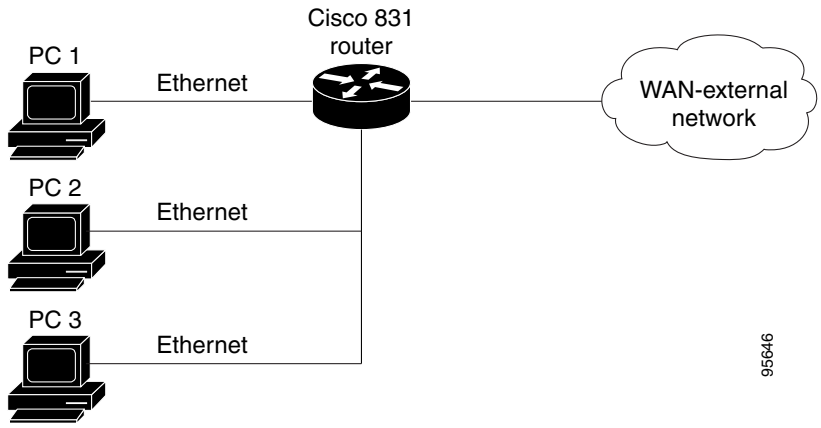
All 802.1X PDUs will be identified as such by the Ethernet driver and will be enqueued to be handled by an 802.1X process. On some platforms, Ethernet drivers have to program the interface address filter so that EAPOL packets can be accepted.

On the router, the receipt of the EAPOL-Start message will result in the source MAC address being "remembered," and an EAPOL-request or identity PDU being sent to the host. The router will send all host-addressed PDUs to the individual MAC address of the host rather than to the multicast address.

802.1X Authentication Sample Topology and Configuration

Figure 112 illustrates a typical scenario in which VPN access control using 802.1X authentication is in place.

Figure 112 Typical 802.1X Authentication Setup



In Figure 112, all the PCs are 802.1X capable hosts, and the Cisco 831 router is an authenticator. All the PCs are connected to the built-in hub or to an external hub. If a PC does not support 802.1X authentication, MAC-based authentication is supported on the Cisco 831 router.

Note

- You can have any kind of connectivity or network beyond the Cisco 831 WAN.
- If there is a switch located between the router and the supplicant (client PC), the EAPOL frames will not reach the router because the switch discards them.
- A supplicant is an entity at one end of a point-to-point LAN segment that is being authenticated by an authenticator that is attached to the other end of that link.

802.1X Supplicant Support

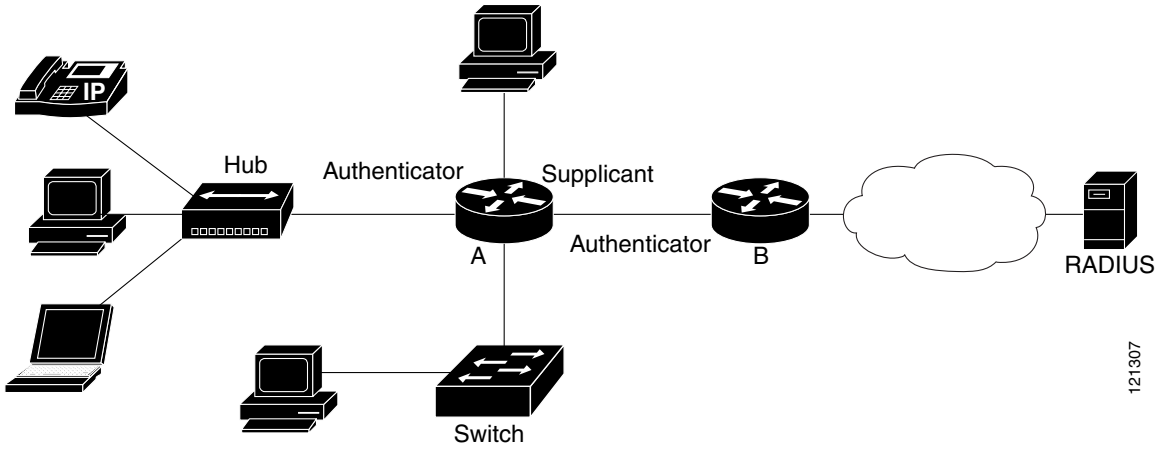
There are deployment scenarios in which a network device (a router acting as an 802.1X authenticator) is placed in an unsecured location and cannot be trusted as an authenticator. This scenario requires that a network device be able to authenticate itself against another network device. The 802.1X supplicant support functionality provides the following solutions for this requirement:

- An Extensible Authentication Protocol (EAP) framework has been included so that the supplicant has the ability to "understand" and "respond" to EAP requests. EAP-Message Digest 5 (EAP-MD5) is currently supported.
- Two network devices that are connected through an Ethernet link can act as a supplicant and as an authenticator simultaneously, thus providing mutual authentication capability.
- A network device that is acting as a supplicant can authenticate itself with more than one authenticator (that is, a single port on a supplicant can be connected to multiple authenticators).

The following illustration is an example of 802.1X supplicant support. The illustration shows that a single supplicant port has been connected to multiple authenticators. Router A is acting as an authenticator to devices that are sitting behind it on the LAN while those devices are acting as supplicants. At the same time, Router B is an authenticator to Router A (which is acting as a supplicant). The RADIUS server is located in the enterprise network.

When Router A tries to authenticate devices on the LAN, it needs to "talk" to the RADIUS server, but before it can allow access to any of the devices that are sitting behind it, it has to prove its identity to Router B. Router B checks the credential of Router A and gives access.

Figure 113 Multiple Instances of Supplicant Support



Authentication Using Passwords and MD5

For information about using passwords and Message Digest 5 (MD5), refer to the following document on Cisco.com:

- *Improving Security on Cisco Routers*

How to Configure VPN Access Control Using 802.1X Authentication

This section includes the following procedures:

- Configuring an AAA RADIUS Server, page 1685
- Configuring a Router, page 1685
- Configuring a PC, page 1699
- Monitoring VPN Access Control Using 802.1X Authentication, page 1701
- Verifying VPN Access Control Using 802.1X Authentication, page 1703

Configuring an AAA RADIUS Server

To configure an AAA RADIUS server, perform the following steps.

Step 1 Configure entries for the network access server and associated shared secrets.

Note The AAA server can be FreeRADIUS or Cisco Secure ACS or any other similar product with 802.1X support.

Step 2 Add the username and configure the password of the user.

Step 3 Configure a global or per-user authentication scheme.

Configuring a Router

This section contains the following procedures:

- Enabling 802.1X Authentication, page 1685 (required)
- Configuring Router and RADIUS Communication, page 1687 (required)
- Configuring 802.1X Parameters (Retransmissions and Timeouts), page 1688 (optional)
- Configuring the Identity Profile, page 1690 (required)
- Configuring the Virtual Template and DHCP, page 1692 (required)
- Configuring the Necessary Access Control Policies, page 1697
- Configuring a Router As a Supplicant, page 1697

Enabling 802.1X Authentication

To enable 802.1X port-based authentication, you should configure the router so that it can communicate with the AAA server, enable 802.1X globally, and enable 802.1X on the interface. To enable 802.1X port-based authentication, perform the following steps.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **aaa new-model**
4. **aaa authentication dot1x default group radius**
5. **dot1x system-auth-control**
6. **identity profile default**
7. **interface** *type slot/port*
8. **dot1x port-control auto**

DETAILED STEPS

	Command	Description
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router (config)# aaa new-model`	Enables AAA.
Step 4	**aaa authentication dot1x default group radius** **Example:** `Router (config)# aaa authentication dot1x default group radius`	Creates an 802.1X port-based authentication method list.
Step 5	**dot1x system-auth-control** **Example:** `Router (config)# dot1x system-auth-control`	Globally enables 802.1X port-based authentication.
Step 6	**identity profile default** **Example:** `Router (config)# identity profile default`	Creates an identity profile and enters dot1x profile configuration mode.
Step 7	**interface** *type slot/port* **Example:** `Router (config)# interface fastethernet 5/1`	Enters interface configuration mode and specifies the interface to be enabled for 802.1X port-based authentication.
Step 8	**dot1x port-control auto** **Example:** `Router (config-if)# dot1x port-control auto`	Enables 802.1X port-based authentication on the interface.

Example

This section provides the following examples:

- 802.1X Configuration
- Verifying 802.1X Authentication

802.1X Configuration

The following example shows that 802.1X authentication has been configured on a router:

```
Router# configure terminal
Router(config)# aaa new-model
Router(config)# aaa authentication dot1x default group radius
Router(config)# dot1x system-auth-control
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x port-control auto
```

Verifying 802.1X Authentication

The following **show dot1x** command sample output shows that 802.1X authentication has been configured on a router:

```
Router# show dot1x all

PortControl       = AUTO
ReAuthentication  = Disabled
ReAuthPeriod      = 3600 Seconds
ServerTimeout     = 30 Seconds
SuppTimeout       = 30 Seconds
QuietWhile        = 120 Seconds
MaxReq            = 2
```

Configuring Router and RADIUS Communication

To configure RADIUS server parameters, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip radius source-interface** *interface-name*
4. **radius-server host** {*hostname* | *ip-address*}
5. **radius-server key** *string*

DETAILED STEPS

	Command	Description
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip radius source-interface** *interface-name* **Example:** `Router (config)# ip radius source-interface ethernet1`	Forces RADIUS to use the IP address of a specified interface for all outgoing RADIUS packets.
Step 4	**radius-server host** {*hostname* | *ip-address*} **Example:** `Router (config)# radius-server host 172.16.39.46`	Configures the RADIUS server host name or IP address of the router. • To use multiple RADIUS servers, reenter this command for each server.
Step 5	**radius-server key** *string* **Example:** `Router (config)# radius-server key radiuskey`	Configures the authorization and encryption key used between the router and the RADIUS daemon running on the RADIUS server. • The key is a text string that must match the encryption key used on the RADIUS server.

Example

The following example shows that RADIUS server parameters have been configured on the router:

```
Router# configure terminal
Router(config)# ip radius source-interface ethernet1
Router(config)# radius-server host 172.16.39.46
Router(config)# radius-server key radiuskey
```

Configuring 802.1X Parameters (Retransmissions and Timeouts)

Various 802.1X retransmission and timeout parameters can be configured. Because all of these parameters have default values, configuring them is optional. To configuring the retransmission and timeout parameters, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/port*
4. **dot1x max-req** *number-of-retries*

5. **dot1x port-control** [**auto** | **force-authorized** | **force-unauthorized**]
6. **dot1x reauthentication**
7. **dot1x timeout tx-period** *seconds*
8. **dot1x timeout server-timeout** *seconds*
9. **dot1x timeout reauth-period** *seconds*
10. **dot1x timeout quiet-period** *seconds*
11. **dot1x timeout ratelimit-period** *seconds*

DETAILED STEPS

	Command	Description
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot/port* **Example:** `Router (config)# interface ethernet 0/1`	Enters interface configuration mode and specifies the interface to be enabled for 802.1X port-based authentication.
Step 4	**dot1x max-req** *number-of-retries* **Example:** `Router (config-if)# dot1x max-req 3`	Sets the maximum number of times that the router sends an EAP request/identity frame (assuming that no response is received) to the supplicant before concluding that the supplicant does not support 802.1X.
Step 5	**dot1x port-control** [**auto** | **force-authorized** | **force-unauthorized**] **Example:** `Router (config-if)# dot1x port-control auto`	Sets the port control value. • **auto** (optional)—Authentication status of the supplicant will be determined by the authentication process. • **force-authorized** (optional)**—**All the supplicants on the interface will be authorized. The **force-authorized** keyword is the default. • **force-unauthorized** (optional)—All the supplicants on the interface will be unauthorized.
Step 6	**dot1x reauthentication** **Example:** `Router (config-if)# dot1x reauthentication`	Enables periodic reauthentication of the supplicants on the interface. • The reauthentication period can be set using the **dot1x timeout** command.

	Command	Description
Step 7	**dot1x timeout tx-period** *seconds* **Example:** `Router (config-if)# dot1x timeout tx-period 60`	Sets the timeout for supplicant retries. • If an 802.1X packet is sent to the supplicant and the supplicant does not send a response, the packet will be sent again after the time that was set using the *seconds* argument. • The value is 1 through 65535 seconds. The default is 30 seconds.
Step 8	**dot1x timeout server-timeout** *seconds* **Example:** `Router (config-if)# dot1x timeout server-timeout 60`	Sets the timeout for RADIUS retries. • If an 802.1X packet is sent to the server, and the server does not send a response, the packet will be sent again after the time that was set using the *seconds* argument. • The value is from 1 to 65535 seconds. The default is 30 seconds.
Step 9	**dot1x timeout reauth-period** *seconds* **Example:** `Router (config-if)# dot1x timeout reauth-period 1800`	Sets the time after which an automatic reauthentication should be initiated. • The value is from 1 to 65535 seconds. The default is 3600 seconds.
Step 10	**dot1x timeout quiet-period** *seconds* **Example:** `Router (config-if)# dot1x timeout quiet-period 600`	The time after which authentication is restarted after the authentication has failed. • The value is from 1 to 65535 seconds. The default is 120 seconds.
Step 11	**dot1x timeout ratelimit-period** *seconds* **Example:** `Router (config-if)# dot1x timeout ratelimit-period 60`	The rate limit period throttles the EAP-START packets from misbehaving supplicants. • The value is from 1 to 65535 seconds.

Example

The following configuration example shows that various retransmission and timeout parameters have been configured:

```
Router# configure terminal
Router(config)# interface ethernet 0
Router(config-if)# dot1x port-control auto
Router(config-if)# dot1x reauthentication
Router(config-if)# dot1x timeout reauth-period 1800
Router(config-if)# dot1x timeout quiet-period 600
Router(config-if)# dot1x timeout supp-timeout 60
Router(config-if)# dot1x timeout server-timeout 60
```

Configuring the Identity Profile

The **identity profile default** command allows you to configure the static MAC addresses of the client that do not support 802.1X and to authorize or unauthorize them statically. The VPN Access Control Using 802.1X Authentication feature allows authenticated and unauthenticated users to be mapped to different interfaces. Under the **dot1x profile** configuration mode, you can specify the virtual template

interface that should be used to create the virtual-access interface to which unauthenticated supplicants will be mapped. To specify which virtual template interface should be used to create the virtual access interface, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **identity profile default**
4. **description text** *line-of-description*
5. **template** *virtual-template*
6. **device** [**authorize** | **not-authorize**] **mac-address** *mac-address*
7. **device authorize type** *device-type*

DETAILED STEPS

	Command	Description
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**identity profile default** **Example:** `Router (config)# identity profile default`	Creates an identity profile and enters identity profile configuration mode.
Step 4	**description** *line-of-description* **Example:** `Router (config-identity-prof)# description description 1`	Associates descriptive text with the profile.
Step 5	**template** *virtual-template* **Example:** `Router (config-identity-prof)# template virtual-template 1`	Specifies the virtual template interface that will serve as the configuration clone source for the virtual interface that is dynamically created for authenticated users.

	Command	Description
Step 6	**device** [**authorize** | **not-authorize**] **mac-address** *mac-address* **Example:** `Router (config-identity-prof)# device authorize mac-address mac-address H.H.H`	Statically authorizes or unauthorizes a supplicant (by giving its MAC address) if the supplicant does not "understand" 802.1X.
Step 7	**device authorize type** *device-type* **Example:** `Router (config-identity-prof)# device authorize type cisco ip phone`	Statically authorizes or unauthorizes a device type.

Example

The following example shows that Cisco IP phones and a specific MAC address have been statically authorized:

```
Router# configure terminal
Router (config)# identity profile default
Router(config-1x-prof)# description put the description here
Router(config-1x-prof)# template virtual-template1
Router(config-1x-prof)# device authorize type cisco ip phone
Router(config-1x-prof)# device authorize mac-address 0001.024B.B4E7
```

Configuring the Virtual Template and DHCP

The VPN Access Control Using 802.1X Authentication feature can be configured with one DHCP pool or two. If there are two pools, the unauthenticated and authenticated devices will get their addresses from separate DHCP pools. For example, the public pool can have an address block that has only local significance, and the private pool can have an address that is routable over the VPN tunnel. To configure your router for a private pool and for a public pool, perform the following steps.

SUMMARY STEPS

Configuring the Identity Profile

1. **enable**
2. **configure terminal**
3. **identity profile default**
4. **description** *description-string*
5. **template** *virtual-template*
6. **exit**

Configuring the DHCP Private Pool

1. **ip dhcp pool** *name*
2. **network** *network-number* [*mask*]
3. **default-router** *address*

Configuring the DHCP Public Pool

1. **ip dhcp pool** *name*
2. **network** *network-number* [*mask*]
3. **default-router** *address*
4. **exit**

Configuring the Interface

1. **configure terminal**
2. **interface** *type slot/port*
3. **ip address** *ip-address mask* [**secondary**]
4. **interface virtual-template** *number*
5. **ip address** *ip-address mask* [**secondary**]
6. **exit**

Configuring an Interface Without Assigning an Explicit IP Address to the Interface

1. **enable**
2. **configure terminal**
3. **interface** *type slot/port*
4. **ip unnumbered** *type number*

DETAILED STEPS

Configuring the Identity Profile

	Command	Description
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**identity profile default** **Example:** `Router (config)# identity profile default`	Creates an identity profile and enters identity profile configuration mode.
Step 4	**description** *description-string* **Example:** `Router (config-identity-prof)# description description_string_goes_here`	Associates descriptive text with the identity profile.

	Command	Description
Step 5	**template** *virtual-template* **Example:** `Router (config-identity-prof)# template virtualtemplate1`	Specifies the virtual template interface that will serve as the configuration clone source for the virtual interface that is dynamically created for authenticated users.
Step 6	**exit** **Example:** `Router (config-identity-prof)# exit`	Exits identity profile configuration mode.

Configuring the DHCP Private Pool

	Command	Description
Step 1	**ip dhcp pool** *name* **Example:** `Router (config)# ip dhcp pool private`	Configures a DHCP private address pool on a Cisco IOS DHCP server and enters DHCP pool configuration mode.
Step 2	**network** *network-number* [*mask*] **Example:** `Router (config-dhcp)# network 10.0.0.1 255.0.0.0`	Configures the subnet number and mask for a DHCP private address pool on a Cisco IOS DHCP server.
Step 3	**default-router** *address* **Example:** `Router (config-dhcp)# default-router 10.2.2.2`	Specifies the default router list for a DHCP client.

Configuring the DHCP Public Pool

	Command	Description
Step 1	**ip dhcp pool** *name* **Example:** `Router (config-dhcp)# ip dhcp pool public`	Configures the DHCP public address pool on a Cisco IOS DHCP server.
Step 2	**network** *network-number* [*mask*] **Example:** `Router (config-dhcp)# network 10.4.4.4.255.0.0.0`	Configures the subnet number and mask for a DHCP public address pool on a Cisco IOS DHCP server.

	Command	Description
Step 3	**default-router** *address* **Example:** `Router (config-dhcp)# default-router 10.12.12.12`	Specifies the default router list for a DHCP client.
Step 4	**exit** **Example:** `Router (config-dhcp)# exit`	Exits DHCP pool configuration mode.

Configuring the Interface

	Command	Description
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**interface** *type slot*/*port* **Example:** `Router (config)# interface loopback 0/1`	Enters interface configuration mode and specifies the interface to be enabled.
Step 3	**ip address** *ip-address mask* [**secondary**] **Example:** `Router (config-if)# ip address 10.5.5.5 255.255.255.0`	Sets the private IP address for the interface.
Step 4	**interface virtual-template** *number* `Router (config-if)# interface virtual-template 1`	Creates a virtual template interface that can be configured and applied dynamically in creating virtual access interfaces.
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Router (config-if)# ip address 10.6.6.6 255.255.255.0`	Sets the public IP address for the interface.
Step 6	**exit** **Example:** `Router (config-if)# exit`	Exits interface configuration mode.

Configuring an Interface Without Assigning an Explicit IP Address to the Interface

	Command	Description
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot/port* **Example:** `Router (config)# interface virtual-template 1/2`	Enters interface configuration mode and specifies the interface to be enabled.
Step 4	**ip unnumbered** *type number* **Example:** `Router (config-if)# ip unnumbered loopback 0`	Enables IP processing on an interface without assigning an explicit IP address to the interface.

Example

The following example shows that the identity profile associates virtual-template1 with unauthenticated supplicants. Virtual-template1 gets its IP address from interface loopback 0, and unauthenticated supplicants are associated with a public pool. Authenticated users are associated with a private pool.

```
Router(config)# identity profile default
Router(config-1x-prof)# description put the description here
Router(config-1x-prof)# template virtual-template1
Router(config-1x-prof)# exit

Router(config)# ip dhcp pool private
Router(config-dhcp)# network 10.0.0.1 255.0.0.0
Router(config-dhcp)# default-router 10.2.2.2
Router(config-dhcp)# exit

Router(config)#ip dhcp pool public
Router(config-dhcp)# network 10.4.4.4 255.0.0.0
Router(config-dhcp)# default-router 10.12.12.12
Router(config-dhcp)# exit

Router(config)# interface loopback0
Router(config-if)# ip address 10.5.5.5 255.255.255.0
Router(config-if)# interface ethernet0
Router(config-if)# ip address 10.6.6.6 255.255.255.0
Router(config-if)# exit

Router(config)# interface virtual-template1
Router(config-if)# ip unnumbered loopback 0
```

Configuring the Necessary Access Control Policies

802.1X authentication separates traffic from authenticated and unauthenticated devices. Traffic from authenticated devices transit via the physical interface, and unauthenticated traffic transits via the Virtual-Template1. Therefore, different policies can be applied on each interface. The configuration will also depend on whether two DHCP pools or a single DHCP pool is being used. If a single DHCP pool is being used, access control can be configured on Virtual-Template1, which will block any traffic from going to the networks to which unauthenticated devices should not have access. These networks (to which unauthenticated devices should not have access) could be the corporate subnetworks protected by the VPN or encapsulated by generic routing encapsulation (GRE). There can also be access control that restricts the access between authenticated and unauthenticated devices.

If two pools are configured, the traffic from a non-trusted pool is routed to the Internet using Network Address Translation (NAT), whereas trusted pool traffic is forwarded via a VPN tunnel. The routing can be achieved by configuring ACLs used by NAT and VPN accordingly.

For an example of an access control policy configuration, see the "Access Control Policies: Example" section.

Configuring a Router As a Supplicant

To configure a router to act as a supplicant, you have to first configure the identity profile that the supplicant will use to obtain its EAP credentials. Then you have to configure the interface as a supplicant Port Access Entity (PAE) type. To configure a router as a supplicant, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **identity profile dot1x**
4. **eap username** *name*
5. **eap password** *password*
6. **exit**
7. **interface** *type number*
8. **dot1x pae supplicant**

DETAILED STEPS

	Command	Description
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command	Description
Step 3	**identity profile dot1x** **Example:** `Router (config)# identity profile dot1x`	Creates an identity profile and enters identity profile configuration mode.
Step 4	**eap username** *name* **Example:** `Router (config-identity-prof)# eap username user1`	Creates an identity profile and username that will be sent to Request-Id packets.
Step 5	**eap username** *password* **Example:** `Router (config-identity-prof)# eap username password`	Password that should be used when replying to a MD5 challenge.
Step 6	**exit** **Example:** `Router (config-identity-prof)# exit`	Exits identity profile configuration mode.
Step 7	**interface** *type number* **Example:** `Router# interface Ethernet1`	Configures an interface type and enters interface configuration mode.
Step 8	**dot1x pae supplicant** **Example:** `Router (config-if)# dot1x pae supplicant`	Sets the PAE type. • The **supplicant** keyword specifies that the interface will be acting only as a supplicant and will not respond to messages that are meant for an authenticator.

Configuring a PC

This section includes the following procedures.

- Configuring a PC for VPN Access Control Using 802.1X Authentication, page 1699
- Enabling 802.1X Authentication on a Windows 2000/XP PC, page 1699
- Enabling 802.1X Authentication on a Windows 2000 PC, page 1699
- Enabling 802.1X Authentication on a Windows XP PC, page 1700
- Enabling 802.1X Authentication on Windows 2000 and Windows XP PCs, page 1700

Configuring a PC for VPN Access Control Using 802.1X Authentication

To configure your PC for VPN Access Control Using 802.1X Authentication, perform the following steps.

Step 1 Enable 802.1X for MD5.

Step 2 Enable DHCP.

Enabling 802.1X Authentication on a Windows 2000/XP PC

802.1X implementation on a Windows 2000/XP PC is unstable. A more stable 802.1X client, AEGIS (beta) for Microsoft Windows, is available at the Meetinghouse Data Communications website at www.mtghouse.com.

Enabling 802.1X Authentication on a Windows 2000 PC

To enable 802.1X authentication on your Windows 2000 PC, perform the following steps.

Step 1 Make sure that the PC has at least Service Pack 3.

Go to the page “Microsoft 802.1x Authentication Client” on the Microsoft Windows 2000 website at the following URL:

http://www.microsoft.com/windows2000/server/evaluation/news/bulletins/8021xclient.asp.

At the above site, download and install 802.1X client for Windows 2000.

If the above site is unavailable, search for the “Q313664: Recommended Update” page on the Microsoft Windows 2000 website at the following URL:
http://www.microsoft.com/windows2000/downloads/recommended/q313664/default.asp

Step 2 Reboot your PC after installing the client.

Step 3 Go to the Microsoft Windows registry and add or install the following entry:

“HKLM\Software\Microsoft\EAPOL\Parameters\General\Global\SupplicantMode REG_DWORD 3”

("SupplicantMode" key entry is not there by default under Global option in the registry. So add a new entry named "SupplicantMode" as REG_DOWORD and then set its value to 3.)

Step 4 Reboot your PC.

Enabling 802.1X Authentication on a Windows XP PC

To enable 802.1X authentication on a Windows XP PC, perform the following steps.

Step 1 Go to the Microsoft Windows registry and install the following entry there:

"HKLM\Software\Microsoft\EAPOL\Parameters\General\Global\SupplicantMode REG_DWORD 3"

Step 2 Reboot your PC.

Enabling 802.1X Authentication on Windows 2000 and Windows XP PCs

To enable 802.1X authentication on Windows 2000 and Windows XP PCs, that is, if you are operating both at the same time, perform the following steps.

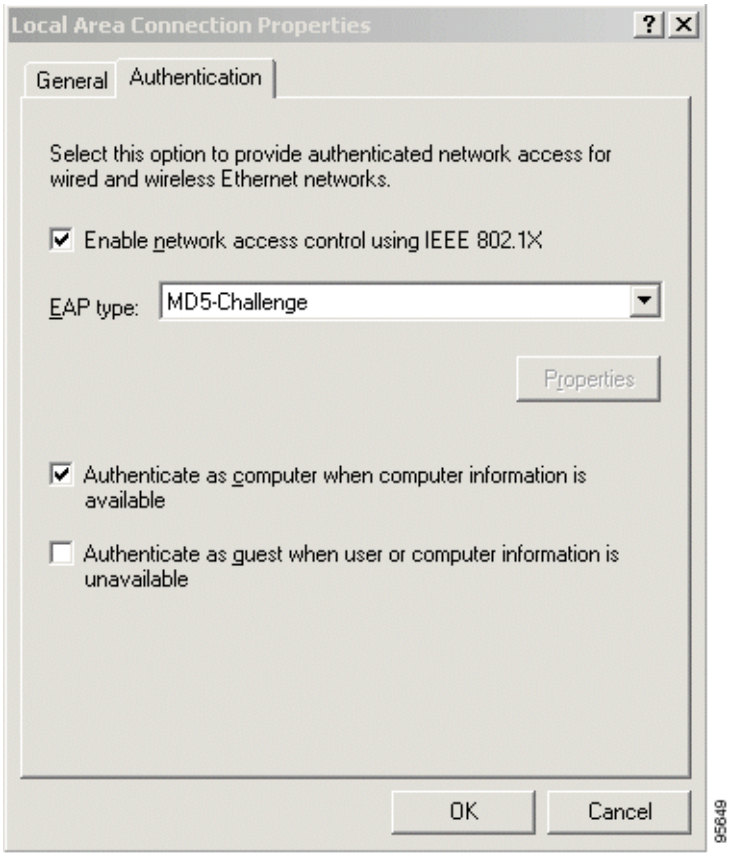
Step 1 Open the Network and Dial-up Connections window on your computer.

Step 2 Right-click the Ethernet interface (Local Area Connection) to open the properties window. It should have a tab called "Authentication."

Click the Authentication tab. Select the check box titled "Enable network access control using IEEE 802.1X."

In a short period of time you should see a dialog box (for Windows 2000) or a floating window asking you to select it. Select it, and when the next window appears, enter the username and password in this dialog box. See Figure 114.

Figure 114 Local Area Connection Properties Window



Monitoring VPN Access Control Using 802.1X Authentication

To monitor VPN Access Control Using 802.1X Authentication, perform the following steps. The commands shown in the steps may be used one at a time and in no particular order.

SUMMARY STEPS

1. **enable**
2. **debug dot1x** [**aaa** | **all** | **process** | **rxdata** | **state-machine** | **txdata** | **vlan**]
3. **clear dot1x**
4. **dot1x initialize** [**interface** *interface-name*]
5. **dot1x re-authenticate** *interface-type interface-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug dot1x** [**aaa** | **all** | **process**| **rxdata** | **state-machine** | **txdata** |**vlan**] **Example:** `Router# debug dot1x all`	Displays 802.1X debugging information. • **aaa**—Information is provided for AAA communications. • **all**—All 802.1X debugging messages are turned on. • **process**—Information is provided regarding the 802.1X process. • **rxdata**—Information is provided for packets that have been received from clients. • **state-machine**—Information is provided regarding the 802.1X state-machine. • **txdata**—Information is provided regarding packets that have been transmitted to clients. • **vlan**—Information is provided regarding the MAC address-based VLAN operation. **Note** VLAN interfaces are currently not supported.
Step 3	**clear dot1x** **Example:** `Router# clear dot1x`	Clears 802.1X interface information.
Step 4	**dot1x initialize** [**interface** *interface-name*] `Router# dot1x initialize interface ethernet 0`	Initializes an interface.
Step 5	**dot1x re-authenticate** *interface-type interface-number* **Example:** `Router# dot1x re-authenticate ethernet 0`	Reauthenticates all the authenticated devices that are attached to the specified interface.

Verifying VPN Access Control Using 802.1X Authentication

To verify VPN Access Control Using 802.1X Authentication, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show dot1x** [**interface** *interface-name* [**details**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show dot1x** [**interface** *interface-name* [**details**]] **Example:** `Router# show dot1x interface ethernet details`	Shows details for an identity profile.

Configuration Examples for VPN Access Control Using 802.1X Authentication

This section includes the following example:

- Typical VPN Access Control Using 802.1X Configuration: Example, page 1703
- Access Control Policies: Example, page 1707
- Router As an Authenticator and Supplicant: Example

Typical VPN Access Control Using 802.1X Configuration: Example

The following sample output shows that VPN access control using 802.1X authentication has been configured. Output is shown for the router and for the gateway.

Router

```
Router# show running-config

Building configuration...

Current configuration: 2100 bytes
!
version 12.3
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
```

```
!
hostname c831-tb
!
memory-size iomem 15
!
aaa new-model
!
!
aaa authentication dot1x default group radius
aaa session-id common
ip subnet-zero
!
ip dhcp pool private
   network 10.0.0.0 255.255.255.0
   default-router 10.0.0.1
   lease 0 0 2
!
ip dhcp pool public
   network 10.3.0.0 255.255.255.0
   default-router 10.3.0.1
!
!
ip audit notify log
ip audit po max-events 100
no ftp-server write-enable
!
!
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key 0 test address 150.0.0.2
!
!
crypto ipsec transform-set t1 ah-md5-hmac esp-des
crypto mib ipsec flowmib history tunnel size 2
crypto mib ipsec flowmib history failure size 2
!
crypto map test 1 ipsec-isakmp
 set peer 150.0.0.2
 set transform-set t1
 match address 101
!
dot1x system-auth-control
identity profile default
 template Virtual-Template1
!
!
interface Loopback0
 ip address 10.3.0.1 255.255.255.0
!
interface Ethernet0
 ip address 10.0.0.1 255.255.255.0
 dot1x port-control auto
 dot1x reauthentication
 dot1x timeout reauth-period 36000
!
interface Ethernet1
 no ip address
 duplex auto
 pppoe enable
 pppoe-client dial-pool-number 1
!
interface Virtual-Template1
 ip unnumbered Loopback0
```

```
 ip access-group 102 in
 ip access-group 102 out
!
interface Dialer0
 ip address 172.0.0.1 255.255.255.0
 ip mtu 1492
 encapsulation ppp
 dialer pool 1
 crypto map test
!
interface Dialer1
 no ip address
!
router rip
 network 10.0.0.0
 network 10.3.0.0
 network 172.0.0.0
!
ip classless
ip http server
no ip http secure-server
!
!
ip access-list extended list1
access-list 101 permit ip 10.0.0.0 0.0.0.255 10.5.0.0 0.0.0.255
access-list 102 deny   ip 10.3.0.0 0.0.0.255 10.5.0.0 0.0.0.255
access-list 102 deny   ip 10.2.0.0 0.0.0.255 10.5.0.0 0.0.0.255
access-list 102 permit ip any any
radius-server host 192.168.140.50 auth-port 1812 acct-port 1646 key radiuskey
!
line con 0
 exec-timeout 0 0
 no modem enable
line aux 0
line vty 0 4
!
scheduler max-task-time 5000
!
end
```

Peer Router As Gateway

```
Router# show running-config

Building configuration...
Current configuration: 1828 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname c3725
!
!
no aaa new-model
ip subnet-zero
!
vpdn enable
!
vpdn-group 1
 accept-dialin
  protocol pppoe
```

```
  virtual-template 1
!
mpls ldp logging neighbor-changes
!
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key 0 test address 172.0.0.1
!
!
crypto ipsec transform-set t1 ah-md5-hmac esp-des
crypto mib ipsec flowmib history tunnel size 2
crypto mib ipsec flowmib history failure size 2
!
crypto map test 1 ipsec-isakmp
 set peer 172.0.0.1
 set transform-set t1
 match address 101
!
no voice hpi capture buffer
no voice hpi capture destination
!
interface Loopback0
 description corporate
 ip address 10.5.5.5 255.255.255.0
!
interface Loopback1
 description internet
 ip address 10.6.6.6 255.255.255.0
!
interface FastEthernet0/0
 ip address 192.168.140.100 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 speed auto
 half-duplex
 pppoe enable
!
interface ATM1/0
 ip address 10.0.0.10 255.255.255.0
 no atm ilmi-keepalive
 pvc 1/43
  protocol ip 10.75.0.4 broadcast
  encapsulation aal5snap
!
!
interface FastEthernet2/0
 no ip address
 speed auto
 full-duplex
!
interface FastEthernet2/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Virtual-Template1
 ip address 10.150.0.2 255.255.255.0
 ip mtu 1492
 crypto map test
!
```

```
!
router rip
 network 10.5.0.0
 network 10.6.0.0
 network 10.75.0.0
 network 172.0.0.0
 network 192.168.140.0
!
ip http server
no ip http secure-server
ip classless
!
access-list 101 permit ip 10.5.0.0 0.0.0.255 10.0.0.1 0.0.0.255
no cdp log mismatch duplex
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 login
!
!
end
```

Access Control Policies: Example

The following output example shows that access control policies have been configured.

Single DHCP pool

```
ip dhcp pool1 private
 network 10.0.0.0 255.255.255.0
 default-router 20.0.0.1
 exit
crypto isakmp policy 1
 authentication pre-share
!
crypto isakmp key test address address
 crypto ipsec transform-set t1 esp-3des esp-sha-hmac
 mode tunnel
 crypto map test 1 ipsec-isakmp
 set peer address
 set transform-set t1
 match address 101
 access-list 101 permit ip 10.0.0.0 0.0.0.255 50.0.0.0 0.0.0.255
 access-list 102 deny ip 10.0.0.0 0.0.0.255 50.0.0.0 0.0.0.255
 access-list 102 permit  ip any any
!
interface Ethernet0
! inside interface
! dot1x configs
!
interface Virtual-Template1
! Deny traffic from going to VPN
 ip access-group 102 in
!
Interface Ethernet1
! outside interface
 crypto map test
```

Two DHCP Pools

```
ip dhcp pool private
 network 10.0.0.0 255.255.255.0
 default-router 20.0.0.1
 exit
!
ip dhcp pool public
 network 10.0.0.1 255.255.255.0
 default-router 10.0.0.2
 exit
!
crypto isakmp policy 1
 authentication pre-share
!
crypto isakmp key test address address
 crypto ipsec transform-set t1 esp-3des esp-sha-hmac
 mode tunnel
 crypto map test 1 ipsec-isakmp
 set peer address
 set transform-set t1
 match address 101
 access-list 101 permit ip 10.0.0.0 0.0.0.255 50.0.0.0 0.0.0.255
 access-list 102 permit ip 10.0.0.1 0.0.0.255 any
!
interface Ethernet0
!inside interface
! dot1x configs
!
interface Loopback0
 ip address 10.0.0.3 255.255.255.0
!
interface Virtual-Template1
 ip unnumbered Loopback0
 ip nat inside
!
Interface Ethernet1
! outside interface
 crypto map test
 ip nat outside
!
ip nat inside source list 102 interface Ethernet1 overload
```

Router Acting As a Supplicant: Example

The following example shows that dot1x module debugging has been turned on. The **show debugging** command output shows that 802.1X interface information has been cleared for all interfaces.

```
Router# debug dot1x supplicant

dot1x supplicant module debugging is on

Router# show debugging

dot1x:
  dot1x supplicant module debugging is on

3w6d: supp_pae_state_transition: Supplicant State Transition: DISCONNECTED -> CONNECTING
3w6d: supp_pae_txStart: << txStart >>: EAPOL-Start to Authenticator
3w6d: dot1x_pakio_send_pak: Zero destination address, sending to multicast
3w6d: dot1x_pakio_send_pak: Sending packet to group PAE address 0180.c200.0003
```

```
3w6d: dot1x_pakio_send_pak: Sending packet to address 0180.c200.0003
3w6d: dot1x_start_supp_timer: Started the Timer for client 0000.0000.0000, 30 seconds
3w6d: dot1x_reset_client: sm->state == CONNECTING
3w6d: clear_dot1x_client_supp_table: Clearing all dot1x supplicant instances
3w6d: supp_pae_state_transition: Supplicant State Transition: AUTHENTICATED -> LOGOFF
3w6d: supp_pae_txLogoff: << Router#txLogoff >>: EAPOL-Logoff to Authenticator
3w6d: dot1x_pakio_send_pak: Sending packet to address 000f.23c4.a401
3w6d: dot1x_client_logoff: sm->state == LOGOFF
3w6d: clear_dot1x_client_supp_bucket: Logoff Sent !!
3w6d: dot1x_reset_client: Stopping timers before re-intialization
3w6d: dot1x_reset_client: Re-initializing the default supplicant
3w6d: supp_pae_state_transition: Supplicant State Transition: CONNECTING -> DISCONNECTED
3w6d: supp_pae_state_transition: Supplicant State Transition: DISCONNECTED -> CONNECTING
3w6d: supp_pae_txStart: << txStart >>: EAPOL-Start to Authenticator
3w6d: dot1x_pakio_send_pak: Zero destination address, sending to multicast
3w6d: dot1x_pakio_send_pak: Sending packet to group PAE address 0180.c200.0003
3w6d: dot1x_pakio_send_pak: Sending packet to address 0180.c200.0003
3w6d: dot1x_start_supp_timer: Started the Timer for client 0000.0000.0000, 30 seconds
3w6d: dot1x_reset_client: sm->state == CONNECTING
3w6d: dot1x_get_def_client_supp_entry: Using the default supplicant instance
3w6d: dot1x_input: Stopping timers on the default supp instance
3w6d: dot1x_stop_supp_timer: Stopped the Timer for client 0000.0000.0000
3w6d: dot1x_get_client_supp_entry: Supplicant 000f.23c4.a401 not found in the supplicant
list
3w6d: dot1x_input: Creating a new supplicant entry
3w6d: dot1x_get_supp_config: Using the default EAP method
3w6d: dot1x_pakio_uplink_addr_set: Uplink address set to 00:0F:23:C4:A4:01
3w6d: dot1x_pakio_init_ios: Initialising common IOS structures for dot1x
3w6d: dot1x_pakio_init_ios: Done.
3w6d: dot1x_eap_init: Initialising EAP method 4
3w6d: dot1x_eap_init: Username:user, password:cisco
3w6d: dot1x_eap_init: sm->state == DISCONNECTED
3w6d: supp_pae_state_transition: Supplicant State Transition: INVALID -> DISCONNECTED
3w6d: supp_pae_state_transition: Supplicant State Transition: DISCONNECTED -> CONNECTING
3w6d: supp_pae_txStart: << txStart >>: EAPOL-Start to Authenticator
3w6d: dot1x_pakio_send_pak: Sending packet to address 000f.23c4.a401
3w6d: dot1x_start_supp_timer: Started the Timer for client 000f.23c4.a401, 30 seconds
3w6d: dot1x_eap_init: sm->state == CONNECTING
3w6d: add_dot1x_client_supp_to_table:
3w6d: dot1x_get_def_client_supp_entry: Using the default supplicant instance000f.23c4.a401
is added to the supplicant list
3w6d: dot1x_input: QUEUE_EVENT (active pkt count tx 0, rx 1, total tx 3, total rx 10)
3w6d: dot1x_input: Stopping timers on the default supp instance
3w6d: dot1x_get_client_supp_entry: Supplicant instance found
3w6d: dot1x_input: Found an existing supplicant entry
3w6d: supp_pae_event_handler: Received packet:
3w6d: supp_pae_state_transition: Supplicant State Transition: CONNECTING -> AQUIRED
3w6d: dot1x_stop_supp_timer: Stopped the Timer for client 000f.23c4.a401
3w6d: supp_pae_txRspId: << txRspId >>: EAPOL-EAP-Response-Id to Authenticator
3w6d: supp_pae_txRspId: ReceivedId is 0x1  and currentId is 0x100
3w6d: dot1x_pakio_send_pak: Sending packet to address 000f.23c4.a401
3w6d: dot1x_start_supp_timer: Started the Timer for client 000f.23c4.a401, 30 seconds
3w6d: dot1x_input: QUEUE_EVENT (active pkt count tx 0, rx 1, total tx 2, total rx 2)
3w6d: supp_pae_event_handler: Received packet:
3w6d: supp_pae_state_transition: Supplicant State Transition: AQUIRED -> AQUIRED
3w6d: dot1x_stop_supp_timer: Stopped the Timer for client 000f.23c4.a401
3w6d: supp_pae_txRspId: << txRspId >>: EAPOL-EAP-Response-Id to Authenticator
3w6d: supp_pae_txRspId: ReceivedId is 0x1  and currentId is 0x1
3w6d: dot1x_pakio_send_pak: Sending packet to address 000f.23c4.a401
3w6d: dot1x_start_supp_timer: Started the Timer for client 000f.23c4.a401, 30 seconds
3w6d: dot1x_input: QUEUE_EVENT (active pkt count tx 0, rx 0, total tx 3, total rx 2)
3w6d: dot1x_get_def_client_supp_entry: Using the default supplicant instance
3w6d: dot1x_input: Stopping timers on the default supp instance
3w6d: dot1x_get_client_supp_entry: Supplicant instance found
```

```
3w6d: dot1x_input: Found an existing supplicant entry
3w6d: supp_pae_event_handler: Received packet:
3w6d: supp_pae_state_transition: Supplicant State Transition: AQUIRED -> AUTHENTICATING
3w6d: dot1x_stop_supp_timer: Stopped the Timer for client 000f.23c4.a401
3w6d: supp_pae_txRspAuth: << txRspAuth >>: EAPOL-EAP-Response to Authenticator
3w6d: dot1x_pakio_send_pak: Sending packet to address 000f.23c4.a401
3w6d: dot1x_start_supp_timer: Started the Timer for client 000f.23c4.a401, 30 seconds
3w6d: dot1x_input: QUEUE_EVENT (active pkt count tx 0, rx 0, total tx 4, total rx 3)
3w6d: dot1x_get_def_client_supp_entry: Using the default supplicant instance
3w6d: dot1x_input: Stopping timers on the default supp instance
3w6d: dot1x_get_client_supp_entry: Supplicant instance found
3w6d: dot1x_input: Found an existing supplicant entry
3w6d: supp_pae_event_handler: Received packet:
3w6d: supp_pae_state_transition: Supplicant State Transition: AUTHENTICATING ->
AUTHENTICATED
3w6d: supp_pae_state_transition: Changing IP addr in AUTHENTICATED state
3w6d: supp_pae_state_transition: Stopped client timers
3w6d: dot1x_stop_supp_timer: Stopped the Timer for client 000f.23c4.a401
3w6d: dot1x_input: QUEUE_EVENT (active pkt count tx 0, rx 0, total tx 4, total rx 4)
```

Additional References

The following sections provide references related to VPN Access Control Using 802.1X Authentication.

Related Documents

Related Topic	Document Title
Passwords and MD5	*Improving Security on Cisco Routers*
AAA	"*Authentication, Authorization, and Acccounting (AAA)*" section of the *Cisco IOS Security Configuration Guide*, Release 12.3
RADIUS	"*Security Server Protocols*" section of the *Cisco IOS Security Configuration Guide*, Release 12.3
IPSec	"*Security and Encryption*" section of the *Cisco IOS Security Configuration Guide*, Release 12.3
DHCP	"Configuring DHCP" chapter of the *Cisco IOS IP Configuration Guide*, Release 12.3
User lists on a Cisco ACS	*User Guide for Cisco Secure ACS for Windows Server Version 3.2.*
Security commands	*Cisco IOS Security Command Reference*

Standards

Standards	Title
IEEE 802.1X protocol	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC-2284	"RFC 2284 (PPP Extensible Authentication Protocol [EAP])" document from *The Internet Requests for Comments (RFC)* document series

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

New Commands

- **aaa authentication dot1x**
- **clear dot1x**
- **debug dot1x**
- **description (identity profile)**
- **device (identity profile)**
- **dot1x initialize**
- **dot1x max-req**
- **dot1x max-start**
- **dot1x pae**

- **dot1x port-control**
- **dot1x re-authenticate (privileged EXEC)**
- **dot1x reauthentication**
- **dot1x system-auth-control**
- **dot1x timeout**
- **eap**
- **show dot1x**
- **template (identity profile)**

Glossary

authenticator—Entity at one end of a point-to-point LAN segment that enforces host authentication. The authenticator is independent of the actual authentication method and functions only as a pass-through for the authentication exchange. It communicates with the host, submits the information from the host to the authentication server, and authorizes the host when instructed to do so by the authentication server.

supplicant—Entity at one end of point-to-point LAN segment that is being authenticated by an authenticator that is attached to the other end of that link.

Note Refer to *Internetworking Terms and Acronyms* for terms not included in this glossary.

WebVPN

The Cisco WebVPN feature provides remote access to enterprise sites by users from anywhere on the Internet. The Secure Socket Layer (SSL) Virtual Private Network (VPN) provides users with secure access to specific enterprise applications, such as e-mail and web browsing, without requiring them to have VPN client software installed on their end-user devices.

Feature History for WebVPN

Release	Modification
12.3(14)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for WebVPN, page 1715
- Restrictions for WebVPN, page 1716
- Information About WebVPN, page 1716
- How to Configure WebVPN, page 1724
- Configuration Examples for WebVPN, page 1737
- Additional References, page 1738
- Command Reference, page 1739

Prerequisites for WebVPN

- To securely access resources on a private network behind a WebVPN gateway, the user of a WebVPN service must have the following:
  - An account (login name and password)
  - An SSL-enabled browser (for example, Internet Explorer, Netscape, Mozilla, or FireFox)

  - Operating system, such as Windows 2000 or Windows XP with Sun MicroSystems Java Runtime.
  - E-mail client, such as Eudora, Microsoft Outlook, or Netscape Mail.
- Before a user can access resources on a private network behind a web VPN, the administrator of a web VPN service has to configure basic WebVPN functionality on a router as shown in the section "Configuring WebVPN."

Restrictions for WebVPN

- If WebVPN has to be enabled on a router that is running HTTP Secure Server, the administrator must configure an IP address for WebVPN using the **gateway-addr** keyword option of the **webvpn enable** command.
- The browsing of URLs that are referred by Macromedia Flash are not modified for the secure retrieval by the WebVPN gateway.
- In Cisco IOS Release 12.3(14)T, this feature supports SSL Version 3. Transport Layer Security (TLS) is not supported.
- "Thin Client" used for TCP port-forwarding applications requires administrative privileges on the computer of the end user.

Information About WebVPN

To configure the WebVPN feature, you should understand the following concept:

- WebVPN, page 1716

WebVPN

The WebVPN feature provides end users with unrestricted, secure remote access to enterprise sites without having VPN installed on their end devices. Users can access the enterprise sites from anywhere on the Internet and can access enterprise applications such as e-mail and web browsing.

This feature provides for an administrator interface and an end-user interface.

Administrator Interface

Enterprise administrators can enable WebVPN functionality for their end users through the command-line-interface (CLI) on their Cisco IOS routers. See the section "Configuring WebVPN."

End User Interface

A user whose enterprise has configured WebVPN can access the enterprise network by launching a browser and connecting to the WebVPN gateway that is hosted by the enterprise network. The user will present his or her credentials, be authenticated, and see the portal page (home page) of the enterprise site. The portal page will display those functionalities (for example, e-mail and Web browsing) to which the user has access on the basis of his or her credentials. If the user has access to all functionalities of the WebVPN gateway, the home page will provide links to all those functionalities.

Note The user interface is primarily an HTML interface.

The following sections explain the user interface in more detail:

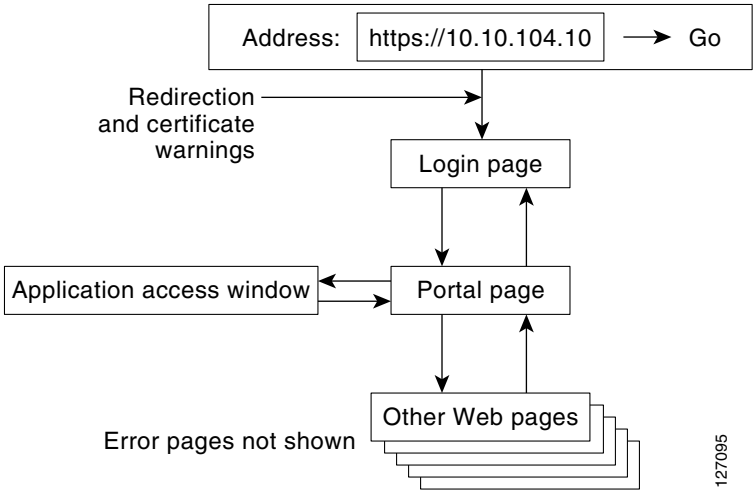
- Page Flow, page 1717
- Initial Connection, page 1717
- Login Page, page 1718
- Certificate Authentication, page 1718
- Logout Page, page 1719
- Portal Page, page 1719
- Remote Servers, page 1720
- DNS and Connection Errors, page 1722
- Session Timeout, page 1723
- TCP Port Forwarding and Application Access, page 1723

Page Flow

This section describes the page flow that an end user will see as he or she uses a WebVPN session. The user first enters the Hypertext Transfer Protocol Secure (HTTPS) URL (https://address) into his or her browser. The user is redirected to https://address/index.html, where the login page is located.

Figure 115 illustrates the flow of pages that the user may expect to see.

Figure 115 Page Flow



Initial Connection

If the user enters the HTTP URL, the browser will be redirected to the equivalent HTTPS URL. Depending on the configuration of the browser, this redirection may cause a warning in the browser of the user indicating that he or she is being redirected to a secure connection.

On establishment of the HTTPS connection, the user may receive a warning about the SSL/TLS certificate. The user should install this certificate. The user does not receive a warning if the administrator has installed a certificate that the browser of the user trusts or if the user had previously installed the certificate.

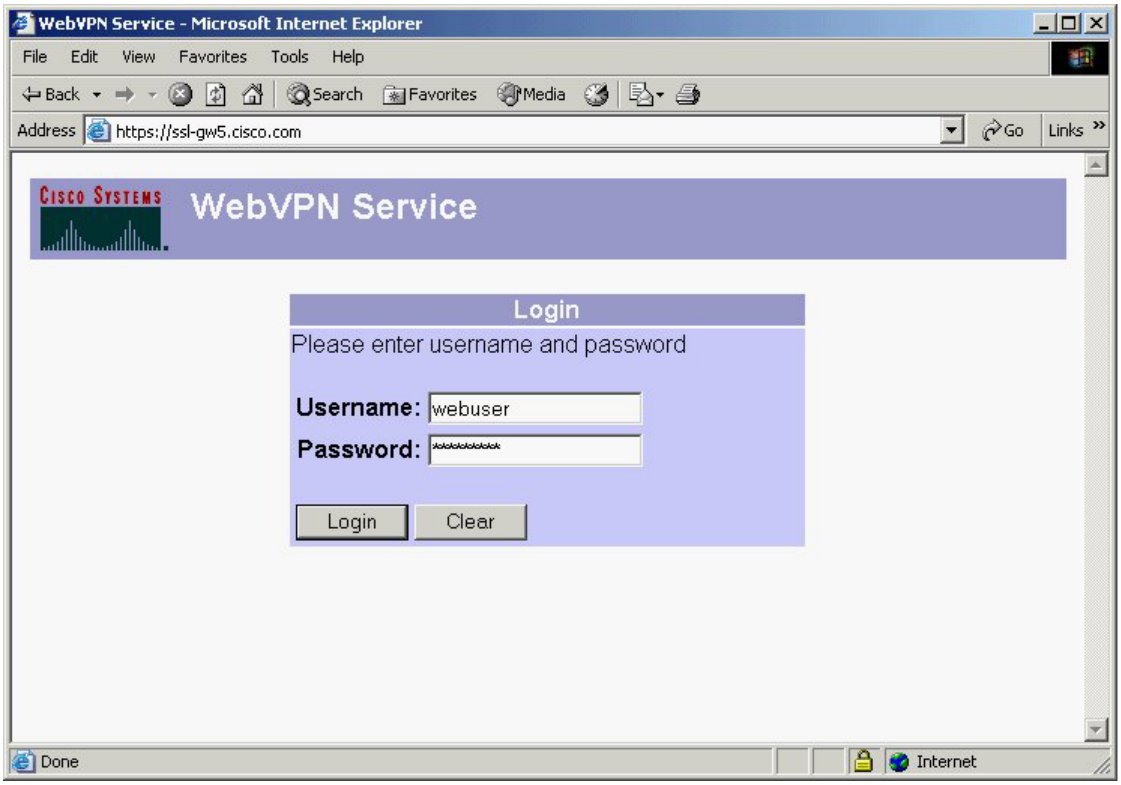
The user will then be connected to the login page.

Login Page

The login page is where the user will be prompted to enter his or her credentials. The credentials consist of a username and password, which are entered into an HTML form. If an authentication failure occurs, the user will be presented with the login page again but with an error message.

Figure 116 illustrates a default login page.

Figure 116 Default Login Page



Note Only the fields that are necessary for the challenge are presented on the login page.

The login page has logos, titles, messages, and colors that may be customized by administrators.

Certificate Authentication

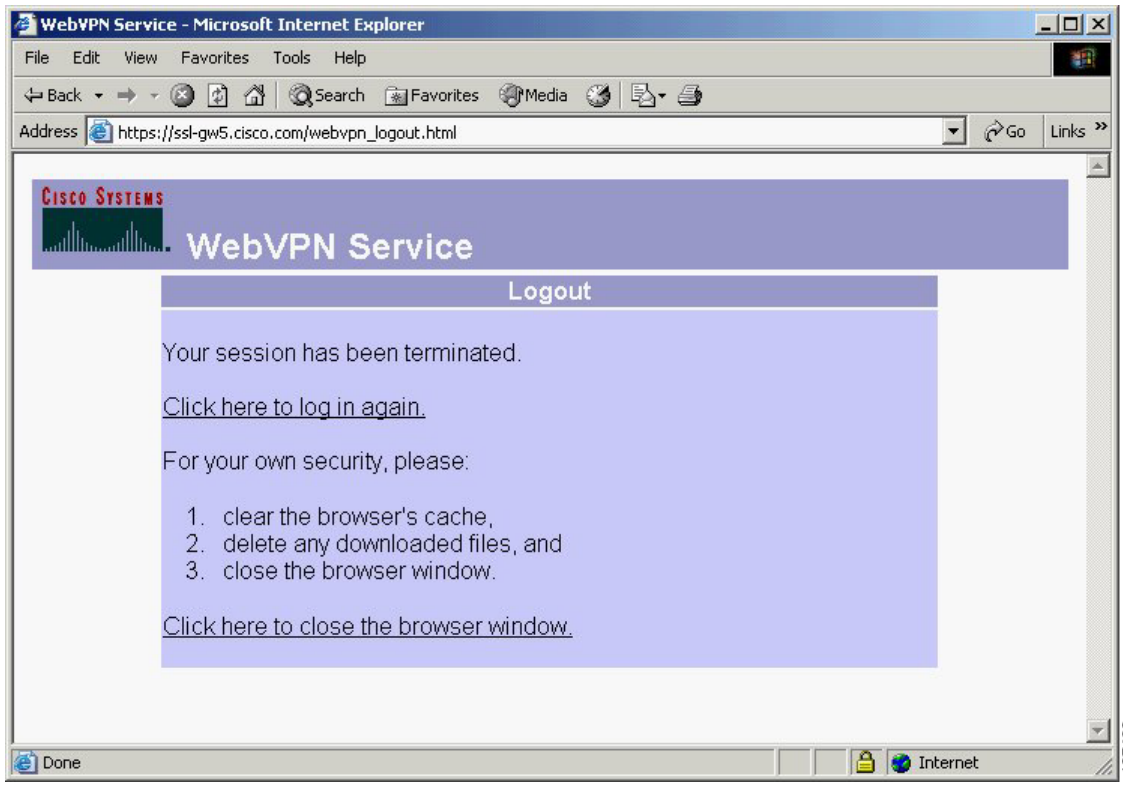
Client certificate authentication is not supported. Only username and password authentication is needed.

Logout Page

If the user clicks the logout link, or if his or her session terminates because of an idle timeout or maximum connection time, the user is presented with the logout page.

Figure 117 illustrates a logout page.

Figure 117 Logout Page



Portal Page

The portal page is the main page for the WebVPN functionality. This page is a customizable page that contains the following:

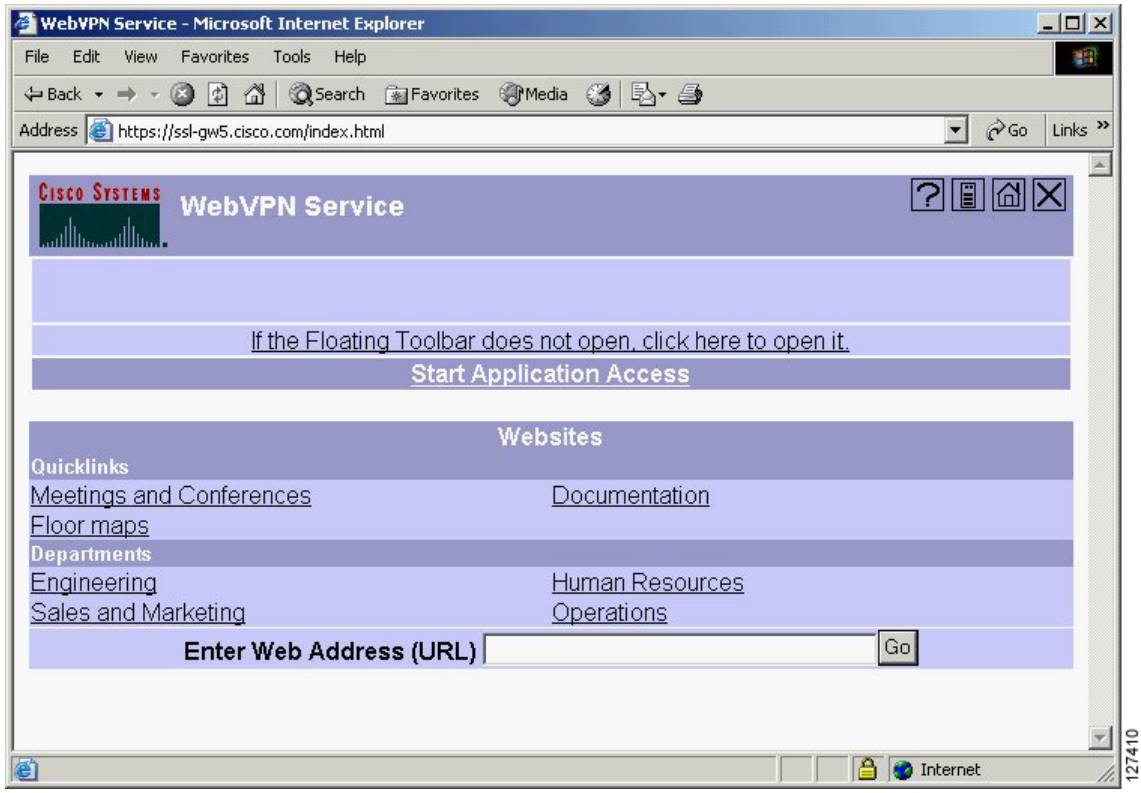
- Custom logo (the default is the Cisco bridge logo
- Custom title (the default is “WebVPN Services”)
- Custom banner (the default is an empty string)
- Custom colors (the default is a combination of white and purples)
- List of web server links (customizable)
- URL entry box (always present)
- Application access link (always present)
- Icon links for Help, Home (that is, the portal page), and Logout
- Link to popup, floating toolbar

Items that have not been configured will not be displayed on the portal page.

Note E-mail access is supported by "thin client," which is downloaded using the application access link.

Figure 118 illustrates a portal page.

Figure 118 Portal Page



Remote Servers

An end user may enter an address or URL path of a website to which he or she wants to visit either in the text box on the portal page or in the text box on the floating toolbar. Pages from the remote server will be displayed in the browser window. The user can then browse to other links on the page normally.

Figure 119 illustrates the portal page of a typical website. By clicking the home icon button on the floating toolbar (see Figure 120), the user can go back to the portal page.

Figure 119 Website with a Toolbar

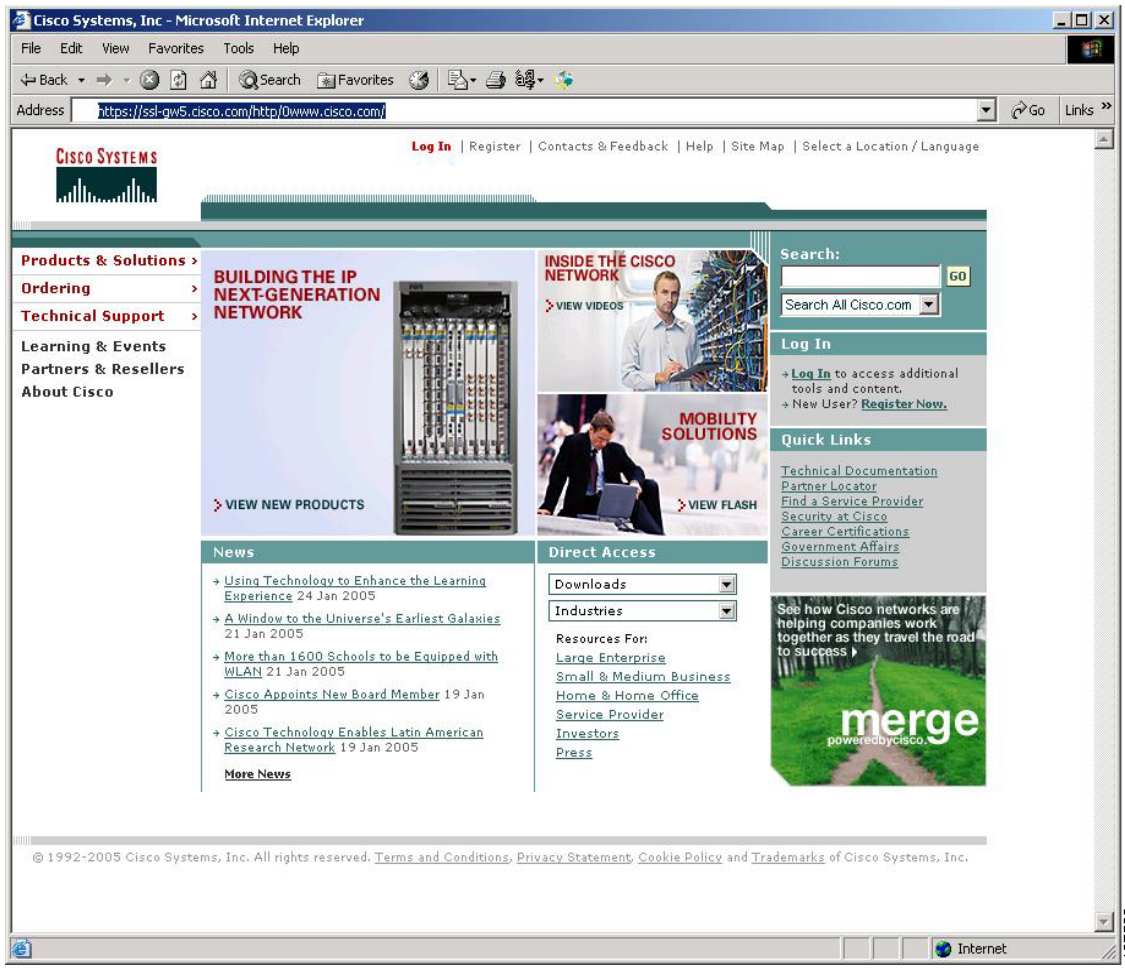
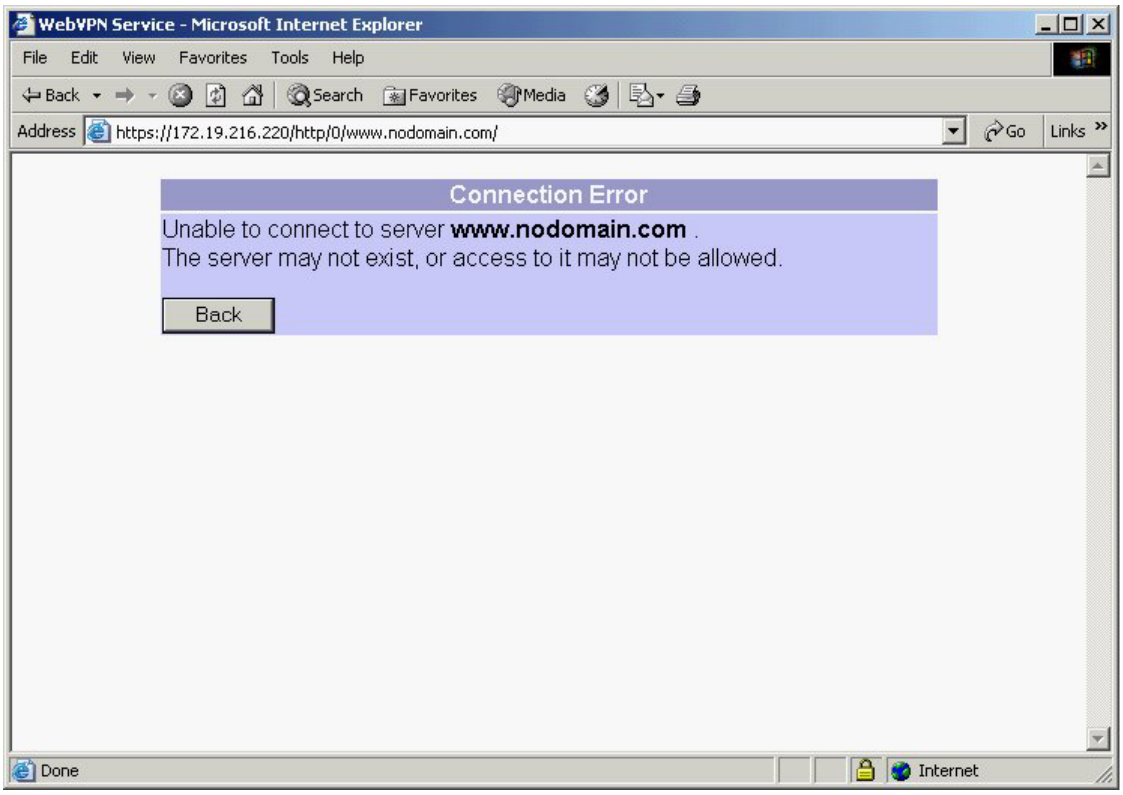


Figure 120 Floating Toolbar

DNS and Connection Errors

If a user specifies a remote server to which he or she cannot connect because of domain naming system (DNS) or other connection errors, the user is presented with a friendly error, as shown in Figure 121. Because of TCP timeouts, it may take a while for connection errors to be returned to the user.

Figure 121 DNS Errors



Session Timeout

Users will be warned when their sessions are about to expire because of inactivity. The user will be presented with a small, centered window as shown in Figure 122. The user will receive a warning approximately 1 minute before the session expires and another warning when the session expires. The time of the workstation will be displayed to indicate when the message was displayed.

The first message will be one of the following:

- "Your session will expire in *x* seconds due to inactivity. Click [Close] to reset the inactivity timer. (browser time and date)"

Clicking the [Close] button on the idle warning message will reset the inactivity timer.

The last message, as shown below, will be displayed when time runs out (depending on whether the reason of the session termination is known):

- "Your session has expired due to inactivity."

Figure 122 Session Inactivity or Timeout Window

TCP Port Forwarding and Application Access

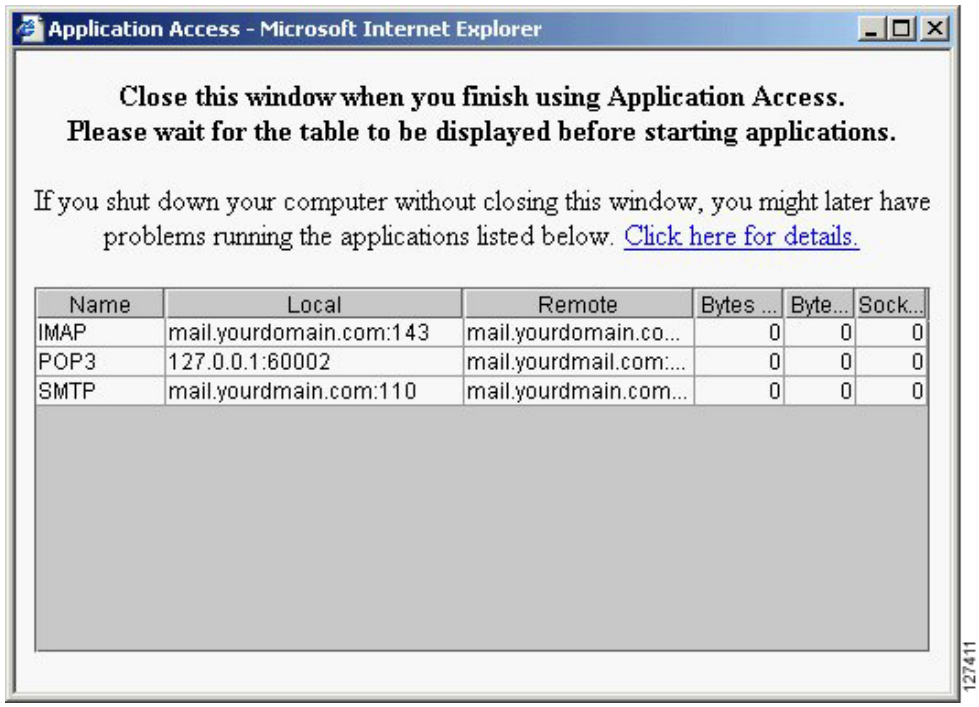
When the Application Access link is clicked, a new window is displayed. This window initiates the downloading of a port-forwarding applet. Another window is then displayed. This window asks the user to verify the certificate with which this applet is signed. When the user accepts the certificate, the applet starts running, and port-forwarding entries are displayed. The administrator should have configured IP addresses, DNS names, and port numbers for the e-mail servers. The user can then launch Email Client, which is configured to contact the above e-mail servers and have them send and receive e-mails. Point of Presence3 (POP3), Internet Message Access Protocol (IMAP), and Simple Mail Transfer Protocol (SMTP) protocols are supported.

This feature will require the Java 1.4 Java Virtual Machine (JVM) to properly support SSL connections. The number of active connections and bytes that are sent and received is also listed on this window. The user may then open the client programs and connect to the local port. This window should not be subject to advertisement or popup blockers.

An attempt will be made to have this window close automatically if the user is logged out using JavaScript. If the session of the user is terminated and a new port forwarding connection is established, the applet will indicate the error.

Figure 123 illustrates a typical port forwarding page.

Figure 123 TCP Port Forwarding Page



How to Configure WebVPN

This section contains the following procedures:

- Configuring WebVPN: Prerequisites, page 1724 (required)
- Configuring WebVPN, page 1728 (required)
- Defining Encryption Algorithms for the SSL Protocol, page 1730 (optional)
- Displaying URL Entries on the Portal Page, page 1731 (optional)
- Maintaining and Monitoring Your WebVPN Functionality, page 1732 (optional)
- Troubleshooting WebVPN, page 1736 (optional)

Configuring WebVPN: Prerequisites

Before configuring WebVPN, an administrator must configure and install the following:

- AAA-Related Configuration, page 1725
- DNS-Related Configuration, page 1726
- Certificates and Trustpoints, page 1726

AAA-Related Configuration

Before configuring WebVPN for a AAA-related configuration, an administrator must create user accounts using either local authentication or authentication via AAA (RADIUS and TACACS+ servers) and configure AAA-related commands.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa group server radius** *group-name*
4. **server** *ip-address* [**auth-port** *port-number*] [**acct-port** *port-number*]
5. **exit**
6. **aaa authentication login** {**default** | **list-name**} *method*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa group server radius** *group-name* **Example:** `Router# aaa group server radius EMAIL-AUTH`	Groups different RADIUS server hosts into distinct lists and distinct methods and enters server-group configuration mode.
Step 4	**server** *ip-address* [**auth-port** *port-number*] [**acct-port** *port-number*] **Example:** `Router (config-server-group)# server 10.1.1.1 auth-port 2 acct-port 3`	Configures the IP address of the RADIUS server for the group server.
Step 5	**exit** **Example:** `Router (config-server-group)# exit`	Exits server-group configuration mode.
Step 6	**aaa authentication login** {**default** | **list-name**} method **Example:** `Router (config)# aaa authentication login default EMAIL-AUTH`	Sets AAA authentication at login.

DNS-Related Configuration

Before configuring WebVPN, an administrator must configure DNS-related commands, such as the following.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip domain name** *name*
4. **ip name server** *server-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip domain name** *name* **Example:** `Router (config)# ip domain name cisco.com`	Defines a default domain name that the Cisco IOS software uses to complete unqualified hostnames (names without a dotted-decimal domain name).
Step 4	**ip name server** *server-address* **Example:** `Router (config)# ip name server 172.16.1.111`	Specifies the address of one or more name servers to use for name and address resolution.

Certificates and Trustpoints

Before configuring WebVPN, an administrator must install certificates and configure trustpoints. To load the certificate, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki import** *trustpointname* **pkcs12** *source url passphrase*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki import** *trustpointname* **pkcs12** *source url passphrase* **Example:** `Router# crypto pki import SSLVPN pkcs12 tftp:cisco`	Imports Rivest, Shamir, and Adelman (RSA) keys.

Examples

After configuring the **crypto pki import pkcs** command, an administrator will see something like the following on his or her console:

```
Router(config)# crypto pki import SSLVPN pkcs12 tftp: cisco
        Address or name of remote host []? 10.1.1.1
        Source filename [SSLVPN]? cert/SSLVPN.cert
        Loading cert/SSLVPN.cert from 10.1.1.1 (via Ethernet1/0):
       !
        Router(config)#
```

The above would generate the crypto-specific commands, as shown in the following sample **show running-config** command output:

```
crypto pki trustpoint SSLVPN
     revocation-check crl
     rsakeypair SSLVPN

crypto pki certificate chain SSLVPN
     certificate 77220E6A00000000130E
.
.
.
```

Configuring WebVPN

To configure WebVPN functionality on your router, perform the following steps. All steps except Step 1 and Step 2 are optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn enable** [**gateway-addr** *ip-address*]
4. **webvpn**
5. **title** *title-string*
6. **login-message** *message-string*
7. **title-color** *color*
8. **secondary-color** *color*
9. **text-color** [**black** | **white**]
10. **secondary-text-color** [**black** | **white**]
11. **idle-timeout** *seconds*
12. **ssl encryption** [**3des-sha1**] [**des-sha-1**] [**rc4-md5**]
13. **ssl trustpoint** *trustpoint-name*
14. **port-forward** {**list** *list-name*} {**local-port** *port-number*} {**remote-server** *server-name-or-ip-address*} {**remote-port** *port-number*}
15. **url-list** *list-name*
16. **logo** [**file** *filename* | **none**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn enable** [**gateway-addr** *ip-address*] **Example:** `Router (config)# webvpn enable`	Enables WebVPN in the system. • The **gateway-addr** keyword and *ip-address* argument enable WebVPN on only the IP address that is specified.
Step 4	**webvpn** **Example:** `Router (config)# webvpn`	Enters the WebVPN configuration mode.
Step 5	**title** *title-string* **Example:** `Router (config-webvpn)# title "Secure Corporate Access: Unauthorized users prohibited"`	(Optional) Enters the HTML title string that is shown in the browser title and on the title bar (also known as the banner).
Step 6	**login-message** *message-string* **Example:** `Router (config-webvpn)# login-message "Please enter your username and password"`	(Optional) Configures the HTML that prompts the user to log in to a web VPN.
Step 7	**title-color** *color* **Example:** `Router (config-webvpn)# title-color green`	(Optional) Specifies the color of the title bars on the login, home, and file access pages.
Step 8	**secondary-color** *color* **Example:** `Router (config-webvpn)# secondary-color yellow`	(Optional) Specifies the color of the secondary title bars on the login, home, and file access pages.
Step 9	**text-color** [**black** | **white**] **Example:** `Router (config-webvpn)# text-color black`	(Optional) Sets the color of the text on the title bars.
Step 10	**secondary-text-color** [**black** | **white**] **Example:** `Router (config-webvpn)# secondary-text-color black`	(Optional) Specifies the color of the text on the secondary bars.

	Command or Action	Purpose
Step 11	**idle-timeout** *seconds* **Example:** `Router (config-webvpn)# idle-timeout 60`	(Optional) Sets the default idle timeout.
Step 12	**ssl encryption** [**3des-sha1**] [**des-sha-1**] [**rc4-md5**] **Example:** `Router (config-webvpn)# ssl encryption 3des-sha1`	Specifies the encryption algorithms that the SSL protocol will use for a SSL Virtual Private Network (SSLVPN).
Step 13	**ssl trustpoint** *trustpoint-name* **Example:** `Router (config-webvpn)# ssl trustpoint Trustpoint1`	Specifies the certificate trustpoint.
Step 14	**port-forward** {**list** *list-name*} {**local-port** *port-number*} {**remote-server** *server-name-or-ip-address*} {**remote-port** *port-number*} **Example:** `Router (config-webvpn)# port-forward list POP3 local-port 60002 remote-server mail.youremail.com remote-port 25`	Lists the set of forwarded ports to which a user has access.
Step 15	**url-list** *list-name* **Example:** `Router (config-webvpn)# url-list My List`	Configures the list of URLs to which a user has access on the portal page of a SSL VPN and enters URL configuration mode.
Step 16	**logo** [**file** *filename* | **none**] **Example:** `Router (config-webvpn-url)# logo file flash://webvpn/company-logo.gif.`	(Optional) Specifies the custom logo image that is displayed on the login and portal pages.

Defining Encryption Algorithms for the SSL Protocol

To define the encryption algorithms that the SSL protocol will use, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn enable** [**gateway-addr** *ip-address*]
4. **webvpn**
5. **ssl encryption** [**3des-sha1**] [**des-sha1**] [**rc4-md5**]
6. **ssl trustpoint** *trustpoint-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn enable** [**gateway-addr** *ip-address*] **Example:** `Router (config)# webvpn enable`	Enables WebVPN in the system.
Step 4	**webvpn** **Example:** `Router (config)# webvpn`	Enters WebVPN configuration mode.
Step 5	**ssl encryption** [**3des-sha1**] [**des-sha1**] [**rc4-md5**] **Example:** `Router (config-webvpn)# ssl encryption 3des-sha1`	Specifies the encryption algorithms.
Step 6	**ssl trustpoint** *trustpoint-name* **Example:** `Router (config-webvpn)# ssl trustpoint Trustpoint1`	Specifies the certificate trustpoint.

Displaying URL Entries on the Portal Page

To display a list of URLs on the portal page from which users may access common resources, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **webvpn enable** [**gateway-addr** *ip-address*]
4. **webvpn**
5. **url-list** *list-name*
6. **heading** *heading-name*
7. **url-text** *text* **url-value** *URL*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**webvpn enable** [**gateway-addr** *ip-address*] **Example:** `Router (config)# webvpn enable`	Enables WebVPN in the system.
Step 4	**webvpn** **Example:** `Router (config)# webvpn`	Enters WebVPN configuration mode.
Step 5	**url-list** *list-name* **Example:** `Router (config-webvpn)# url-list englist`	Configures the list of URLs to which a user has access on the portal page.
Step 6	**heading** *heading-name* **Example:** `Router (config-webvpn-url)# heading Engineering`	Sets the heading that is displayed above all URLs on the portal page.
Step 7	**url-text** *text* **url-value** *URL* **Example:** `Router (config-webvpn-url)# url-text ENG url-value http://www.eng.mycompany.com`	Sets the text of the link to be displayed on the home page and the URL that is under the link.

Maintaining and Monitoring Your WebVPN Functionality

To maintain and monitor your WebVPN functionality, you may use the following **debug** and **show** commands. The **enable** command is required for each **debug** and **show** command.

SUMMARY STEPS

1. **enable**
2. **debug webvpn** [**aaa** | **cookie** | **dns** | **http** | **port-forward** | **webservice**]
3. **show webvpn sessions**
4. **show webvpn statistics**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug webvpn** [**aaa** | **cookie** | **dns** | **http** | **port-forward** | **webservice**] **Example:** `Router# debug webvpn port-forward`	Enables web VPN session monitoring.
Step 3	**show webvpn sessions** **Example:** `Router# show webvpn sessions`	Displays information about WebVPN sessions.
Step 4	**show webvpn statistics** **Example:** `Router# show webvpn statistics`	Displays WebVPN statistics.

Examples

The following examples show **debug webvpn** output for various WebVPN sessions:

```
Router# debug webvpn

*Jan 19 03:05:22.796: SSLVPN: Entering APPL with Context: 0x0D2B1EB0,
          Data buffer(buffer: 0x0D2EF888, data: 0x1A7E756C, len: 335, offset: 0, domain:
0)
*Jan 19 03:05:22.796: SSLVPN: http request: / with domain cookie
*Jan 19 03:05:22.796: SSLVPN: [Q]Client side Chunk data written..
 buffer=0x0D2EF748 total_len=1600 bytes=1600 tcb=0x0C5920C8
*Jan 19 03:05:22.796: SSLVPN: Client side Chunk data written..
 buffer=0x0D2EF8A8 total_len=1167 bytes=1167 tcb=0x0C5920C8
*Jan 19 03:05:22.836: SSLVPN: Entering APPL with Context: 0x0D2B1EB0,
          Data buffer(buffer: 0x0D2EF888, data: 0x1A7E836C, len: 383, offset: 0, domain:
0)
*Jan 19 03:05:22.836: SSLVPN: http request: /paramdef.js with domain cookie
*Jan 19 03:05:22.836: SSLVPN: Created 323 byte content data to send to external client
*Jan 19 03:05:22.836: SSLVPN: Client side Chunk data written..
 buffer=0x0D2EF8A8 total_len=440 bytes=440 tcb=0x0C5920C8
*Jan 19 03:05:22.860: SSLVPN: Entering APPL with Context: 0x0D2B1EB0,
          Data buffer(buffer: 0x0D2EF888, data: 0x1A7E916C, len: 381, offset: 0, domain:
0)
*Jan 19 03:05:22.860: SSLVPN: http request: /shared.js with domain cookie
*Jan 19 03:05:22.860: SSLVPN: [Q]Client side Chunk data written..
 buffer=0x0D2EF8A8 total_len=1600 bytes=1600 tcb=0x0C5920C8
*Jan 19 03:05:22.860: SSLVPN: Client side Chunk data written..
 buffer=0x0D2EF748 total_len=986 bytes=986 tcb=0x0C5920C8
*Jan 19 03:05:22.896: SSLVPN: Entering APPL with Context: 0x0D2B1EB0,
          Data buffer(buffer: 0x0D2EF888, data: 0x1A7E9F6C, len: 384, offset: 0, domain:
0)
*Jan 19 03:05:22.896: SSLVPN: http request: /img/logo.gif with domain cookie
*Jan 19 03:05:22.896: SSLVPN: Created 552 byte content data to send to external client
```

```
*Jan 19 03:05:22.896: SSLVPN: Client side Chunk data written..
 buffer=0x0D2EF748 total_len=669 bytes=669 tcb=0x0C5920C8
```

The following is sample output when authentication has failed and when authentication has passed:

```
Router# debug webvpn

*Jan 19 03:08:28.428: SSLVPN: AAA authentication request sent for user: "cisco"
*Jan 19 03:08:28.428: SSLVPN: AAA Authentication Failed !

*Jan 19 03:08:42.148: SSLVPN: AAA authentication request sent for user: "cisco"
*Jan 19 03:08:42.148: SSLVPN: AAA Authentication Passed !
```

The following sample output displays WebVPN cookie output during login:

```
Router# debug webvpn cookie

*Jan 19 03:10:38.880: SSLVPN: ipaddr: 172.107.163.142, index: 11, time: 3315093038,
random: 210936245
*Jan 19 03:10:38.880: SSLVPN: Created gateway cookie:
2154537870@11@3315093038@210936245@ssl-vpn
*Jan 19 03:10:38.900: SSLVPN: Gateway cookie 2154537870@11@3315093038@210936245@ssl-vpn is
converted to ip: 128.107.163.142, gw_index: 11, time: 3315093038, rand: 210936245,
context_name: ssl-vpn
*Jan 19 03:10:39.348: SSLVPN: Gateway cookie 2154537870@11@3315093038@2109362245@ssl-vpn is
converted to ip: 172.107.163.142, gw_index: 11, time: 3315093038, rand: 210936245,
context_name: ssl-vpn
```

The following sample output displays WebVPN cookie information during the browsing of a website that is serving cookies:

```
Router# debug webvpn cookie

*Jan 19 03:12:10.480: SSLVPN: Enter Cookie mangler with Context: 0x0D2B1FA0,
          buffer: 0x0D2EF728, buffer->data: 0x1A8BBF6C, buffer->len: 510,
          cookie: 0x1A8BBFB5, length: 152
*Jan 19 03:12:10.480: SSLVPN: Limited cookie parser element display: Set-Cookie
*Jan 19 03:12:10.480: SSLVPN: Enter Cookie mangler with Context: 0x0D2B1FA0,
          buffer: 0x0D2EF728, buffer->data: 0x1A8BBF6C, buffer->len: 510,
          cookie: 0x1A8BBFC1, length: 140
*Jan 19 03:12:10.480: SSLVPN: Unlimited cookie parser element display: PREF
*Jan 19 03:12:10.480: SSLVPN: Unlimited cookie parser element display:
ID=1554661243d89be2:TM=1106105034:LM=1106105034:S=CqAJHwx2xudAY1YM
*Jan 19 03:12:10.480: SSLVPN: Limited cookie parser element display: expires
*Jan 19 03:12:10.480: SSLVPN: Limited cookie parser element display: Sun, 17-Jan-2038
19:14:07 GMT
*Jan 19 03:12:10.480: SSLVPN: Limited cookie parser element display: path
*Jan 19 03:12:10.480: SSLVPN: Limited cookie parser element display: /
*Jan 19 03:12:10.480: SSLVPN: Limited cookie parser element display: domain
*Jan 19 03:12:10.480: SSLVPN: Limited cookie parser element display: .google.com
*Jan 19 03:12:10.480: SSLVPN: Saved cookie name: PREF
*Jan 19 03:12:10.480: SSLVPN: Created internal cookie: 11@73@33150931300@3318152330
*Jan 19 03:12:10.480: SSLVPN: Saved cookie domain: .google.com
*Jan 19 03:12:10.480: SSLVPN: Saved cookie name: PREF
*Jan 19 03:12:10.480: SSLVPN: Created internal cookie: 11@73@33150931300@3318152330
*Jan 19 03:12:10.480: SSLVPN: Saved cookie domain: .google.com
*Jan 19 03:12:10.484: SSLVPN: Enter Cookie unmangler with Context: 0x0D2B1EB0,
          buffer: 0x0D2EF728, buffer->data: 0x1A8BCD6C, buffer->len: 589,
          cookie: 0x1A8BCEA3, length: 276
*Jan 19 03:12:10.484: SSLVPN: Limited cookie parser element display: Cookie
*Jan 19 03:12:10.484: SSLVPN: Unlimited cookie parser element display: PREF
*Jan 19 03:12:10.484: SSLVPN: Unlimited cookie parser element display:
11@73@33150931300@3318152330
```

```
*Jan 19 03:12:10.484: SSLVPN: Received internal cookie 11@73@3315093130@3318152330 is
converted to gw-index: 11, int-index: 73, time: 3315093130, rand: 3318152330
*Jan 19 03:12:10.484: SSLVPN: Limited cookie parser element display: .google.com
*Jan 19 03:12:10.484: SSLVPN: Cookie domain- unmangled request matched
*Jan 19 03:12:10.484: SSLVPN: Unlimited cookie parser element display:
ID=1554661243d89be2:TM=1106105034:LM=1106105034:S=CqAJHwx2xudAY1YM
*Jan 19 03:12:10.488: SSLVPN: Unlimited cookie parser element display:
CP_GUTC=128.107.163.142.1100045930344008
*Jan 19 03:12:10.488: SSLVPN: Not a mangled internal cookie - ignore
*Jan 19 03:12:10.488: SSLVPN: Limited cookie parser element display:
2154537870@11@3315093038@210936245@ssl-vpn
*Jan 19 03:12:10.488: SSLVPN: Gateway cookie 2154537870@11@3315093038@210936245@ssl-vpn is
converted to ip: 128.107.163.142, gw_index: 11, time: 3315093038, rand: 210936245,
context_name: ssl-vpn
```

The following sample output displays WebVPN HTTP during browsing:

```
Router# debug webvpn http

*Jan 19 03:16:15.164: Original client request
*Jan 19 03:16:15.164: GET /http/0/gmail.google.com/gmail/help/about.html HTTP/1.1
*Jan 19 03:16:15.164:
*Jan 19 03:16:15.164: SSLVPN: HTTP Header parsing complete
*Jan 19 03:16:15.164: SSLVPN: * HTTP request complete
*Jan 19 03:16:15.200: Original server response

*Jan 19 03:16:15.200: HTTP/1.1 200 OK
*Jan 19 03:16:15.200:
*Jan 19 03:16:15.200: SSLVPN: Content type requires mangling
*Jan 19 03:16:15.236: Original client request

*Jan 19 03:16:15.236: GET /http/0/gmail.google.com/gmail/help/images/logo.gif HTTP/1.1
*Jan 19 03:16:15.236:
*Jan 19 03:16:15.236: SSLVPN: HTTP Header parsing complete
*Jan 19 03:16:15.236: SSLVPN: * HTTP request complete
*Jan 19 03:16:15.264: Original server response

*Jan 19 03:16:15.264: HTTP/1.1 200 OK
*Jan 19 03:16:15.264:
*Jan 19 03:16:15.264: SSLVPN: Contents need no mangling, parse no more
*Jan 19 03:16:15.264:  All contents seen in HTTP_RES_PARSE_NOMORE
*Jan 19 03:16:15.264: SSLVPN: Deallocating HTTP info
*Jan 19 03:16:15.276: Original client request

*Jan 19 03:16:15.276: GET
/http/0/gmail.google.com/gmail/help/images/corner_tl_sharp.gif HTTP/1.1 *Jan 19
03:16:15.276:
*Jan 19 03:16:15.276: SSLVPN: HTTP Header parsing complete
*Jan 19 03:16:15.276: SSLVPN: * HTTP request complete
*Jan 19 03:16:15.296: Original server response

*Jan 19 03:16:15.296: HTTP/1.1 200 OK
*Jan 19 03:16:15.296:
*Jan 19 03:16:15.296: SSLVPN: Contents need no mangling, parse no more
*Jan 19 03:16:15.296:  *** Parsing of response body over
*Jan 19 03:16:15.296: SSLVPN: Deallocating HTTP info
```

The following sample output displays WebVPN web service information:

```
Router# debug webvpn webservice

*Jan 19 03:18:39.060: SSLVPN: Date: Wed, 19 Jan 2005 03:18:39 GMT, Expires: Wed, 19 Jan
2005 02:18:39 GMT
```

```
*Jan 19 03:18:39.060: SSLVPN: Created 2608 byte content data to send to external client
for requested file: /webvpn.html
*Jan 19 03:18:39.100: SSLVPN: Date: Wed, 19 Jan 2005 03:18:39 GMT, Expires: Wed, 19 Jan
2005 02:18:39 GMT
*Jan 19 03:18:39.120: SSLVPN: Date: Wed, 19 Jan 2005 03:18:39 GMT, Expires: Wed, 19 Jan
2005 02:18:39 GMT
*Jan 19 03:18:39.120: SSLVPN: Created 2459 byte content data to send to external client
for requested file: /shared.js
*Jan 19 03:18:39.152: SSLVPN: Date: Wed, 19 Jan 2005 03:18:39 GMT, Expires: Wed, 19 Jan
2005 02:18:39 GMT
*Jan 19 03:18:47.496: SSLVPN: Date: Wed, 19 Jan 2005 03:18:47 GMT, Expires: Wed, 19 Jan
2005 02:18:47 GMT
*Jan 19 03:18:47.496: SSLVPN: Created 1375 byte content data to send to external client
for requested file: /logon.html
*Jan 19 03:18:47.516: SSLVPN: HTTP request: 0, path: /paramdef.js
*Jan 19 03:18:47.516: SSLVPN: Date: Wed, 19 Jan 2005 03:18:47 GMT, Expires: Wed, 19 Jan
2005 02:18:47 GMT
*Jan 19 03:18:48.036: SSLVPN: HTTP request: 0, path: /index.html
*Jan 19 03:18:48.036: SSLVPN: Date: Wed, 19 Jan 2005 03:18:48 GMT, Expires: Wed, 19 Jan
2005 02:18:48 GMT
*Jan 19 03:18:48.036: SSLVPN: Created 8269 byte content data to send to external client
for requested file: /index.html
*Jan 19 03:18:48.220: SSLVPN: HTTP request: 0, path: /toolbarframe.html
*Jan 19 03:18:48.220: SSLVPN: Date: Wed, 19 Jan 2005 03:18:48 GMT, Expires: Wed, 19 Jan
2005 02:18:48 GMT
*Jan 19 03:18:48.220: SSLVPN: Created 1312 byte content data to send to external client
for requested file: /toolbarframe.html
*Jan 19 03:18:48.256: SSLVPN: HTTP request: 0, path: /img/logo.gif
*Jan 19 03:18:48.256: SSLVPN: Date: Wed, 19 Jan 2005 03:18:48 GMT, Expires: Wed, 19 Jan
2005 02:18:48 GMT
*Jan 19 03:18:48.268: SSLVPN: HTTP request: 0, path: /test.html
*Jan 19 03:18:48.268: SSLVPN: Date: Wed, 19 Jan 2005 03:18:48 GMT, Expires: Wed, 19 Jan
2005 02:18:48 GMT
*Jan 19 03:18:48.268: SSLVPN: Created 684 byte content data to send to external client for
requested file: /test.html
*Jan 19 03:18:48.316: SSLVPN: HTTP request: 0, path: /toolbar.html
*Jan 19 03:18:48.316: SSLVPN: Date: Wed, 19 Jan 2005 03:18:48 GMT, Expires: Wed, 19 Jan
2005 02:18:48 GMT
*Jan 19 03:18:48.316: SSLVPN: Created 2618 byte content data to send to external client
for requested file: /toolbar.html
*Jan 19 03:18:48.364: SSLVPN: HTTP request: 0, path: /tools.html
*Jan 19 03:18:48.364: SSLVPN: Date: Wed, 19 Jan 2005 03:18:48 GMT, Expires: Wed, 19 Jan
2005 02:18:48 GMT
*Jan 19 03:18:48.364: SSLVPN: Created 2284 byte content data to send to external client
for requested file: /tools.html
```

Troubleshooting WebVPN

The following situations may occur when using the WebVPN feature:

The user is unable to establish an SSL connection to the WebVPN gateway.Verify whether the TCP listeners are correctly created using the **show tcp brief all** command. The listener on port 80 is used for redirecting HTTP connections to HTTPS connections. The listener on port 443 is an HTTPS listener.

```
Router# show tcp brief all

TCB             Local Address     Foreign Address     (state)
652998D8        *.80              *.*                 LISTEN
64CDDE00        *.443             *.*                 LISTEN
6548B37C        *.5060            *.*                 LISTEN
652D3928        *.1723            *.*                 LISTEN
```

If services that use an HTTP secure server are enabled along with WebVPN on the same router, WebVPN must be enabled with the **gateway-addr** keyword option of the **webvpn enable** command by specifying the IP address to enable web VPN service. Two TCP listeners can be seen in the output of the **show tcp brief all** command as follows:

```
Router# show tcp brief all

TCB             Local Address      Foreign Address     (state)
652998D8        *.80               *.*                 LISTEN
64CDDE00        *.443              *.*                 LISTEN
6548B37C        *.5060             *.*                 LISTEN
652D3928        *.1723             *.*                 LISTEN
632988C8        10.1.1.1.80        *.*                 LISTEN
63CDDE1F        10.1.1.1.443       *.*                 LISTEN
```

In the above example, TCP traffic for port 443, which is destined for this router and not on IP address 10.1.1.1, is handled by TCB listener 64CDDE00. Web VPN traffic is handled by TCB listener 63CDDE1F.

If a cookie is not enabled properly on a browser, WebVPN may not work. For example, if a cookie is set to “High” in Internet Explorer (at Tools>Internet Options>Privacy), a user cannot log into WebVPN. In this situation, the cookie has to be set no higher than “Medium High.”

Configuration Examples for WebVPN

This section provides the following configuration examples:

- WebVPN Enabled Globally: Example, page 1737
- WebVPN Enabled on a Specific IP Address: Example, page 1738

WebVPN Enabled Globally: Example

The following is sample running configuration for WebVPN that is enabled globally (on all IP addresses in the system).

```
Router# show running-config

webvpn enable
!
webvpn
 logo file flash:/mylogo.gif
 title-color #FF9933
 text-color black
 ssl encryption 3des-sha1 rc4-md5
 ssl trustpoint WebVPN
 url-list "quicklinks"
   heading "Quicklinks"
   url-text "Meetings and Conferences" url-value
"http://www.mydomain.com/resources/meetings.html"
   url-text "Floor maps" url-value "http://www.mydomain.com/resources/floormaps.html"
   url-text "Documentation" url-value "http://www.mydomain.com/eng/documents"
 url-list "Departments"
   url-text "Engineering" url-value "http://www.mydomain.com/eng"
   url-text "Human Resources" url-value "http://www.mydomain.com/HR"
```

```
   url-text "Sales and Marketing" url-value "http://www.mydomain.com/sandm"
   url-text "Operations" url-value "http://www.mydomain.com/ops"

 login-message "Enter your email-id and password"
!
```

WebVPN Enabled on a Specific IP Address: Example

The following is sample output from a running configuration for WebVPN that is enabled on IP address 10.1.1.1. This configuration also enables e-mail.

```
Router# show running-config

access via port-forwarding.

!
webvpn enable gateway-addr 10.1.1.1
!
webvpn
 logo file flash:/mylogo.gif
 title-color #FF9933
 text-color black
 ssl encryption 3des-sha1 rc4-md5
 ssl trustpoint WebVPN
 url-list "Search"
   heading "Search Engines"
   url-text "Google" url-value "http://www.google.com"
   url-text "Altavista" url-value "http://www.altavista.com"
   url-text "Ask Jeeves" url-value "http://www.askjeeves.com"
 login-message "Enter your email-id and password"
 port-forward list IMAP local-port 60013 remote-server mail.yourdomain.com remote-port 143
 port-forward list POP3 local-port 60014 remote-server mail.yourdomain.com remote-port 25
 port-forward list SMTP local-port 60015 remote-server mail.yourdomain.com remote-port 110
```

Additional References

The following sections provide references related to WebVPN.

Related Documents

Related Topic	Document Title
Cisco IOS security commands	*Cisco IOS Security Command Reference*, Release 12.3T
Other Cisco IOS commands	*Cisco IOS Command Reference*, Release 12.3T

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **debug webvpn**
- **heading**
- **idle-timeout**
- **login-message**
- **logo**
- **port-forward**
- **secondary-color**
- **secondary-text-color**
- **show webvpn sessions**
- **show webvpn statistics**

- **ssl encryption**
- **ssl trustpoint**
- **text-color**
- **title**
- **title-color**
- **url-list**
- **url-text**
- **webvpn**
- **webvpn enable**

Cisco Systems

Part 7: Secure Infrastructure

AutoSecure

By using a single command-line interface (CLI), the AutoSecure feature allows a user to perform the following functions:

- Disable common IP services that can be exploited for network attacks
- Enable IP services and features that can aid in the defense of a network when under attack.

This feature also simplifies the security configuration of a router and hardens the router configuration.

Feature History for AutoSecure

Release	Modification
12.3(1)	This feature was introduced.
12.2(18)S	This feature was integrated into Cisco IOS Release 12.2(18)S.
12.3(8)T	Support for the roll-back functionality and system logging messages were added to Cisco IOS Release 12.3(8)T

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Information About AutoSecure, page 1744
- How to Configure AutoSecure, page 1748
- Configuration Examples for AutoSecure, page 1751
- Additional References, page 1754
- Command Reference, page 1755

Information About AutoSecure

To configure the AutoSecure feature, you should understand the following concepts:

- Benefits of AutoSecure, page 1744
- Secure Management Plane, page 1745
- Secure Forwarding Plane, page 1748

Benefits of AutoSecure

Simplified Router Security Configuration

AutoSecure is valuable to customers without special Security Operations Applications because it allows them to quickly secure their network without thorough knowledge of all the Cisco IOS features.

This feature eliminates the complexity of securing a router by creating a new CLI that automates the configuration of security features and disables certain features enabled by default that could be exploited for security holes.

Enhanced Password Security

AutoSecure provides the following mechanisms to enhance security access to the router:

- The ability to configure a required minimum password length, which can eliminate common passwords that are prevalent on most networks, such as “lab” and “cisco.”

  To configure a minimum password length, use the **security passwords min-length** command.

- Syslog messages are generated after the number of unsuccessful attempts exceeds the configured threshold.

  To configure the number of allowable unsuccessful login attempts (the threshold rate), use the **security authentication failure rate** command.

Roll-Back and System Logging Message Support

In Cisco IOS Release 12.3(8)T, support for roll-back of the AutoSecure configuration is introduced. Roll-back enables a router to revert back to its preautosecure configuration state if the AutoSecure configuration fails.

Note Prior to Cisco IOS Release 12.3(8)T, roll-back of the AutoSecure configuration is unavailable; thus, you should always save the running configuration before configuring AutoSecure.

System Logging Messages capture any changes or tampering of the AutoSecure configuration that were applied on the running configuration. That is, more detailed audit trail information is provided when autosecure is executed.

Secure Management Plane

Securing the management plane is one of two focus areas for the AutoSecure feature. (The other focus area is described in the following section, "Secure Forwarding Plane.") Securing the management plane is done by turning off certain global and interface services that can be potentially exploited for security attacks and turning on global services that help mitigate the threat of attacks. Secure access and secure logging are also configured for the router.

Caution If your device is managed by a network management (NM) application, securing the management plane could turn off some services like HTTP server and disrupt the NM application support.

The following subsections define how AutoSecure helps to secure the management plane:

- Disable Global Services
- Disable Per Interface Services
- Enable Global Services
- Secure Access to the Router
- Log for Security

Disable Global Services

After enabling this feature (via the **auto secure** command), the following global services will be disabled on the router without prompting the user:

- Finger—Collects information about the system (reconnaissance) before an attack. If enabled, the information can leave your device vulnerable to attacks.
- PAD—Enables all packet assembler and disassembler (PAD) commands and connections between PAD devices and access servers. If enabled, it can leave your device vulnerable to attacks.
- Small Servers—Causes TCP and User Datagram Protocol (UDP) diagnostic port attacks: a sender transmits a volume of fake requests for UDP diagnostic services on the router, consuming all CPU resources.
- Bootp Server—Bootp is an insecure protocol that can be exploited for an attack.
- HTTP Server—Without secure-http or authentication embedded in the HTTP server with an associated ACL, the HTTP server is insecure and can be exploited for an attack. (If you must enable the HTTP server, you will be prompted for the proper authentication or access list.)

Note If you are using Security Device Manager (SDM), you must manually enable the HTTP server via the **ip http server** command.

- Identification Service—An unsecure protocol, defined in RFC 1413, that allows one to query a TCP port for identification. An attacker can access private information about the user from the ID server.
- CDP—If a large number of Cisco Discovery Protocol (CDP) packets are sent to the router, the available memory of the router can be consumed, causing the router to crash.

Caution NM applications that use CDP to discover network topology will not be able to perform discovery.

- NTP—Without authentication or access-control, Network Time Protocol (NTP) is insecure and can be used by an attacker to send NTP packets to crash or overload the router. (If you want to turn on NTP, you must configure NTP authentication using Message Digest 5 (MD5) and the **ntp access-group** command. If NTP is enabled globally, disable it on all interfaces on which it is not needed.)
- Source Routing—Provided only for debugging purposes, so source routing should be disabled in all other cases. Otherwise, packets may slip away from some of the access control mechanisms that they should have gone through.

Disable Per Interface Services

After enabling this feature, the following per interface services will be disabled on the router without prompting the user:

- ICMP redirects—Disabled on all interfaces. Does not add a useful functionality to a correctly configured to network, but it could be used by attackers to exploit security holes.
- ICMP unreachables—Disabled on all interfaces. Internet Control Management Protocol (ICMP) unreachables are a known cause for some ICMP-based denial of service (DoS) attacks.
- ICMP mask reply messages—Disabled on all interfaces. ICMP mask reply messages can give an attacker the subnet mask for a particular subnetwork in the internetwork.
- Proxy-Arp—Disabled on all interfaces. Proxy-Arp requests are a known cause for DoS attacks because the available bandwith and resources of the router can be consumed in an attempt to respond to the repeated requests that are sent by an attacker.
- Directed Broadcast—Disabled on all interfaces. Potential cause of SMURF attacks for DoS.
- Maintenance Operations Protocol (MOP) service—Disabled on all interfaces.

Enable Global Services

After enabling this feature, the following global services will be enabled on the router without prompting the user:

- The **service password-encryption** command—Prevents passwords from being visible in the configuration.
- The **service tcp-keepalives-in** and **service tcp-keepalives-out** commands—Ensures that abnormally terminated TCP sessions are removed.

Secure Access to the Router

Caution If your device is managed by an NM application, securing access to the router could turn off vital services and may disrupt the NM application support.

After enabling this feature, the following options in which to secure access to the router are available to the user:

- If a text banner does not exist, users will be prompted to add a banner. This feature provides the following sample banner:

```
Authorized access only
    This system is the property of ABC Enterprise
    Disconnect IMMEDIATELY if you are not an authorized user!
    Contact abc@xyz.com +99 876 543210 for help.
```

- The login and password (preferably a secret password, if supported) are configured on the console, AUX, vty, and tty lines. The **transport input** and **transport output** commands are also configured on all of these lines. (Telnet and secure shell (SSH) are the only valid transport methods.) The **exec-timeout** command is configured on the console and AUX as 10.
- When the image on the device is a crypto image, AutoSecure enables SSH and secure copy (SCP) for access and file transfer to and from the router. The **timeout** *seconds* and **authentication-retries** *integer* options for the **ip ssh** command are configured to a minimum number. (Telnet and FTP are not affected by this operation and remain operational.)
- If the AutoSecure user specifies that their device does not use Simple Network Management Protocol (SNMP), one of the following functionalities will occur:
  - In interactive mode, the user is asked whether to disable SNMP regardless of the values of the community strings, which act like passwords to regulate access to the agent on the router.
  - In non-interact mode, SNMP will be disabled if the community string is “public” or “private.”

Note After AutoSecure has been enabled, tools that use SNMP to monitor or configure a device will be unable to communicate with the device via SNMP.

- If authentication, authorization, and accounting (AAA) is not configured, configure local AAA. Autosecure will prompt users to configure a local username and password on the router.

Log for Security

After this feature is enabled, the following logging options, which allow you to identify and respond to security incidents, are available:

- Sequence numbers and time stamps for all debug and log messages. This option is useful when auditing logging messages.
- Logging messages can be generated for login-related events; for example, the message “ Blocking Period when Login Attack Detected” will be displayed when a login attack is detected and the router enters “quiet mode.” (Quiet mode means that the router will not allows any login attempts via Telnet, HTTP, or SSH.)

  For more information on login system messages, see the Cisco IOS Release 12.3(4)T feature module “Cisco IOS Login Enhancements.”
- The **logging console critical** command, which sends system logging (syslog) messages to all available TTY lines and limits messages based on severity.
- The **logging buffered** command, which copies logging messages to an internal buffer and limits messages logged to the buffer based on severity.
- The **logging trap debugging** command, which allows all commands with a severity higher than debugging to be sent to the logging server.

Secure Forwarding Plane

To minimize the risk of attacks on the router forward plane, AutoSecure provides the following functions:

- Cisco Express Forwarding (CEF)—AutoSecure enables CEF or distributed CEF (dCEF) on the router whenever possible. Because there is no need to build cache entries when traffic starts arriving for new destinations, CEF behaves more predictably than other modes when presented with large volumes of traffic addressed to many destinations. Thus, routers configured for CEF perform better under SYN attacks than routers using the traditional cache.

Note CEF consumes more memory than a traditional cache.

- If the TCP intercept feature is available, it can be configured on the router for connection timeout.
- If strict Unicast Reverse Path Forwarding (uRPF) is available, it can be configured on the router to help mitigate problems that are caused by the introduction of forged (spoofed) IP source addresses. uRPF discards IP packets that lack a verifiable IP source address.
- If the router is being used as a firewall, it can be configured for context-based access control (CBAC) on public interfaces that are facing the Internet.

Note At the beginning of the AutoSecure dialogue, you will be prompted for a list of public interfaces

How to Configure AutoSecure

This section contains the following procedures:

- Configuring AutoSecure, page 1748 (required)
- Configuring Additional Security, page 1749 (required)
- Verifying AutoSecure, page 1750 (optional)

Configuring AutoSecure

To configure AutoSecure, you must perform the following tasks.

The auto secure Command

The **auto secure** command takes you through a semi-interactive session (also known as the AutoSecure dialogue) to secure the management and forwarding planes. This command gives you the option to secure just the management or the forwarding plane; if neither option is selected, the dialogue will ask you to configure both planes.

This command also allows you to go through all noninteractive configuration portions of the dialogue before the interactive portions. The noninteractive portions of the dialogue can be enabled by selecting the optional **no-interact** keyword.

Caution Although the **auto secure** command helps to secure a router, it does not guarantee the complete security of the router.

Restrictions

The AutoSecure configuration can be configured at run time or setup time. If any related configuration is modified after AutoSecure has been enabled, the AutoSecure configuration may not be fully effective.

SUMMARY STEPS

1. **enable**
2. **auto secure** [**management** | **forwarding**] [**no-interact** | **full**] [**ntp** | **login** | **ssh** | **firewall** | **tcp-intercept**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	`enable` **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	`auto secure [management | forwarding] [no-interact | full] [ntp | login | ssh | firewall | tcp-intercept]` **Example:** `Router# auto secure`	Secures the management and forwarding planes of the router. • **management**—Only the management plane will be secured. • **forwarding**—Only the forwarding plane will be secured. • **no-interact**—The user will not be prompted for any interactive configurations. • **full**—The user will be prompted for all interactive questions. This is the default.

Configuring Additional Security

To enable enhanced security access to your router, perform the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **security passwords min-length** *length*
4. **enable password** {*password* | [*encryption-type*] *encrypted-password*}
5. **security authentication failure rate** *threshold-rate* **log**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**security passwords min-length** *length* **Example:** `Router(config)# security passwords min-length 6`	Ensures that all configured passwords are at least a specified length. • *length*—Minimum length of a configured password.
Step 4	**enable password** {*password* | [*encryption-type*] *encrypted-password*} **Example:** `Router(config)# enable password elephant`	Sets a local password to control access to various privilege levels.
Step 5	**security authentication failure rate** *threshold-rate* **log** **Example:** `Router(config)# security authentication failure rate 10 log`	Configures the number of allowable unsuccessful login attempts. • *threshold-rate*—Number of allowable unsuccessful login attempts. • **log**—Syslog authentication failures if the rate exceeds the threshold.

Verifying AutoSecure

To verify that the AutoSecure feature is working successfully, perform the following optional steps:

SUMMARY STEPS

1. **enable**
2. **show auto secure config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**show auto secure config** **Example:** `Router# show auto secure config`	(Optional) Displays all configuration commands that have been added as part of the AutoSecure configuration.

Configuration Examples for AutoSecure

This section provides the following configuration example:

- AutoSecure Configuration Dialogue: Example, page 1751

AutoSecure Configuration Dialogue: Example

The following example is a sample AutoSecure dialogue. After you enable the **auto secure** command, the feature will automatically prompt you with a similar dialogue unless you enable the **no-interact** keyword. (For information on which features are disabled and which features are enabled, see the sections, "Secure Management Plane" and "Secure Forwarding Plane" earlier in this document.)

```
Router# auto secure
                --- AutoSecure Configuration ---

*** AutoSecure configuration enhances the security of the router but it will not make
router absolutely secure from all security attacks ***

All the configuration done as part of AutoSecure will be shown here. For more details of
why and how this configuration is useful, and any possible side effects, please refer to
Cisco documentation of AutoSecure.
At any prompt you may enter '?' for help.
Use ctrl-c to abort this session at any prompt.

Gathering information about the router for AutoSecure

Is this router connected to internet? [no]:y
Enter the number of interfaces facing internet [1]:
Interface                  IP-Address OK? Method Status
Protocol
FastEthernet0/1            10.1.1.1  YES NVRAM  up down

FastEthernet1/0            10.2.2.2  YES NVRAM  up down

FastEthernet1/1            10.0.0.1  YES NVRAM  up up

Loopback0                  unassigned YES NVRAM  up up

FastEthernet0/0            10.0.0.2  YES NVRAM  up down
```

```
Enter the interface name that is facing internet:FastEthernet0/0

Securing Management plane services..

Disabling service finger
Disabling service pad
Disabling udp & tcp small servers
Enabling service password encryption
Enabling service tcp-keepalives-in
Enabling service tcp-keepalives-out
Disabling the cdp protocol

Disabling the bootp server
Disabling the http server
Disabling the finger service
Disabling source routing
Disabling gratuitous arp
Enable secret is either not configured or is same as enable password
Enter the new enable secret:abc123
Configuring aaa local authentication
Configuring console, Aux and vty lines for
local authentication, exec-timeout, transport

Configure SSH server? [yes]:
Enter the domain-name:cisco.com

Configuring interface specific AutoSecure services
Disabling the following ip services on all interfaces:

 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
Disabling mop on Ethernet interfaces

Securing Forwarding plane services..

Enabling CEF (it might have more memory requirements on some low end
platforms)

Enabling unicast rpf on all interfaces connected to internet

Configure CBAC Firewall feature? [yes/no]:yes

This is the configuration generated:

no service finger
no service pad
no service udp-small-servers
no service tcp-small-servers
service password-encryption
service tcp-keepalives-in
service tcp-keepalives-out
no cdp run
no ip bootp server
no ip http server
no ip finger
no ip source-route
no ip gratuitous-arps
no ip identd
security passwords min-length 6
security authentication failure rate 10 log
enable secret 5 $1$CZ6G$GkGOnHdNJCO3CjNHHyTUA.
```

```
aaa new-model
aaa authentication login local_auth local
line console 0
 login authentication local_auth
 exec-timeout 5 0
 transport output telnet
line aux 0
 login authentication local_auth
 exec-timeout 10 0
 transport output telnet
line vty 0 4
 login authentication local_auth
 transport input telnet
ip domain-name cisco.com
crypto key generate rsa general-keys modulus 1024
ip ssh time-out 60
ip ssh authentication-retries 2
line vty 0 4
 transport input ssh telnet
service timestamps debug datetime localtime show-timezone msec
service timestamps log datetime localtime show-timezone msec
logging facility local2
logging trap debugging
service sequence-numbers
logging console critical
logging buffered
int FastEthernet0/1
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
 no mop enabled
int FastEthernet1/0
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
 no mop enabled
int FastEthernet1/1
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
 no mop enabled
int FastEthernet0/0
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
 no mop enabled
ip cef

interface FastEthernet0/0
 ip verify unicast reverse-path
ip inspect audit-trail
ip inspect dns-timeout 7
ip inspect tcp idle-time 14400
ip inspect udp idle-time 1800
ip inspect name autosec_inspect cuseeme timeout 3600
ip inspect name autosec_inspect ftp timeout 3600
```

```
ip inspect name autosec_inspect http timeout 3600
ip inspect name autosec_inspect rcmd timeout 3600
ip inspect name autosec_inspect realaudio timeout 3600
ip inspect name autosec_inspect smtp timeout 3600
ip inspect name autosec_inspect tftp timeout 30
ip inspect name autosec_inspect udp timeout 15
ip inspect name autosec_inspect tcp timeout 3600
access-list 100 deny ip any any
interface FastEthernet0/0
 ip inspect autosec_inspect out
 ip access-group 100 in
!
end


Apply this configuration to running-config? [yes]:yes

Applying the config generated to running-config
The name for the keys will be:ios210.cisco.com

% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys ...[OK]

Router#
```

Additional References

The following sections provide references related to AutoSecure.

Related Documents

Related Topic	Document Title
Login functionality (such as login delays and login blocking periods)	*Cisco IOS Login Enhancements*, Cisco IOS Release 12.3(4)T feature module
Additional information regarding router configuration	*Cisco IOS Configuration Fundamentals Configuration Guide*
Additional router configuration commands	*Cisco IOS Configuration Fundamentals Command Reference*

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 1918	*Address Allocation for Private Internets*
RFC 2267	*Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **auto secure**
- **security authentication failure rate**
- **security passwords min-length**
- **show auto secure config**

Cisco IOS Login Enhancements

The Cisco IOS Login Enhancements feature allows users to better secure their Cisco IOS devices when creating a virtual connection, such as Telnet, secure shell (SSH), or HTTP. Thus, users can help slow down dictionary attacks and help protect their router from a possible denial-of-service (DoS) attack.

Feature History for Cisco IOS Login Enhancements

Release	Modification
12.3(4)T	This feature was introduced.
12.2(25)S	This feature was integrated into Cisco IOS Release 12.2(25)S.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Information About Cisco IOS Login Enhancements, page 1757
- How to Configure Cisco IOS Login Enhancements, page 1759
- Configuration Examples for Login Parameters, page 1762
- Additional References, page 1763
- Command Reference, page 1764

Information About Cisco IOS Login Enhancements

To use login enhancements, you should understand the following concept:

- Login Enhancements Functionality Overview, page 1758

Login Enhancements Functionality Overview

To better configure security when opening a virtual login connection, the following requirements have been added to the login process:

- Delays Between Successive Login Attempts
- Login Shutdown If DoS Attacks Are Suspected
- Generation of System Logging Messages for Login Detection

Delays Between Successive Login Attempts

A Cisco IOS device can accept virtual connections as fast as they can be processed.Introducing a delay between login attempts helps to protect your router from a possible dictionary attack, which attempts to gain access to your username and password information. Delays can be enabled in one of the following ways:

- Via the new global configuration mode command, **login delay**, which allows you to specify a specific number of seconds.
- Via the **login block-for** command. You must enter this command before issuing the **login delay** command; however, if you enter only the **login block-for** command, a login delay of one second is automatically enforced.
- Via the **auto secure** command. If you enable autosecure, a login delay of one second is automatically enforced.

Login Shutdown If DoS Attacks Are Suspected

If the configured number of connection attempts fail within a specified time period, the Cisco IOS device will not accept any additional connections for a “quiet period.” (Hosts that are permitted by a predefined access-control list [ACL] are excluded from the quiet period.)

The number of failed connection attempts that trigger the quiet period can be specified via the new global configuration mode command **login block-for**. The predefined ACL that is excluded from the quiet period can be specified via the new global configuration mode command **login quiet-mode access-class**.

This functionality is disabled by default, and it is not enabled if autosecure if enabled.

Generation of System Logging Messages for Login Detection

After the router switches to and from quiet mode, logging messages are generated. Also, if configured, logging messages are generated upon every successful or failed login request.

Logging messages can be generated for successful login requests via the new global configuration command **login on-success**; the **login on-failure** command generates logs for failed login requests.

Logging messages for failed login attempts are automatically enabled when the **auto secure** command is issued; they are not automatically enabled for successful login attempts via autosecure.

Note Currently, only logging messages can be generated for login-related events. Support for simple network management protocol (SNMP) traps will be added in a later release.

System Logging Messages for a Quiet Period

The following logging message is generated after the router switches to quiet-mode:

```
00:04:07:%SEC_LOGIN-1-QUIET_MODE_ON:Still timeleft for watching failures is 158 seconds,
[user:sfd] [Source:10.4.2.11] [localport:23] [Reason:Invalid login], [ACL:22] at 16:17:23
UTC Wed Feb 26 2003
```

The following logging message is generated after the router switches from quiet mode back to normal mode:

```
00:09:07:%SEC_LOGIN-5-QUIET_MODE_OFF:Quiet Mode is OFF, because block period timed out at
16:22:23 UTC Wed Feb 26 2003
```

System Logging Messages for Successful and Failed Login Requests

The following logging message is generated upon a successful login request:

```
00:04:32:%SEC_LOGIN-5-LOGIN_SUCCESS:Login Success [user:test] [Source:10.4.2.11]
[localport:23] at 20:55:40 UTC Fri Feb 28 2003
```

The following logging message is generated upon a failed login request:

```
00:03:34:%SEC_LOGIN-4-LOGIN_FAILED:Login failed [user:sdfs] [Source:10.4.2.11]
[localport:23] [Reason:Invalid login] at 20:54:42 UTC Fri Feb 28 2003
```

How to Configure Cisco IOS Login Enhancements

This section contains the following procedures:

- Configuring Login Parameters, page 1759 (Required)
- Verifying Login Parameters, page 1761 (Optional)

Configuring Login Parameters

Use this task to configure your Cisco IOS device for login parameters that help detect suspected DoS attacks and slow down dictionary attacks.

Login Parameter Defaults

All login parameters are disabled by default. You must issue the **login block-for** command, which enables default login functionality, before using any other login commands. After the **login block-for** command is enabled, the following defaults are enforced:

- A default login delay of one second
- All login attempts made via Telnet, SSH, and HTTP are denied during the quiet period; that is, no ACLs are exempt from the login period until the **login quiet-mode access-class** command is issued.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **login block-for** *seconds* **attempts** *tries* **within** *seconds*
4. **login quiet-mode access-class** {*acl-name* | *acl-number*}

5. **login delay** *seconds*
6. **login on-failure log** [**every** *login*]
7. **login on-success log** [**every** *login*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**login block-for** *seconds* **attempts** *tries* **within** *seconds* **Example:** `Router(config)# login block-for 100 attempts 2 within 100`	Configures your Cisco IOS device for login parameters that help provide DoS detection. **Note** This command must be issued before any other login command can be used.
Step 4	**login quiet-mode access-class** {*acl-name* | *acl-number*} **Example:** `Router(config)# login quiet-mode access-class myacl`	(Optional) Specifies an ACL that is to be applied to the router when it switches to quiet mode. If this command is not enabled, all login requests will be denied during quiet mode.
Step 5	**login delay** *seconds* **Example:** `Router(config)# login delay 30`	(Optional) Configures a delay between successive login attempts.
Step 6	**login on-failure log** [**every** *login*] **Example:** `Router(config)# login on-failure log`	(Optional) Generates logging messages for failed login attempts.
Step 7	**login on-success log** [**every** *login*] **Example:** `Router(config)# login on-success log every 5`	(Optional) Generates logging messages for successful login attempts.

What to Do Next

After you have configured login parameters on your router, you may wish to verify the settings. To complete this task, see the following section "Verifying Login Parameters."

Verifying Login Parameters

Use this task to verify the applied login configuration and present login status on your router.

SUMMARY STEPS

1. **enable**
2. **show login** [**failures**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show login** [**failures**] **Example:** `Router# show login`	Displays login parameters. • **failures**—Displays information related only to failed login attempts.

Examples

The following sample output from the **show login** command verifies that no login parameters have been specified:

```
Router# show login

No login delay has been applied.
No Quiet-Mode access list has been configured.
All successful login is logged and generate SNMP traps.
All failed login is logged and generate SNMP traps

Router NOT enabled to watch for login Attacks
```

The following sample output from the **show login** command verifies that the **login block-for** command is issued. In this example, the command is configured to block login hosts for 100 seconds if 16 or more login requests fail within 100 seconds; five login requests have already failed.

```
Router# show login

A default login delay of 1 seconds is applied.
No Quiet-Mode access list has been configured.
All successful login is logged and generate SNMP traps.
All failed login is logged and generate SNMP traps.

Router enabled to watch for login Attacks.
If more than 15 login failures occur in 100 seconds or less, logins will be disabled for
100 seconds.

Router presently in Watch-Mode, will remain in Watch-Mode for 95 seconds.
Present login failure count 5.
```

The following sample output from the **show login** command verifies that the router is in quiet mode. In this example, the **login block-for** command was configured to block login hosts for 100 seconds if 3 or more login requests fail within 100 seconds.

```
Router# show login

A default login delay of 1 seconds is applied.
No Quiet-Mode access list has been configured.
All successful login is logged and generate SNMP traps.
All failed login is logged and generate SNMP traps.

Router enabled to watch for login Attacks.
If more than 2 login failures occur in 100 seconds or less, logins will be disabled for
100 seconds.

Router presently in Quiet-Mode, will remain in Quiet-Mode for 93 seconds.
Denying logins from all sources.
```

The following sample output from **show login failures** command shows all failed login attempts on the router:

```
Router# show login failures

Information about login failure's with the device

Username      Source IPAddr  lPort Count  TimeStamp
try1          10.1.1.1        23    1     21:52:49 UTC Sun Mar 9 2003
try2          10.1.1.2        23    1     21:52:52 UTC Sun Mar 9 2003
```

The following sample output from **show login failures** command verifies that no information is presently logged:

```
Router# show login failures

*** No logged failed login attempts with the device.***
```

Configuration Examples for Login Parameters

This section includes the following example:

- Setting Login Parameters: Example, page 1762

Setting Login Parameters: Example

The following example shows how to shows how to configure your router to enter a 100 second quiet period if 15 failed login attempts is exceeded within 100 seconds; all login requests will be denied during the quiet period except hosts from the ACL “myacl.” Also, logging messages will be generated for every 10th failed login and every 15th successful login.

```
Router(config)# login block-for 100 attempts 15 within 100
Router(config)# login quiet-mode access-class myacl
Router(config)# login on-failure log every 10
Router(config)# login on-success log every 15
```

Additional References

The following sections provide references related to Cisco IOS Login Enhancements.

Related Documents

Related Topic	Document Title
AutoSecure	*AutoSecure*, Cisco IOS Release 12.3(1) feature module
Basic configuration information and tasks	*Configuration Fundamentals and Network Management Configuration Guide*
Basic configuration commands	*Configuration Fundamentals and Network Management Command Reference*, Release 12.3 T

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **login block-for**
- **login delay**
- **login on-failure**
- **login on-success**
- **login quiet-mode access-class**
- **show login**

Cisco IOS Resilient Configuration

The Cisco IOS Resilient Configuration feature enables a router to secure and maintain a working copy of the running image and configuration so that those files can withstand malicious attempts to erase the contents of persistent storage (NVRAM and flash).

Feature History for Cisco IOS Resilient Configuration

Release	Modification
12.3(8)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Restrictions for Cisco IOS Resilient Configuration, page 1765
- Information About Cisco IOS Resilient Configuration, page 1766
- How to Use Cisco IOS Resilient Configuration, page 1766
- Additional References, page 1770
- Command Reference, page 1771

Restrictions for Cisco IOS Resilient Configuration

- This feature is available only on platforms that support a Personal Computer Memory Card International Association (PCMCIA) Advanced Technology Attachment (ATA) disk. There must be enough space on the storage device to accommodate at least one Cisco IOS image (two for upgrades) and a copy of the running configuration. IOS Files System (IFS) support for secure file systems is also needed by the software.
- It may be possible to force removal of secured files using an older version of Cisco IOS software that does not contain file system support for hidden files.

- This feature can be disabled only by using a console connection to the router. With the exception of the upgrade scenario, feature activation does not require console access.
- You cannot secure a bootset with an image loaded from the network. The running image must be loaded from persistent storage to be secured as primary.
- Secured files will not appear on the output of a **dir** command issued from an executive shell because the IFS prevents secure files in a directory from being listed. ROM monitor (ROMMON) mode does not have any such restriction and can be used to list and boot secured files. The running image and running configuration archives will not be visible in the Cisco IOS **dir** command output. Instead, use the **show secure bootset** command to verify archive existence.

Information About Cisco IOS Resilient Configuration

Before using Cisco IOS Resilient Configuration, you should understand the following concept:

- Feature Design of Cisco IOS Resilient Configuration, page 1766

Feature Design of Cisco IOS Resilient Configuration

A great challenge of network operators is the total downtime experienced after a router has been compromised and its operating software and configuration data erased from its persistent storage. The operator must retrieve an archived copy (if any) of the configuration and a working image to restore the router. Recovery must then be performed for each affected router, adding to the total network downtime.

The Cisco IOS Resilient Configuration feature is intended to speed up the recovery process. The feature maintains a secure working copy of the router image and the startup configuration at all times. These secure files cannot be removed by the user. This set of image and router running configuration is referred to as the primary bootset.

The following factors were considered in the design of Cisco IOS Resilient Configuration:

- The configuration file in the primary bootset is a copy of the running configuration that was in the router when the feature was first enabled.
- The feature secures the smallest working set of files to preserve persistent storage space. No extra space is required to secure the primary Cisco IOS image file.
- The feature automatically detects image or configuration version mismatch.
- Only local storage is used for securing files, eliminating scalability maintenance challenges from storing multiple images and configurations on TFTP servers.
- The feature can be disabled only through a console session.

How to Use Cisco IOS Resilient Configuration

This section contains the following procedures:

- Archiving a Router Configuration, page 1767
- Restoring an Archived Router Configuration, page 1768

Archiving a Router Configuration

This task describes how to save a primary bootset to a secure archive in persistent storage.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **secure boot-image**
4. **secure boot-config**
5. **end**
6. **show secure bootset**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**secure boot-image** **Example:** `Router(config)# secure boot-image`	Enables Cisco IOS image resilience.
Step 4	**secure boot-config** **Example:** `Router(config)# secure boot-config`	Stores a secure copy of the primary bootset in persistent storage.
Step 5	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.
Step 6	**show secure bootset** **Example:** `Router# show secure bootset`	(Optional) Displays the status of configuration resilience and the primary bootset filename.

Examples

This section provides the following output example:

- Sample Output for the show secure bootset Command, page 1768

Sample Output for the show secure bootset Command

The following example displays sample output from the **show secure bootset** command:

```
Router# show secure bootset

IOS resilience router id JMX0704L5GH

IOS image resilience version 12.3 activated at 08:16:51 UTC Sun Jun 16 2002
Secure archive slot0:c3745-js2-mz type is image (elf) []
  file size is 25469248 bytes, run size is 25634900 bytes
  Runnable image, entry point 0x80008000, run from ram

IOS configuration resilience version 12.3 activated at 08:17:02 UTC Sun Jun 16 2002
Secure archive slot0:.runcfg-20020616-081702.ar type is config
configuration archive size 1059 bytes
```

Restoring an Archived Router Configuration

This task describes how to restore a primary bootset from a secure archive after the router has been tampered with (by an NVRAM erase or a disk format).

Note To restore an archived primary bootset, Cisco IOS image resilience must have been enabled and a primary bootset previously archived in persistent storage.

SUMMARY STEPS

1. **reload**
2. **dir** [*filesystem*:]
3. **boot** [*partition-number*:][*filename*]
4. **no**
5. **enable**
6. **configure terminal**
7. **secure boot-config** [**restore** *filename*]
8. **end**
9. **copy** *filename* **running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**reload** **Example:** `Router# reload`	(Optional) Enters ROM monitor mode, if necessary.
Step 2	**dir** [*filesystem*:] **Example:** `rommon 1 > dir slot0:`	Lists the contents of the device that contains the secure bootset file. • The device name can be found in the output of the **show secure bootset** command.
Step 3	**boot** [*partition-number*:][*filename*] **Example:** `rommon 2 > boot slot0:c3745-js2-mz`	Boots up the router using the secure bootset image.
Step 4	**no** **Example:** `--- System Configuration Dialog ---` `Would you like to enter the initial configuration dialog? [yes/no]: no`	(Optional) Declines to enter an interactive configuration session in setup mode. • If the NVRAM was erased, the router enters setup mode and prompts the user to initiate an interactive configuration session.
Step 5	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 6	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 7	**secure boot-config** [**restore** *filename*] **Example:** `Router(config)# secure boot-config restore slot0:rescue-cfg`	Restores the secure configuration to the supplied filename.
Step 8	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.
Step 9	**copy** *filename* **running-config** **Example:** `Router# copy slot0:rescue-cfg running-config`	Copies the restored configuration to the running configuration.

Additional References

The following sections provide references related to Cisco IOS Resilient Configuration.

Related Documents

Related Topic	Document Title
Additional commands: complete command syntax, command mode, defaults, usage guidelines, and examples	The *Cisco IOS Configuration Fundamentals and Network Management Command Reference*, Release 12.3 T

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **secure boot-config**
- **secure boot-image**
- **show secure bootset**

Image Verification

The Image Verification feature allows users to automatically verify the integrity of Cisco IOS images. Thus, users can be sure that the image is protected from accidental corruption, which can occur at any time during transit, starting from the moment the files are generated by Cisco until they reach the user.

Feature History for Image Verification

Release	Modification
12.2(18)S	This feature was introduced.
12.0(26)S	This feature was integrated into Cisco IOS Release 12.0(26)S.
12.3(4)T	This feature was integrated in Cisco IOS Release 12.3(4)T.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Restrictions for Image Verification, page 1773
- Information About Image Verification, page 1774
- How to Use Image Verification, page 1774
- Configuration Examples for Image Verification, page 1777
- Additional References, page 1779
- Command Reference, page 1780

Restrictions for Image Verification

Cisco IOS Release 12.2(18)S and 12.0(26)S Only

Image Verification is applied to and attempted on any file; however, if the file is not an image file, image verification will not occur and you will see the following error, “SIGNATURE-NOT-FOUND.”

Cisco IOS Release 12.3(4)T Only

Image Verification is applied only to image files. If any other file type is copied or verified, you will not receive a warning that image verification did occur, and the command (copy or verify) will silently succeed.

Information About Image Verification

To use image authentication for your Cisco IOS images, you should understand the following concepts:

- Benefit of Image Verification, page 1774
- How Image Verification Works, page 1774

Benefit of Image Verification

The efficiency of Cisco IOS routers is improved because the routers can now automatically detect when the integrity of an image is accidentally corrupted as a result of transmission errors or disk corruption.

How Image Verification Works

Because a production image undergoes a sequence of transfers before it is copied into the memory of a router, the integrity of the image is at risk of accidental corruption every time a transfer occurs. When downloading an image from Cisco.com, a user can run a message-digest5 (MD5) hash on the downloaded image and verify that the MD5 digest posted on Cisco.com is the same as the MD5 digest that is computed on the user's server. However, many users choose not to run an MD5 digest because it is 128-bits long and the verification is manual. Image verification allows the user to automatically validate the integrity of all downloaded images, thereby, significantly reducing user interaction.

How to Use Image Verification

This section contains the following procedures:

- Globally Verifying the Integrity of an Image, page 1774
- Verifying the Integrity of an Image That Is About to Be Copied, page 1775
- Verifying the Integrity of an Image That Is About to Be Reloaded, page 1776

Globally Verifying the Integrity of an Image

The **file verify auto** command enables image verification globally; that is, all images that are to be copied (via the **copy** command) or reloaded (via the **reload** command) are automatically verified. Although both the **copy** and **reload** commands have a **/verify** keyword that enables image verification, you must issue the keyword each time you want to copy or reload an image. The **file verify auto** command enables image verification by default, so you no longer have to specify image verification multiple times.

If you have enabled image verification by default but prefer to disable verification for a specific image copy or reload, the **/noverify** keyword, along with either the **copy** or the **reload** command, will override the **file verify auto** command.

Use this task to enable automatic image verification.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **file verify auto**
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**file verify auto** **Example:** `Router(config)# file verify auto`	Enables automatic image verification.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode. You must exit global configuration mode if you are going to copy or reload an image.

What to Do Next

After issuing the **file verify auto** command, you do not have to issue the **/verify** keyword with the **copy** or the **reload** command because each image that is copied or reloaded will be automatically verified.

Verifying the Integrity of an Image That Is About to Be Copied

When issuing the **copy** command, you can verify the integrity of the copied file by entering the **/verify** keyword. If the integrity check fails, the copied file will be deleted. If the file that is about to be copied does not have an embedded hash (an old image), you will be prompted whether or not to continue with the copying process. If you choose to continue, the file will be successfully copied; if you choose not to continue, the copied file will be deleted.

Without the **/verify** keyword, the **copy** command could copy a file that is not valid. Thus, after the **copy** command has been successfully executed, you can issue the **verify** command at any time to check the integrity of the files that are in the storage of the router.

Use this task to verify the integrity of an image before it is copied onto a router.

SUMMARY STEPS

1. **enable**
2. **copy** [**/erase**] [**/verify** | **/noverify**] *source-url destination-url*
3. **verify** [**/md5** [*md5-value*]] *filesystem*:[*file-url*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**copy** [**/erase**] [**/verify** | **/noverify**] *source-url destination-url* **Example:** `Router# copy /verify tftp://10.1.1.1/jdoe/c7200-js-mz disk0:`	Copies any file from a source to a destination. • **/verify**—Verifies the signature of the destination file. If verification fails, the file will be deleted. • **/noverify**—Does not verify the signature of the destination file before the image is copied. **Note** **/noverify** is often issued if the **file verify auto** command is enabled, which automatically verifies the signature of all images that are copied.
Step 3	**verify** [**/md5** [*md5-value*]] *filesystem:*[*file-url*] **Example:** `Router# verify bootflash://c7200-kboot-mz.121-8a.E`	(Optional) Verifies the integrity of the images in the router's storage.

Verifying the Integrity of an Image That Is About to Be Reloaded

By issuing the **reload** command with the **/verify** keyword, the image that is about to be loaded onto your system will be checked for integrity. If the **/verify** keyword is specified, image verification will occur before the system initiates the reboot. Thus, if verification fails, the image will not be loaded.

Note Because different platforms obtain the file that is to be loaded in various ways, the file specified in BOOTVAR will be verified. If a file is not specified, the first file on each subsystem will be verified.

On certain platforms, because of variables such as the configuration register, the file that is verified may not be the file that is loaded.

Use this task to verify the integrity of an image before it is reloaded onto a router.

SUMMARY STEPS

1. **enable**
2. **reload** [
   [**warm**] [**/verify** | **/noverify**] *text* |
   [**warm**] [**/verify** | **/noverify**] **in** [*hh*:]*mm* [*text*] |
   [**warm**] [**/verify** | **/noverify**] **at** *hh*:*mm* [*month day* | *day month*] [*text*] |
   [**warm**] [**/verify** | **/noverify**] **cancel**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**reload** [[**warm**] [**/verify** | **/noverify**] *text* | [**warm**] [**/verify** | **/noverify**] **in** [*hh*:]*mm* [*text*] | [**warm**] [**/verify** | **/noverify**] **at** *hh*:*mm* [*month day* | *day month*] [*text*] | [**warm**] [**/verify** | **/noverify**] **cancel**] **Example:** `Router# reload /verify`	Reloads the operating system. • **/verify**—Verifies the signature of the destination file. If verification fails, the file will be deleted. • **/noverify**—Does not verify the signature of the destination file before the image is reloaded. **Note** **/noverify** is often issued if the **file verify auto** command is enabled, which automatically verifies the signature of all images that are copied.

Configuration Examples for Image Verification

This section contains the following configuration examples:

- Global Image Verification: Example, page 1777
- Image Verification via the copy Command: Example, page 1778
- Image Verification via the reload Command: Example, page 1778
- verify Command Sample Output: Example, page 1778

Global Image Verification: Example

The following example shows how to enable automatic image verification. After enabling this command, image verification will automatically occur for all images that are either copied (via the **copy** command) or reloaded (via the **reload** command).

```
Router(config)# file verify auto
```

Image Verification via the copy Command: Example

The following example shows how to specify image verification before copying an image:

```
Router# copy /verify tftp://10.1.1.1/jdoe/c7200-js-mz disk0:

Destination filename [c7200-js-mz]?
Accessing tftp://10.1.1.1/jdoe/c7200-js-mz...
Loading jdoe/c7200-js-mz from 10.1.1.1 (via FastEthernet0/0):!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 19879944 bytes]

19879944 bytes copied in 108.632 secs (183003 bytes/sec)
Verifying file integrity of disk0:/c7200-js-mz
.....................................................................................
.....................................................................................
.....................................................................................
.........................Done!
Embedded Hash MD5 :CFA258948C4ECE52085DCF428A426DCD
Computed Hash    MD5 :CFA258948C4ECE52085DCF428A426DCD
CCO Hash            MD5 :44A7B9BDDD9638128C35528466318183

Signature Verified
```

Image Verification via the reload Command: Example

The following example shows how to specify image verification before reloading an image onto the router:

```
Router# reload /verify

Verifying file integrity of bootflash:c7200-kboot-mz.121-8a.E
%ERROR:Signature not found in file bootflash:c7200-kboot-mz.121-8a.E.
Signature not present. Proceed with verify? [confirm]
Verifying file disk0:c7200-js-mz
.....................................................................
.....................................................Done!
Embedded Hash   MD5 :CFA258948C4ECE52085DCF428A426DCD
Computed Hash   MD5 :CFA258948C4ECE52085DCF428A426DCD
CCO Hash        MD5 :44A7B9BDDD9638128C35528466318183
Signature Verified

Proceed with reload? [confirm]n
```

verify Command Sample Output: Example

The following example shows how to specify image verification via the **verify** command:

```
Router# verify disk0:c7200-js-mz

%Filesystem does not support verify operations
Verifying file integrity of disk0:c7200-js-mz.....................................
.....................................................................................Done!
Embedded Hash   MD5 :CFA258948C4ECE52085DCF428A426DCD
Computed Hash   MD5 :CFA258948C4ECE52085DCF428A426DCD
CCO Hash        MD5 :44A7B9BDDD9638128C35528466318183
```

```
Signature Verified
```

Additional References

The following sections provide references related to Image Verification.

Related Documents

Related Topic	Document Title
Configuration tasks and information for loading, maintaining, and rebooting system images	The section "File Management" in the *Cisco IOS Configuration Fundamentals and Network Management Configuration Guide*
Additional commands for loading, maintaining, and rebooting system images	*Cisco IOS Configuration Fundamentals and Network Management Command Reference*, Release 12.3 T

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

New Command

- **file verify auto**

Modified Commands

- **copy**
- **reload**
- **verify**

IP Source Tracker

The IP Source Tracker feature allows you to gather information about the traffic that is flowing to a host that is suspected of being under attack. This feature also allows you to easily trace an attack to its entry point into the network.

Feature History for IP Source Tracker

Release	Modification
12.0(21)S	This feature was introduced on the Cisco 12000 series.
12.0(22)S	This feature was implemented on the Cisco 7500 series.
12.0(26)S	This feature was implemented on Cisco 12000 series IP Service Engine (ISE) line cards.
12.3(7)T	This feature was integrated into Cisco IOS Release 12.3(7)T.
12.2(25)S	This feature was integrated into Cisco IOS Release 12.2(25)S.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Restrictions for IP Source Tracker, page 1782
- Information About IP Source Tracker, page 1782
- How to Configure IP Source Tracker, page 1784
- Configuration Examples for IP Source Tracker, page 1787
- Additional References, page 1789
- Command Reference, page 1790

Restrictions for IP Source Tracker

Packets Can Be Dropped for Routers

IP source tracking is designed to track attacks against hosts. Packets can be dropped if the line card or port adapter CPU is overwhelmed. Therefore, when used to track an attack against a router, IP source tracking can drop control packets, such as Border Gateway Protocol (BGP) updates.

Engine 0 and 1 Performances Affected on Cisco 12000 Series

There is no performance impact for packets destined to nontracked IP addresses on Engine 2 and Engine 4 line cards because the IP source tracker affects only tracked destinations. Engine 0 and Engine 1 performances are affected because on these engines all packets are switched by the CPU.

Note On Cisco 7500 series routers, there is no performance impact on destinations that are not tracked.

Information About IP Source Tracker

To configure source tracking, you should understand the following concepts:

- Identifying and Tracking Denial of Service Attacks, page 1782
- Using IP Source Tracker, page 1783
- Benefits of IP Source Tracker, page 1784

Identifying and Tracking Denial of Service Attacks

One of the many challenges faced by customers today is the tracking and blocking denial-of-service (DoS) attacks. Counteracting a DoS attack involves intrusion detection, source tracking, and blocking. This functionality addresses the need for source tracking.

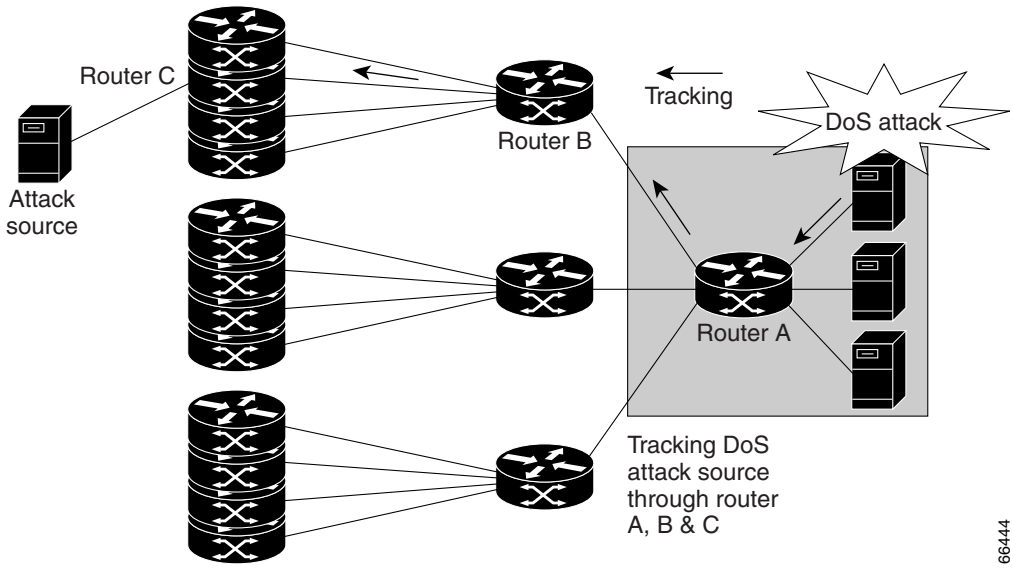
To trace attacks, NetFlow and access control lists (ACLs) have been used. To block attacks, committed access rate (CAR) and ACLs have been used. Support for these features on the Cisco 12000 series Internet router has depended on the type of line card used. Support for these features on the Cisco 7500 series routers depends upon the type of port adapter used. There is, therefore, a need to develop a way to receive information that both traces the source of an attack and is supported on all line cards and port adapters.

Normally, when you identify the host that is subject to a DoS attack, you must determine the network ingress point to effectively block the attack. This process starts at the router closest to the host.

For example, in Figure 124, you would start at Router A and try to determine the next upstream router to examine. Traditionally, you would apply an output ACL to the interface connecting to the host to log packets that match the ACL. The logging information is dumped to the router console or system log. You then have to analyze this information, and possibly go through several ACLs in succession to identify the input interface for the attack. In this case the information points back to Router B.

You then repeat this process on Router B, which leads back to Router C, an ingress point into the network. At this point you can use ACLs or CAR to block the attack. This procedure can require applying several ACLs that generate an excessive amount of output to analyze, making this procedure cumbersome and error prone.

Figure 124 Source Tracking in a DoS Attack



Using IP Source Tracker

IP source tracker provides an easier, more scalable alternative to output ACLs for tracking DoS attacks, and it works as follows:

- After you identify the destination being attacked, enable tracking for the destination address on the whole router by entering the **ip source-track** command.
- Each line card creates a special Cisco Express Forwarding (CEF) entry for the destination address being tracked. For line cards or port adapters that use specialized Application-Specific Integrated Circuit (ASICs) for packet switching, the CEF entry is used to punt packets to the line card's or port adapter's CPU.
- Each line card CPU collects information about the traffic flow to the tracked destination.
- The data generated is periodically exported to the router. To display a summary of the flow information, enter the **show ip source-track summary** command. To display more detailed information for each input interface, enter the **show ip source-track** command.
- Statistics provide a breakdown of the traffic to each tracked IP address. This breakdown allows you to determine which upstream router to analyze next. You can shut down the IP source tracker on the current router by entering the **no ip source-track** command, and reopen it on the upstream router.
- Repeat Step 1 to Step 5 until you identify the source of the attack.
- Apply CAR or ACLs to limit or stop the attack.

IP Source Tracker: Hardware Support

IP source tracking is supported on all Engine 0, 1, 2, and 4 line cards in the Cisco 12000 series Internet router. It is also supported on all port adapters and RSPs that have CEF switching enabled on Cisco 7500 series routers.

Benefits of IP Source Tracker

Complete Tracking Information Provided

IP source tracking generates all the necessary information in an easy-to-use format to track the network entry point of a DoS attack.

Tracking an Unlimited Number of IPs Simultaneously

IP source tracking allows you to track multiple IPs at the same time. By default there is no limit. To limit the number of IPs that are simultaneously tracked, use the **ip source-track address-limit** command.

Complete Network Coverage for Cisco 12000 Series and Cisco 7500 Series Routers as of 12.0(26)S

Because IP source tracking is supported on all line cards on the Cisco 12000 series routers and on all port adapters on Cisco 7500 series routers, it allows you to track DoS attacks across your entire network.

Note For Cisco IOS Release 12.0(21)S and 12.0(22)S, IP source tracking is supported only on Engine 0, 1, 2, and 4 line cards on Cisco 12000 series routers; that is, Engine 3 is not supported.

How to Configure IP Source Tracker

This section contains the following procedures:

- Configuring IP Source Tracking, page 1784 (required)
- Verifying IP Source Tracking, page 1785 (optional)

Configuring IP Source Tracking

To configure IP source tracking for a host under attack, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip source-track** *ip-address*
4. **ip source-track address-limit** *number*
5. **ip source-track syslog-interval** *number*
6. **ip source-track export-interval** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip source-track** *ip-address* **Example:** `Router(config)# ip source-track 100.10.0.1`	Enables IP source tracking for a specified host.
Step 4	**ip source-track address-limit** *number* **Example:** `Router(config)# ip source-track address-limit 10`	(Optional) Limits the number of hosts that can be simultaneously tracked at any given time. **Note** If this command is not enabled, there is no limit to the number of hosts that be can tracked.
Step 5	**ip source-track syslog-interval** *number* **Example:** `Router(config)# ip source-track syslog-interval 2`	(Optional) Sets the time interval, in minutes, used to generate syslog messages that indicate IP source tracking is enabled. **Note** If this command is not enabled, system log messages are not generated.
Step 6	**ip source-track export-interval** *number* **Example:** `Router(config)# ip source-track export-interval 30`	(Optional) Sets the time interval, in seconds, used to export IP tracking statistics that are collected in the line cards to the gigabit route processor (GRP) and the port adapters to the route switch processor (RSP). **Note** If this command is not enabled, traffic flow information is exported to the GRP and RSP every 30 seconds.

What to Do Next

After you have configured source tracking on your network device, you can verify your configuration and source tracking statistics, such as traffic flow. To complete this task, see the following section "Verifying IP Source Tracking."

Verifying IP Source Tracking

To verify the status of source tracking, such as packet processing and traffic flow information, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show ip source-track** [*ip-address*] [**summary** | **cache**]
3. **show ip source-track export flows**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip source-track** [*ip-address*] [**summary** | **cache**] **Example:** `Router# show ip source-track summary`	Displays traffic flow statistics for tracked IP host addresses
Step 3	**show ip source-track export flows** **Example:** `Router# show ip source-track export flows`	Displays the last 10 packet flows that were exported from the line card to the route processor. **Note** This command can be issued only on distributed platforms, such as the GRP and the RSP.

Examples

The following example, which is sample output from the **show ip source-track summary** command, shows how to verify that IP source tracking is enabled for one or more hosts:

```
Router# show ip source-track summary

Address          Bytes    Pkts    Bytes/s   Pkts/s
10.0.0.1          119G   1194M    443535      4432
192.168.1.1       119G   1194M    443535      4432
192.168.42.42     119G   1194M    443535      4432
```

The following example, which is sample output from the **show ip source-track summary** command, shows how to verify that no traffic has yet to be received for the destination hosts that are being tracked:

```
Router# show ip source-track summary

Address        Bytes   Pkts   Bytes/s   Pkts/s
10.0.0.1           0      0         0        0
192.168.1.1        0      0         0        0
192.168.42.42      0      0         0        0
```

The following example, which is sample output from the **show ip source-track** command, shows how to verify that IP source tracking is processing packets to the hosts and exporting statistics from the line card or port adapter to the GRP and RSP:

```
Router# show ip source-track

Address         SrcIF    Bytes   Pkts   Bytes/s   Pkts/s
10.0.0.1        PO0/0    119G   1194M    513009     5127
192.168.1.1     PO0/0    119G   1194M    513009     5127
192.168.42.42   PO0/0    119G   1194M    513009     5127
```

Configuration Examples for IP Source Tracker

This section includes the following examples:

- Configuring IP Source Tracking: Example, page 1787
- Verifying Source Interface Statistics for All Tracked IP Addresses: Example, page 1787
- Verifying a Flow Statistic Summary for All Tracked IP Addresses: Example, page 1788
- Verifying Detailed Flow Statistics Collected by a Line Card: Example, page 1788
- Verifying Flow Statistics Exported from Line Cards and Port Adapters: Example, page 1788

Configuring IP Source Tracking: Example

The following example shows how to configure IP source tracking on all line cards and port adapters in the router. In this example, each line card or port adapter collects traffic flow data to host address 100.10.0.1 for 2 minutes before creating an internal system log entry; packet and flow information recorded in the system log is exported for viewing to the route processor or switch processor every 60 seconds.

```
Router# configure interface
Router(config)# ip source-track 100.10.0.1
Router(config)# ip source-track syslog-interval 2
Router(config)# ip source-track export-interval 60
```

Verifying Source Interface Statistics for All Tracked IP Addresses: Example

The following example displays a summary of the traffic flow statistics that are collected on each source interface for tracked host addresses.

```
Router# show ip source-track

Address         SrcIF     Bytes    Pkts     Bytes/s     Pkts/s
10.0.0.1        PO2/0         0       0           0          0
192.168.9.9     PO1/2      131M    511M        1538          6
192.168.9.9     PO2/0      144G   3134M     6619923     143909
```

Verifying a Flow Statistic Summary for All Tracked IP Addresses: Example

The following example displays a summary of traffic flow statistics for all hosts that are being tracked; it shows that no traffic has yet been received.

```
Router# show ip source-track summary

Address              Bytes    Pkts     Bytes/s     Pkts/s
10.0.0.1                 0       0           0          0
100.10.1.1            131M    511M        1538          6
192.168.9.9           146G   3178M     6711866     145908
```

Verifying Detailed Flow Statistics Collected by a Line Card: Example

The following example displays traffic flow information that is collected on line card 0 for all tracked hosts.

```
Router# exec slot 0 show ip source-track cache

========= Line Card (Slot 0) =======

IP packet size distribution (7169M total packets):
   1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
   .000 .000 .000 0.00 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

    512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
   .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 278544 bytes
  1 active, 4095 inactive, 13291 added
  198735 ager polls, 0 flow alloc failures
  Active flows timeout in 0 minutes
  Inactive flows timeout in 15 seconds
  last clearing of statistics never
Protocol         Total    Flows   Packets Bytes  Packets Active(Sec) Idle(Sec)
--------         Flows     /Sec     /Flow  /Pkt     /Sec     /Flow     /Flow

SrcIf          SrcIPaddress    DstIf          DstIPaddress    Pr TOS Flgs  Pkts
Port Msk AS                    Port Msk AS    NextHop              B/Pk  Active
PO0/0          101.1.1.0       Null           100.1.1.1       06 00  00      55K
0000 /0  0                     0000 /0  0     0.0.0.0               100    10.1
```

Verifying Flow Statistics Exported from Line Cards and Port Adapters: Example

The following example displays packet flow information that is exported from line cards and port adapters to the GRP and the RSP:

```
Router# show ip source-track export flows

SrcIf         SrcIPaddress    DstIf         DstIPaddress    Pr SrcP DstP  Pkts
PO0/0         101.1.1.0       Null          100.1.1.1       06 0000 0000    88K
PO0/0         101.1.1.0       Null          100.1.1.3       06 0000 0000    88K
PO0/0         101.1.1.0       Null          100.1.1.2       06 0000 0000    88K
```

Additional References

The following sections provide references related to IP Source Tracker.

Related Documents

Related Topic	Document Title
ACLs	The section "Filtering IP Packets Using Access Lists" in the chapter "Configuring IP Services" of the *Cisco IOS IP Configuration Guide*
Dynamic ACLs	The chapter "Configuring Lock-and-Key Security (Dynamic Access Lists)" in the *Cisco IOS Security Configuration Guide*
DoS prevention	The chapter "Configuring TCP Intercept (Preventing Denial-of-Service Attacks)" in the *Cisco IOS Security Configuration Guide*

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **ip source-track**
- **ip source-track address-limit**
- **ip source-track export-interval**
- **ip source-track syslog-interval**
- **show ip source-track**
- **show ip source-track export flows**

IP Traffic Export

The IP Traffic Export feature allows users to configure their router to export IP packets that are received on multiple, simultaneous WAN or LAN interfaces. The unaltered IP packets are exported on a single LAN or VLAN interface, thereby, easing deployment of protocol analyzers and monitoring devices.

Feature History for IP Traffic Export

Release	Modification
12.3(4)T	This feature was introduced.
12.2(25)S	This feature was integrated into Cisco IOS Release 12.2(25)S.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Restrictions for IP Traffic Export, page 1791
- Information About IP Traffic Export, page 1792
- How to Use IP Traffic Export, page 1792
- Configuration Examples for IP Traffic Export, page 1796
- Additional References, page 1800
- Command Reference, page 1801

Restrictions for IP Traffic Export

Platform Restriction

IP traffic export is intended only for software switching platforms; distributed architectures are not supported.

IP Packet Forwarding Performance Impact

When IP traffic export is enabled, a delay is incurred on the outbound interface when packets are captured and transmitted across the interface. Performance delays increase with the increased number of interfaces that are monitored and the increased number of destination hosts.

Exported Traffic Limitation

- The MAC address of the device that is receiving the exported traffic must be on the same VLAN or directly connected to one of the router interfaces. (Use the **show arp** command to determine the MAC address of device that is directly connected to an interface.)
- The outgoing interface for exported traffic must be Ethernet (10/100/1000). (Incoming (monitored) traffic can traverse any interface.)

Information About IP Traffic Export

To use the IP traffic export, you should understand the following concept:

- Benefits of IP Traffic Export, page 1792

Benefits of IP Traffic Export

Simplified IDS Deployment

Without the ability to export IP traffic, the Intrusion Detection System (IDS) probe must be inline with the network device to monitor traffic flow. IP traffic export eliminates the probe placement limitation, allowing users to place an IDS probe in any location within their network or direct all exported traffic to a VLAN that is dedicated for network monitoring. Allowing users to choose the optimal location of their IDS probe reduces processing burdens.

Also, because packet processing that was once performed on the network device can now be performed away from the network device, the need to enable IDS with the Cisco IOS software can be elimintaed.

IP Traffic Export Functionality Benefits

Users can configure their router to perform the following tasks:

- Filter copied packets via an access control list (ACL)
- Filter copied packets via sampling, which allows you to export one in every few packets in which you are interested. Use this option when it is not necessary to export all incoming traffic. Also, sampling is useful when a monitored ingress interface can send traffic faster than the egress interface can transmit it.
- Configure bidirectional traffic on an interface. (By default, only incoming traffic is exported.)

How to Use IP Traffic Export

This section contains the following procedures:

- Configuring IP Traffic Export, page 1793
- Displaying IP Traffic Export Configuration Data, page 1795

Configuring IP Traffic Export

Use this task to configure IP traffic export profiles, which enable IP traffic to be exported on an ingress interface and allow you to specify profile attributes, such as the outgoing interface for exporting traffic.

Note Packet exporting is performed before packet switching or filtering.

IP Traffic Export Profiles Overview

All packet export configurations are specified via IP traffic export profiles, which consist of IP-traffic-export-related command-line interfaces (CLIs) that control various attributes for both incoming and outgoing exported IP traffic. You can configure a router with multiple IP traffic export profiles. (Each profile must have a different name.) You can apply different profiles on different interfaces.

The two different IP traffic export profiles are as follows:

- The global configuration profile, which is configured via the **ip traffic-export profile** command.
- The IP traffic export submode configuration profile, which is configured via any of the following router IP Traffic Export (RITE) commands—**bidirectional**, **incoming**, **interface**, **mac-address**, and **outgoing**.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip traffic-export profile** *profile-name*
4. **interface** *interface-name*
5. **bidirectional**
6. **mac-address** *H.H.H*
7. **incoming** {**access-list** {*standard* | *extended* | *named*} | **sample one-in-every** *packet-number*}
8. **outgoing** {**access-list** {*standard* | *extended* | *named*} | **sample one-in-every** *packet-number*}
9. **exit**
10. **interface** *type number*
11. **ip traffic-export apply profile** *profile-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip traffic-export profile** *profile-name* **Example:** `Router(config)# ip traffic-export profile my_rite`	Creates or edits an IP traffic export profile, enables the profile on an ingress interface, and enters RITE configuration mode.
Step 4	**interface** *interface-name* **Example:** `Router(config-rite)# interface FastEthernet 0/1`	Specifies the outgoing (monitored) interface for exported traffic. **Note** If you do not issue this command, the profile will not recognize an interface in which to send the captured IP traffic.
Step 5	**bidirectional** **Example:** `Router(config-rite)# bidirectional`	(Optional) Exports incoming and outgoing IP traffic on the monitored interface. **Note** If this command is not enabled, only incoming traffic is exported.
Step 6	**mac-address** *H.H.H* **Example:** `Router(config-rite)# mac-address 00a.8aab.90a0`	Specifies the 48-bit address of the destination host that is receiving the exported traffic. **Note** If you do not issue this command, the profile will not recognize a destination host in which to send the exported packets.
Step 7	**incoming** {**access-list** {*standard* | *extended* | *named*} | **sample one-in-every** *packet-number*} **Example:** `Router(config-rite)# incoming access-list my_acl`	(Optional) Configures filtering for incoming traffic. After you have created a profile via the **ip traffic-export profile,** this functionality is enabled by default.
Step 8	**outgoing** {**access-list** {*standard* | *extended* | *named*} | **sample one-in-every** *packet-number*} **Example:** `Router(config-rite)# outgoing sample one-in-every 50`	(Optional) Configures filtering for outgoing export traffic. **Note** If you issue this command, you must also issue the **bidirectional** command, which enables outgoing traffic to be exported. However, only routed traffic (such as passthrough traffic) is exported; that is, traffic that originates from the network device is not exported.
Step 9	**exit**	Exits RITE configuration mode.

	Command or Action	Purpose
Step 10	**interface** *type number* **Example:** `Router(config)# interface FastEthernet0/0`	Configures an interface type and enters interface configuration mode.
Step 11	**ip traffic-export apply profile** *profile-name* **Example:** `Router(config-if)# ip traffic-export apply profile my_rite`	Enables IP traffic export on an ingress interface.

Troubleshooting Tips

Creating an IP Traffic Export Profile

The **interface** and **mac-address** commands are required to successfully create a profile. If these commands are not issued, you will receive the following profile incomplete message if the **show running config** command is issued:

```
ip traffic-export profile newone
! No outgoing interface configured
! No destination mac-address configured
```

Applying an IP Traffic Export Profile to an interface

The following system logging messages should appear immediately after you activate and deactivate a profile from an interface (via the **ip traffic-export apply profile** command):

- Activated profile:

  ```
  %RITE-5-ACTIVATE: Activated IP traffic export on interface FastEthernet 0/0.
  ```

- Deactivated profile:

  ```
  %RITE-5-DEACTIVATE: Deactivated IP traffic export on interface FastEthernet 0/0.
  ```

If you attempt to apply an incomplete profile to an interface, you will receive the following message:

```
Router(config-if)# ip traffic-export apply newone
RITE: profile newone has missing outgoing interface
```

What to Do Next

After you have configured a profile and enabled the profile on an ingress interface, you can monitor IP traffic exporting events and verify your profile configurations. To complete these steps, refer to the following task "Displaying IP Traffic Export Configuration Data."

Displaying IP Traffic Export Configuration Data

This task allows you to verify IP traffic export parameters such as the monitored ingress interface, which is where the IP traffic is exported, and outgoing and incoming IP packet information, such as configured ACLs. You can also use this task to monitor packets that are captured and then transmitted across an interface to a destination host. Use this optional task to help you troubleshoot any problems with your exported IP traffic configurations.

SUMMARY STEPS

1. **enable**
2. **debug ip traffic-export events**
3. **show ip traffic-export** [**interface** *interface-name* | **profile** *profile-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug ip traffic-export events** **Example:** `Router# debug ip traffic-export events`	Enables debugging messages for exported IP traffic packets events.
Step 3	**show ip traffic-export** [**interface** *interface-name* | **profile** *profile-name*] **Example:** `Router# show ip traffic-export`	Displays information related to exported IP traffic events. • **interface** *interface-name*—Only data associated with the monitored ingress interface is shown. • **profile** *profile-name*—Only flow statistics, such as exported packets and the number of bytes, are shown.

Examples

The following sample output from the **show ip traffic-export** command is for the profile "one." This example is for a single, configured interface. If multiple interfaces are configured, the information shown below is displayed for each interface.

```
Router# show ip traffic-export

Router IP Traffic Export Parameters
Monitored Interface             FastEthernet0/0
        Export Interface                FastEthernet0/1
        Destination MAC address 0030.7131.abfc
        bi-directional traffic export is off
Input IP Traffic Export Information     Packets/Bytes Exported    0/0
        Packets Dropped           0
        Sampling Rate             one-in-every 1 packets
        No Access List configured
        Profile one is Active
```

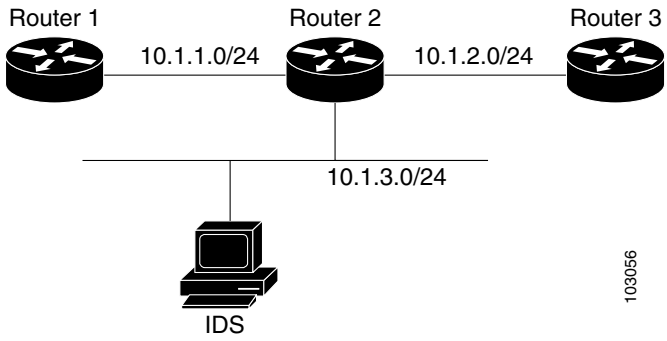
Configuration Examples for IP Traffic Export

This section includes the following configuration example:

- Exporting IP Traffic Configuration: Example, page 1797

Exporting IP Traffic Configuration: Example

Figure 1 and the following sample output from the **show running-config** command illustrate how to configure Router 2 to export the incoming traffic from Router 1 to IDS:




```
Router# show running-config

Building configuration...

Current configuration :2349 bytes
!
! Last configuration change at 20:35:39 UTC Wed Oct 8 2003
! NVRAM config last updated at 20:35:39 UTC Wed Oct 8 2003
!
version 12.3
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
service udp-small-servers
!
hostname rite-3745
!
boot system flash:c3745-js-mz.123-1.8.PI2d
no logging console
enable password lab
!
no aaa new-model
ip subnet-zero
!
!
no ip domain lookup
!
!
ip cef
!
ip traffic-export profile a92340)__-2304109%((#((%(#
! This export profile is not complete [missing outgoing interface name]
!
ip traffic-export profile myprofile
  interface FastEthernet1/0.1
  mac-address 6666.6666.3333
mpls ldp logging neighbor-changes
no scripting tcl init
no scripting tcl encdir
!
!
!
```

```
!
!
!
!
!
!
!
no voice hpi capture buffer
no voice hpi capture destination
!
!
!
!
controller T1 0/0
 framing sf
 linecode ami
!
controller T1 0/1
 framing sf
 linecode ami
!
controller T1 4/0
 framing sf
 linecode ami
!
controller T1 4/1
 framing sf
 linecode ami
!
!
!
interface FastEthernet0/0
 ip address 10.0.0.94 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 mac-address 6666.6666.4444
 ip address 10.1.1.2 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet1/0
 load-interval 30
 duplex auto
 speed auto
!
interface FastEthernet1/0.1
 encapsulation dot1Q 202
 ip address 10.1.3.2 255.255.255.0
 no ip redirects
 no cdp enable
!
interface Serial1/0
 shutdown
 clockrate 125000
 no fair-queue
!
interface FastEthernet1/1
 mac-address 6666.6666.5555
 ip address 10.1.2.2 255.255.255.0
 duplex auto
 speed auto
!
```

```
interface Serial1/1
 shutdown
 clockrate 125000
!
router ospf 100
 log-adjacency-changes
 network 10.1.0.0 0.0.255.255 area 0
!
ip http server
ip classless
!
!
!
ip access-list standard IP-1
 permit 1.1.1.0 0.0.0.255
!
ip access-list extended IP-2
 permit ip 2.2.2.0 0.0.0.255 any
ip access-list extended UDP
 permit udp any any
ip access-list extended liang
access-list 1 permit any
access-list 101 permit udp any any
snmp-server engineID local 0000000902000004C1C59140
snmp-server community public RO
snmp-server enable traps tty
!
!
!
!
control-plane
!
!
!
!
!
!
dial-peer cor custom
!
!
!
gateway
!
!
line con 0
 exec-timeout 0 0
 stopbits 1
line aux 0
line vty 0 4
 password lab
 login
!
ntp clock-period 17175608
ntp server 10.0.0.2
!
end
```

Additional References

The following sections provide references related to IP Traffic Export.

Related Documents

Related Topic	Document Title
Configuring IDS	The chapter "Configuring Cisco IOS Firewall Intrusion Detection System" in the section "Traffic Filtering and Firewalls" of the *Cisco IOS Security Configuration Guide*.
Configuring IP	The chapter "Configuring IP Services" in the section "IP Addressing and Services" of the *Cisco IOS IP Configuration Guide*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **bidirectional**
- **debug ip traffic-export events**
- **incoming**
- **interface (RITE)**
- **ip traffic-export apply profile**
- **ip traffic-export profile**
- **mac-address (RITE)**
- **outgoing**

Role-Based CLI Access

The Role-Based CLI Access feature allows the network administrator to define "views," which are a set of operational commands and configuration capabilities that provide selective or partial access to Cisco IOS EXEC and configuration (Config) mode commands. Views restrict user access to Cisco IOS command-line interface (CLI) and configuration information; that is, a view can define what commands are accepted and what configuration information is visible. Thus, network administrators can exercise better control over access to Cisco networking devices.

Feature History for Role-Based CLI Access

Release	Modification
12.3(7)T	This feature was introduced.
12.3(11)T	The CLI view capability was extended to restrict user access on a per-interface level, and additional CLI views were introduced to support the extended view capability. Also, support to group configured CLI views into a superview was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for Role-Based CLI Access, page 1804
- Restrictions for Role-Based CLI Access, page 1804
- Information About Role-Based CLI Access, page 1804
- How to Use Role-Based CLI Access, page 1805
- Configuration Examples for Role-Based CLI Access, page 1811
- Additional References, page 1814
- Command Reference, page 1815

Prerequisites for Role-Based CLI Access

Your image must support CLI views.

Restrictions for Role-Based CLI Access

Lawful Intercept Images Limitation

Because CLI views are a part of the Cisco IOS parser, CLI views are a part of all platforms and Cisco IOS images. However, the lawful intercept view is available only in images that contain the lawful intercept subsystem.

Maximum Number of Allowed Views

The maximum number of CLI views and superviews, including one lawful intercept view, that can be configured is 15. (This does not include the root view.)

Information About Role-Based CLI Access

To create and use views, you should understand the following concepts:

- Benefits of Using CLI Views, page 1804
- Root View, page 1804
- View Authentication via a New AAA Attribute, page 1805

Benefits of Using CLI Views

Views: Detailed Access Control

Although users can control CLI access via both privilege levels and enable mode passwords, these functions do not provide network administrators with the necessary level of detail needed when working with Cisco IOS routers and switches. CLI views provide a more detailed access control capability for network administrators, thereby, improving the overall security and accountability of Cisco IOS software.

As of Cisco IOS Release 12.3(11)T, network administrators can also specify an interface or a group of interfaces to a view; thereby, allowing access on the basis of specified interfaces.

Root View

When a system is in “root view,” it has all of the access privileges as a user who has level 15 privileges. If the administrator wishes to configure any view to the system (such as a CLI view, a superview, or a lawful intercept view), the system must be in root view.

The difference between a user who has level 15 privileges and a root view user is that a root view user can configure a new view and add or remove commands from the view. Also, when you are in a CLI view, you have access only to the commands that have been added to that view by the root view user.

View Authentication via a New AAA Attribute

View authentication is performed by an external authentication, authorization, and accounting (AAA) server via the new attribute "cli-view-name."

AAA authentication associates only one view name to a particular user; that is, only one view name can be configured for a user in an authentication server.

How to Use Role-Based CLI Access

This section contains the following procedures:

- Configuring a CLI View, page 1805 (required)
- Configuring a Lawful Intercept View, page 1807 (optional)
- Configuring a Superview, page 1809 (optional)
- Monitoring Views and View Users, page 1811 (optional)

Configuring a CLI View

Use this task to create a CLI view and add commands or interfaces to the view, as appropriate.

Prerequisites

Before you create a view, you must perform the following tasks:

- Enable AAA via the **aaa new-model** command. (For more information on enabling AAA, see the chapter "Configuring Authentication" in the *Cisco IOS Security Configuration Guide*, Release 12.3.
- Ensure that your system is in root view—not privilege level 15.

SUMMARY STEPS

1. **enable view**
2. **configure terminal**
3. **parser view** *view-name*
4. **secret 5** *encrypted-password*
5. **commands** *parser-mode* {**include** | **include-exclusive** | **exclude**} [**all**] [**interface** *interface-name* | *command*]
6. **exit**
7. **exit**
8. **enable** [*privilege-level*] [**view** *view-name*]
9. **show parser view** [**all**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable view** **Example:** `Router> enable view`	Enables root view. • Enter your privilege level 15 password (for example, root password) if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**parser view** *view-name* **Example:** `Router(config)# parser view first`	Creates a view and enters view configuration mode.
Step 4	**secret 5** encrypted-*password* **Example:** `Router(config-view)# secret 5 secret`	Associates a command-line interface (CLI) view or superview with a password. **Note** You must issue this command before you can configure additional attributes for the view.
Step 5	**commands** *parser-mode* {**include** | **include-exclusive** | **exclude**} [**all**] [**interface** *interface-name* | *command*] **Example:** `Router(config-view)# commands exec include show version`	Adds commands or interfaces to a view. • *parser-mode*—The mode in which the specified command exists. • **include**—Adds a command or an interface to the view and allows the same command or interface to be added to an additional view. • **include-exclusive**—Adds a command or an interface to the view and excludes the same command or interface from being added to all other views. • **exclude**—Excludes a command or an interface from the view; that is, customers cannot access a command or an interface. • **all**—A "wildcard" that allows every command in a specified configuration mode that begins with the same keyword or every subinterface for a specified interface to be part of the view. • **interface** *interface-name*— Interface that is added to the view. • *command*—Command that is added to the view.
Step 6	**exit** **Example:** `Router(config-view)# exit`	Exits view configuration mode.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 8	**enable** [*privilege-level*] [**view** *view-name*] **Example:** `Router# enable view first`	Prompts the user for a password, which allows the user to access a configured CLI view, and is used to switch from one view to another view. After the correct password is given, the user can access the view.
Step 9	**show parser view** [**all**] **Example:** `Router# show parser view`	(Optional) Displays information about the view that the user is currently in. • **all**—Displays information for all views that are configured on the router. **Note** Although this command is available for both root and lawful intercept users, the **all** keyword is available only to root users. However, the **all** keyword can be configured by a user in root view to be available for users in lawful intercept view and CLI view.

Troubleshooting Tips

After you have successfully created a view, a system message such as the following will be displayed:

```
%PARSER-6-VIEW_CREATED: view 'first' successfully created.
```

After you have successfully deleted a view, a system message such as the following will be displayed:

```
%PARSER-6-VIEW_DELETED: view 'first' successfully deleted.
```

You must associate a password with a view. If you do not associate a password, and you attempt to add commands to the view via the **commands** command, a system message such as the following will be displayed:

```
%Password not set for view <viewname>.
```

Configuring a Lawful Intercept View

Use this task to initialize and configure a view for lawful-intercept-specific commands and configuration information. (Only an administrator or a user who has level 15 privileges can initialize a lawful intercept view.)

About Lawful Intercept Views

Like a CLI view, a lawful intercept view restricts access to specified commands and configuration information. Specifically, a lawful intercept view allows a user to secure access to lawful intercept commands that are held within the TAP-MIB, which is a special set of simple network management protocol (SNMP) commands that store information about calls and users.

Commands available in lawful intercept view belong to one of the following categories:

- Lawful intercept commands that should not be made available to any other view or privilege level
- CLI views that are useful for lawful intercept users but do not have to be excluded from other views or privilege levels

Prerequisites

Before you initialize a lawful intercept view, ensure that the privilege level is set to 15 via the **privilege** command.

SUMMARY STEPS

1. **enable view**
2. **configure terminal**
3. **li-view** *li-password* **user** *username* **password** *password*
4. **username** [**lawful-intercept**] *name* [**privilege** *privilege-level* | **view** *view-name*] **password** *password*
5. **parser view** *view-name*
6. **secret 5** *encrypted-password*
7. **name** *new-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable view** **Example:** `Router> enable view`	Enables root view. • Enter your privilege level 15 password (for example, root password) if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**li-view** *li-password* **user** *username* **password** *password* **Example:** `Router(config)# li-view lipass user li_admin password li_adminpass`	Initializes a lawful intercept view. After the li-view is initialized, you must specify at least one user via **user** *username* **password** *password* options.
Step 4	**username** [**lawful-intercept** [*name*] [**privilege** *privilege-level* | **view** *view-name*] **password** *password* **Example:** `Router(config)# username lawful-intercept li-user1 password li-user1pass`	Configures lawful intercept users on a Cisco device.

	Command or Action	Purpose
Step 5	**parser view** *view-name* **Example:** `Router(config)# parser view li view name`	(Optional) Enters view configuration mode, which allows you to change the lawful intercept view password or the lawful intercept view name.
Step 6	**secret 5** encrypted-*password* **Example:** `Router(config-view)# secret 5 secret`	(Optional) Changes an existing password for a lawful intercept view.
Step 7	**name** *new-name* **Example:** `Router(config-view)# name second`	(Optional) Changes the name of a lawful intercept view. If this command is not issued, the default name of the lawful intercept view is "li-view."

Troubleshooting Tips

To display information for all users who have access to a lawful intercept view, issue the **show users lawful-intercept** command. (This command is available only to authorized lawful intercept view users.)

Configuring a Superview

Use this task to create a superview and add at least one CLI view to the superview.

About Superviews

A superview consists of one or more CLI views, which allow users to define what commands are accepted and what configuration information is visible. Superviews allow a network administrator to easily assign all users within configured CLI views to a superview instead of having to assign multiple CLI views to a group of users.

Superviews contain the following characteristics:

- A CLI view can be shared among multiple superviews.
- Commands cannot be configured for a superview; that is, you must add commands to the CLI view and add that CLI view to the superview.
- Users who are logged into a superview can access all of the commands that are configured for any of the CLI views that are part of the superview.
- Each superview has a password that is used to switch between superviews or from a CLI view to a superview.
- If a superview is deleted, all CLI views associated with that superview will not be deleted too.

Adding CLI Views to a Superview

You can add a view to a superview only after a password has been configured for the superview (via the **secret 5** command). Thereafter, issue the **view** command in view configuration mode to add at least one CLI view to the superview.

Note Before adding a CLI view to a superview, ensure that the CLI views that are added to the superview are valid views in the system; that is, the views have been successfully created via the **parser view** command.

SUMMARY STEPS

1. **enable view**
2. **configure terminal**
3. **parser view** *superview-name* **superview**
4. **secret 5** *encrypted-password*
5. **view** *view-name*
6. **exit**
7. **exit**
8. **show parser view** [**all**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable view** **Example:** `Router> enable view`	Enables root view. • Enter your privilege level 15 password (for example, root password) if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**parser view** *superview-name* **superview** **Example:** `Router(config)# parser view su_view1 superview`	Creates a superview and enters view configuration mode.
Step 4	**secret 5** *encrypted-password* **Example:** `Router(config-view)# secret 5 secret`	Associates a command-line interface (CLI) view or superview with a password. **Note** You must issue this command before you can configure additional attributes for the view.
Step 5	**view** *view-name* **Example:** `Router(config-view)# view view_three`	Adds a normal CLI view to a superview. Issue this command for each CLI view that is to be added to a given superview.
Step 6	**exit** **Example:** `Router(config-view)# exit`	Exits view configuration mode.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 8	**show parser view** [**all**] **Example:** `Router# show parser view`	(Optional) Displays information about the view that the user is currently in. • **all**—Displays information for all views that are configured on the router. **Note** Although this command is available for both root and lawful intercept users, the **all** keyword is available only to root users. However, the **all** keyword can be configured by a user in root view to be available for users in lawful intercept view and CLI view.

Monitoring Views and View Users

To display debug messages for all views—root, CLI, lawful intercept, and super, use the **debug parser view** command in privileged EXEC mode.

Configuration Examples for Role-Based CLI Access

This section contains the following configuration examples:

- Configuring a CLI View: Example, page 1811
- Verifying a CLI View: Example, page 1812
- Configuring a Lawful Intercept View: Example, page 1813
- Configuring a Superview: Example, page 1814

Configuring a CLI View: Example

The following example show how to configure two CLI views, “first” and “second.” Thereafter, you can verify the CLI view in the running configuration.

```
Router(config)# parser view first
00:11:40:%PARSER-6-VIEW_CREATED:view 'first' successfully created.
Router(config-view)# secret 5 firstpass
Router(config-view)# command exec include show version
Router(config-view)# command exec include configure terminal
Router(config-view)# command exec include all show ip
Router(config-view)# exit
Router(config)# parser view second
00:13:42:%PARSER-6-VIEW_CREATED:view 'second' successfully created.
Router(config-view)# secret 5 secondpass
Router(config-view)# command exec include-exclusive show ip interface
Router(config-view)# command exec include logout
Router(config-view)# exit
!
```

```
!
Router(config-view)# do show run | beg view
parser view first
 secret 5 $1$MCmh$QuZaU8PIMPlff9sFCZvgW/
 commands exec include configure terminal
 commands exec include configure
 commands exec include all show ip
 commands exec include show version
 commands exec include show
!
parser view second
 secret 5 $1$iP2M$R16BXKecMEiQesxLyqygW.
 commands exec include-exclude show ip interface
 commands exec include show ip
 commands exec include show
 commands exec include logout
!
```

Verifying a CLI View: Example

After you have configured the CLI views "first" and "second," you can issue the **enable view** command to verify which commands are available in each view. The following example shows which commands are available inside the CLI view "first" after the user has logged into this view. (Because the **show ip** command is configured with the all option, a complete set of suboptions is shown, except the **show ip interface** command, which is using the include-exclusive keyword in the second view.)

```
Router# enable view first
Password:

00:28:23:%PARSER-6-VIEW_SWITCH:successfully set to view 'first'.
Router# ?
Exec commands:
  configure  Enter configuration mode
  enable     Turn on privileged commands
  exit       Exit from the EXEC
  show       Show running system information

Router# show ?

  ip       IP information
  parser   Display parser information
  version  System hardware and software status

Router# show ip ?

  access-lists            List IP access lists
  accounting              The active IP accounting database
  aliases                 IP alias table
  arp                     IP ARP table
  as-path-access-list     List AS path access lists
  bgp                     BGP information
  cache                   IP fast-switching route cache
  casa                    display casa information
  cef                     Cisco Express Forwarding
  community-list          List community-list
  dfp                     DFP information
  dhcp                    Show items in the DHCP database
  drp                     Director response protocol
  dvmrp                   DVMRP information
  eigrp                   IP-EIGRP show commands
  extcommunity-list       List extended-community list
  flow                    NetFlow switching
```

```
  helper-address          helper-address table
  http                    HTTP information
  igmp                    IGMP information
  irdp                    ICMP Router Discovery Protocol
.
.
.
```

Configuring a Lawful Intercept View: Example

The following example shows how to configure a lawful intercept view, add users to the view, and verify the users that were added:

```
!Initialize the LI-View.
Router(config-view)# li-view lipass user li_admin password li_adminpass
00:19:25:%PARSER-6-LI_VIEW_INIT:LI-View initialized.
Router(config-view)# end

! Enter the LI-View; that is, check to see what commands are available within the view.
Router# enable view li-view
Password:

Router#
00:22:57:%PARSER-6-VIEW_SWITCH:successfully set to view 'li-view'.
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# parser view li-view
Router(config-view)# ?
View commands:
  commands  Configure commands for a view
  default   Set a command to its defaults
  exit      Exit from view configuration mode
  name      New LI-View name      ===This option only resides in LI View.
  no        Negate a command or set its defaults
  password  Set a password associated with CLI views

Router(config-view)#

! NOTE:LI View configurations are never shown as part of 'running-configuration'.

! Configure LI Users.
Router(config)# username lawful-intercept li-user1 password li-user1pass
Router(config)# username lawful-intercept li-user2 password li-user2pass

! Displaying LI User information.
Router# show users lawful-intercept

li_admin
li-user1
li-user2
Router#
```

Configuring a Superview: Example

The following sample output from the **show running-config** command shows that “view_one” and “view_two” have been added to superview “su_view1,” and “view_three” and “view_four” have been added to superview “su_view2”:

```
!
parser view su_view1 superview
 secret 5 <encoded password>
 view view_one
 view view_two
!
parser view su_view2 superview
 secret 5 <encoded password>
 view view_three
 view view_four
!
```

Additional References

The following sections provide references related to Role-Based CLI Access.

Related Documents

Related Topic	Document Title
SNMP, MIBs, CLI configuration	*Cisco IOS Configuration Fundamentals and Network Management Configuration Guide*, Release 12.3
Privilege levels	*Cisco IOS Security Configuration Guide*, Release 12.3

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new and modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **commands (view)**
- **debug parser view**
- **enable**
- **li-view**
- **name (view)**
- **parser view**
- **parser view superview**
- **secret 5**
- **show parser view**
- **show users**
- **username**
- **view**

Security with Passwords, Privilege Levels, and Login Usernames for CLI Sessions on Networking Devices

This part consists of the following:

- Configuring Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices
- No Service Password-Recovery

Configuring Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices

Cisco IOS based networking devices provide several features that can be used to implement basic security for CLI sessions using only the operating system running on the device. These features include the following:

- Different levels of authorization for CLI sessions to control access to commands that can modify the status of the networking device versus commands that are used to monitor the device
- Assigning passwords to CLI sessions
- Requiring users log in to a networking device with a username
- Changing the privilege levels of commands to create new authorization levels for CLI sessions.

This module is a guide to implementing a baseline level of security for your networking devices. It focuses on the least complex options available for implementing a baseline level of security. If you have networking devices installed in your network with no security options configured, or you are about to install a networking device and you need help understanding the how to implement a baseline of security, this document will help you.

Module History

This module was first published on May 2nd, 2005, and last updated on May 2nd, 2005.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all features. To find information about feature support and configuration, use the "Feature Information for Configuring Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices" section on page 1858.

Contents

- Restrictions for Configuring Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices, page 1820
- Information About Configuring Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices, page 1820
- How To Configure Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices, page 1832

- Configuration Examples for Configuring Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices, page 1853
- Where to Go Next, page 1856
- Additional References, page 1857
- Feature Information for Configuring Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices, page 1858

Restrictions for Configuring Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices

Your networking device must not be configured to use any local or remote authentication, authorization, and accounting (AAA) security features. This document describes only the non-AAA security features that can be configured locally on the networking device.

For information how to configure AAA security features that can be run locally on a networking device, or for information on how to configure remote AAA security using TACACS+ or RADIUS servers, see the *Cisco IOS Security Configuration Guide,* Release 12.4.

Information About Configuring Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices

To configure router security with passwords, CLI privilege levels and usernames, you should understand the following concepts:

- Benefits of Creating a Security Scheme for Your Networking Device, page 1820
- Cisco IOS CLI Modes, page 1821
- Cisco IOS CLI Sessions, page 1828
- Protect Access to Cisco IOS EXEC Modes, page 1829
- Cisco IOS Password Encryption Levels, page 1829
- Cisco IOS CLI Session Usernames, page 1831
- Cisco IOS Privilege Levels, page 1831
- Cisco IOS Password Configuration, page 1832

Benefits of Creating a Security Scheme for Your Networking Device

The foundation of a good security scheme in the network is the protection of the user interfaces of the networking devices from unauthorized access. Protecting access to the user interfaces on your networking devices prevents unauthorized users from making configuration changes that can disrupt the stability of your network or compromise your network security.

The Cisco IOS features described in this document can be combined in many different ways to create a unique security scheme for each of your networking devices. Here are some possible examples that you can configure:

- You can enable non administrative users to run a subset of the administrative commands available on the networking device by lowering the entitlement level for the commands to the non administrative privilege level. This can be useful for the following scenarios:
  - ISPs that want their first-line technical support staff to perform tasks such as enabling new interfaces for new customers or resetting the connection for a customer whose connection has stopped passing traffic. See the "Configuring and Verifying a Networking Device to Allow Non Administrative Users to Shutdown and Enable Interfaces: Example" section on page 1855 section for an example of how to do this.
  - When you want your first-line technical support staff to have the ability to clear console port sessions that were disconnected improperly from a terminal server. See the "Configuring and Verifying a Networking Device to Allow Non Administrative Users to Clear Remote CLI Sessions: Example" section on page 1853 section for an example of how to do this.
  - When you want your first-line technical support staff to have the ability to view, but not change, the configuration of a networking device to facilitate troubleshooting a networking problem. See the "Configuring and Verifying a Networking Device to Allow Non Administrative Users to View the Running Configuration Automatically: Example" section on page 1855 section for an example of how to do this.

Cisco IOS CLI Modes

To aid in the configuration of Cisco devices, the Cisco IOS command-line interface is divided into different command modes. Each command mode has its own set of commands available for the configuration, maintenance, and monitoring of router and network operations. The commands available to you at any given time depend on the mode you are in. Entering a question mark (**?**) at the system prompt (router prompt) allows you to obtain a list of commands available for each command mode.

The use of specific commands allows you to navigate from one command mode to another. The standard order in which a user would access the modes is as follows: user EXEC mode; privileged EXEC mode; global configuration mode; specific configuration modes; configuration submodes; and configuration subsubmodes.

Note The default configuration of a Cisco IOS software based networking device only allows you to configure passwords to protect access to user EXEC mode (for local, and remote CLI sessions) and privileged EXEC mode. This document describes how you can provide additional levels of security by protecting access to other modes, and commands, using a combination of usernames, passwords and the **privilege** command.

Most EXEC mode commands are one-time commands, such as **show** or **more** commands, which show the current configuration status, and **clear** commands, which clear counters or interfaces. EXEC mode commands are not saved across reboots of the router.

From privileged EXEC mode, you can enter *global configuration mode*. In this mode, you can enter commands that configure general system characteristics. You also can use global configuration mode to enter specific configuration modes. Configuration modes, including global configuration mode, allow you to make changes to the running configuration. If you later save the configuration, these commands are stored across router reboots.

From global configuration mode you can enter a variety of protocol-specific or feature-specific configuration modes. The CLI hierarchy requires that you enter these specific configuration modes only through global configuration mode. For example, *interface configuration mode*, is a commonly used configuration mode.

From configuration modes, you can enter configuration submodes. Configuration submodes are used for the configuration of specific features within the scope of a given configuration mode. As an example, this chapter describes the *subinterface configuration mode*, a submode of the interface configuration mode.

ROM monitor mode is a separate mode used when the router cannot boot properly. If your system (router, switch, or access server) does not find a valid system image to load when it is booting, the system will enter ROM monitor mode. ROM monitor (ROMMON) mode can also be accessed by interrupting the boot sequence during startup. ROMMON is not covered in this document because it does not have any security features available in it.

The following sections contain detailed information on these command modes:

- User EXEC Mode
- Privileged EXEC Mode
- Global Configuration Mode
- Interface Configuration Mode
- Subinterface Configuration Mode

User EXEC Mode

When you start a session on a router, you generally begin in *user EXEC mode*, which is one of two access levels of the EXEC mode. For security purposes, only a limited subset of EXEC commands are available in user EXEC mode. This level of access is reserved for tasks that do not change the configuration of the router, such as determining the router status.

If your device is configured to require users to log-in the log-in process will require a username and a password. You may try three times to enter a password before the connection attempt is refused.

User EXEC mode is set by default to privilege level 1. Privileged EXEC mode is set by default to privilege level 15. For more information see the "Privileged EXEC Mode" section on page 1823. When you are logged into a networking device in user EXEC mode your session is running at privilege level 1. By default the EXEC commands at privilege level 1 are a subset of those available at privilege level 15. When you are logged into a networking device in privileged EXEC mode your session is running at privilege level 15. You can move commands to any privilege level between 1 and 15 using the **privilege** command. See the "Cisco IOS Privilege Levels" section on page 1831 for more information on privilege levels and the **privilege** command.

In general, the user EXEC commands allow you to connect to remote devices, change terminal line settings on a temporary basis, perform basic tests, and list system information.

To list the available user EXEC commands, use the following command:

Command	Purpose
`Router(config)# ?`	Lists the user EXEC mode commands

The user EXEC mode prompt consists of the host name of the device followed by an angle bracket (>), as shown in the following example:

```
Router>
```

The default host name is generally `Router,` unless it has been changed during initial configuration using the **setup** EXEC command. You also change the host name using the **hostname** global configuration command.

Note Examples in Cisco IOS documentation assume the use of the default name of "Router." Different devices (for example, access servers) may use a different default name. If the routing device (router, access server, or switch) has been named with the **hostname** command, that name will appear as the prompt instead of the default name.

To list the commands available in user EXEC mode, enter a question mark (**?**) as shown in the following example:

```
Router> ?
Exec commands:
 <1-99>           Session number to resume
 connect          Open a terminal connection
 disconnect       Disconnect an existing telnet session
 enable           Turn on privileged commands
 exit             Exit from Exec mode
 help             Description of the interactive help system
 lat              Open a lat connection
 lock             Lock the terminal
 login            Log in as a particular user
 logout           Exit from Exec mode and log out
 menu             Start a menu-based user interface
 mbranch          Trace multicast route for branch of tree
 mrbranch         Trace reverse multicast route to branch of tree
 mtrace           Trace multicast route to group
 name-connection  Name an existing telnet connection
 pad              Open a X.29 PAD connection
 ping             Send echo messages
 resume           Resume an active telnet connection
 show             Show running system information
 systat           Display information about terminal lines
 telnet           Open a telnet connection
 terminal         Set terminal line parameters
 tn3270           Open a tn3270 connection
 trace            Trace route to destination
 where            List active telnet connections
 x3               Set X.3 parameters on PAD
```

The list of commands will vary depending on the software feature set and router platform you are using.

Note You can enter commands in uppercase, lowercase, or mixed case. Only passwords are case sensitive. However, Cisco IOS documentation convention is to always present commands in lowercase.

Privileged EXEC Mode

In order to have access to all commands, you must enter *privileged EXEC mode*, which is the second level of access for the EXEC mode. Normally, you must enter a password to enter privileged EXEC mode. In privileged EXEC mode, you can enter any EXEC command, because privileged EXEC mode is a superset of the user EXEC mode commands.

Because many privileged EXEC mode commands set operating parameters, privileged EXEC level access should be password protected to prevent unauthorized use. The privileged EXEC command set includes those commands contained in user EXEC mode. Privileged EXEC mode also provides access to configuration modes through the **configure** command, and includes advanced testing commands, such as **debug**.

Privileged EXEC mode is set by default to privilege level 15. User EXEC mode is set by default to privilege level 1. For more information see the "User EXEC Mode" section on page 1822. When you are logged into a networking device in privileged EXEC mode your session is running at privilege level 15. When you are logged into a networking device in user EXEC mode your session is running at privilege level 1. By default the EXEC commands at privilege level 15 are a superset of those available at privilege level 1. You can move commands to any privilege level between 1 and 15 using the **privilege** command. See the "Cisco IOS Privilege Levels" section on page 1831 for more information on privilege levels and the **privilege** command.

The privileged EXEC mode prompt consists of the host name of the device followed by a pound sign(#), as shown in the following example:

```
Router#
```

To access privileged EXEC mode, use the following command:

Command	Purpose
`Router> enable` `Password` `Router# exit` `Router>`	Enables privileged EXEC mode. • If a privileged EXEC mode password has been configured the system will prompt you for a password after you issue the enable command. • Use the exit command to leave privileged EXEC mode.

Note Privileged EXEC mode is sometimes referred to as "enable mode," because the **enable** command is used to enter the mode.

If a password has been configured on the system, you will be prompted to enter it before being allowed access to privileged EXEC mode. The password is not displayed on the screen and is case sensitive. If an enable password has not been set, privileged EXEC mode can be accessed only by a local CLI session (terminal connected to the console port).

If you attempt to access privileged EXEC mode on a router over a remote connection, such as a telnet connection, and you have not configured a password for privileged EXEC mode you will see the **% No password set** error message. For more information on remote connections see the "Remote CLI Sessions" section on page 1828. The system administrator uses the **enable secret** or **enable password** global configuration commands to set the password that restricts access to privileged EXEC mode. For information on configuring a password for privileged EXEC mode, see the "Protecting Access to Privileged Exec Mode" section on page 1837.

To return to user EXEC mode, use the following command:

Command	Purpose
`Router# disable`	Exits from privileged EXEC mode to user EXEC mode.

The following example shows the process of accessing privileged EXEC mode:

```
Router> enable
Password:<letmein>
Router#
```

Note that the password will not be displayed as you type, but is shown here for illustrational purposes.To list the commands available in privileged EXEC mode, issue the **?** command at the prompt. From privileged EXEC mode you can access global configuration mode, which is described in the following section.

Note Because the privileged EXEC command set contains all of the commands available in user EXEC mode, some commands can be entered in either mode. In Cisco IOS documentation, commands that can be entered in either user EXEC mode or privileged EXEC mode are referred to as EXEC mode commands. If user or privileged is not specified in the documentation, assume that you can enter the referenced commands in either mode.

Global Configuration Mode

The term "global" is used to indicate characteristics or features that affect the system as a whole. Global configuration mode is used to configure your system globally, or to enter specific configuration modes to configure specific elements such as interfaces or protocols. Use the **configure terminal** privileged EXEC command to enter global configuration mode.

To access global configuration mode, use the following command in privileged EXEC mode:

Command	Purpose
`Router# configure terminal`	From privileged EXEC mode, enters global configuration mode.

The following example shows the process of entering global configuration mode from privileged EXEC mode:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
```

Note that the system prompt changes to indicate that you are now in global configuration mode. The prompt for global configuration mode consists of the host-name of the device followed by `(config)` and the pound sign (`#`). To list the commands available in privileged EXEC mode, issue the **?** command at the prompt.

Commands entered in global configuration mode update the running configuration file as soon as they are entered. In other words, changes to the configuration take effect each time you press the Enter or Return key at the end of a valid command. However, these changes are not saved into the startup configuration file until you issue the **copy running-config startup-config** EXEC mode command. This behavior is explained in more detail later in this document.

As shown in the example above, the system dialogue prompts you to end your configuration session (exit configuration mode) by pressing the Control (Ctrl) and "z" keys simultaneously; when you press these keys, **^Z** is printed to the screen. You can actually end your configuration session by entering the Ctrl-Z key combination, using the **end** command, using the Ctrl-C key combination. The **end** command is the recommended way to indicate to the system that you are done with the current configuration session.

Caution If you use Ctrl-Z at the end of a command line in which a valid command has been typed, that command will be added to the running configuration file. In other words, using Ctrl-Z is equivalent to hitting the Enter (Carriage Return) key before exiting. For this reason, it is safer to end your configuration session using the **end** command. Alternatively, you can use the Ctrl-C key combination to end your configuration session without sending a Carriage Return signal.

You can also use the **exit** command to return from global configuration mode to EXEC mode, but this only works in global configuration mode. Pressing Ctrl-Z or entering the **end** command will always take you back to EXEC mode regardless of which configuration mode or configuration submode you are in.

To exit global configuration command mode and return to privileged EXEC mode, use one of the following commands:

Command	Purpose
`Router(config)# end` or `Router(config)# ^Z`	Ends the current configuration session and returns to privileged EXEC mode.
`Router(config)# exit`	Exits the current command mode and returns to the preceding mode. For example, exits from global configuration mode to privileged EXEC mode.

From global configuration mode, you can enter a number of protocol-specific, platform-specific, and feature-specific configuration modes.

Interface configuration mode, described in the following section, is an example of a configuration mode you can enter from global configuration mode.

Interface Configuration Mode

One example of a specific configuration mode you enter from global configuration mode is interface configuration mode.

Many features are enabled on a per-interface basis. Interface configuration commands modify the operation of an interface such as an Ethernet, FDDI, or serial port. Interface configuration commands always follow an **interface** global configuration command, which defines the interface type.

For details on interface configuration commands that affect general interface parameters, such as bandwidth or clock rate, refer to the Release 12.2 *Cisco IOS Interface Configuration Guide*. For protocol-specific commands, refer to the appropriate Cisco IOS software command reference.

To access and list the interface configuration commands, use the following command:

Command	Purpose
`Router(config)# interface` *type number*	Specifies the interface to be configured, and enters interface configuration mode.

In the following example, the user enters interface configuration mode for serial interface 0. The new prompt, *hostname*`(config-if)#`, indicates interface configuration mode.

```
Router(config)# interface serial 0
Router(config-if)#
```

To exit interface configuration mode and return to global configuration mode, enter the **exit** command.

Configuration submodes are configuration modes entered from other configuration modes (besides global configuration mode). Configuration submodes are for the configuration of specific elements within the configuration mode. One example of a configuration submode is subinterface configuration mode, described in the following section.

Subinterface Configuration Mode

From interface configuration mode, you can enter subinterface configuration mode. Subinterface configuration mode is a submode of interface configuration mode. In subinterface configuration mode you can configure multiple virtual interfaces (called subinterfaces) on a single physical interface. Subinterfaces appear to be distinct physical interfaces to the various protocols.

For detailed information on how to configure subinterfaces, refer to the appropriate documentation module for a specific protocol in the Cisco IOS software documentation set.

To access subinterface configuration mode, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# interface` *type number*	Specifies the virtual interface to be configured and enters subinterface configuration mode.

In the following example, a subinterface is configured for serial line 2, which is configured for Frame Relay encapsulation. The subinterface is identified as "2.1" to indicate that it is subinterface 1 of serial interface 2. The new prompt *hostname*`(config-subif)#` indicates subinterface configuration mode. The subinterface can be configured to support one or more Frame Relay PVCs.

```
Router(config)# interface serial 2
Router(config-if)# encapsulation frame-relay
Router(config-if)# interface serial 2.1
Router(config-subif)#
```

To exit subinterface configuration mode and return to interface configuration mode, use the **exit** command. To end your configuration session and return to privileged EXEC mode, press Ctrl-Z or enter the **end** command.

Cisco IOS CLI Sessions

This section describes the following concepts:

- Local CLI Sessions, page 1828
- Remote CLI Sessions, page 1828
- Terminal Lines are Used for Local and Remote CLI Sessions, page 1828

Local CLI Sessions

Local CLI sessions require direct access to the the console port of the networking device. Local CLI sessions start in user EXEC mode. See the "Cisco IOS CLI Modes" section on page 1821 for more information on the different modes that are supported on your networking device. All of the tasks required to configure and manage a networking device can be done using a local CLI session. The most common method for establishing a local CLI session is to connect the serial port on a PC to the console port of the networking device and then to launch a terminal emulation application on the PC. The type of cable and connectors required and the settings for the terminal emulation application on the PC are dependant on the type of networking device that you are configuring. See to the documentation for your networking device for more information on setting it up for a local CLI session.

Remote CLI Sessions

Remote CLI sessions are created between a host such as a PC and a networking device such as a router over a network using a remote terminal access application such as Telnet and Secure Shell (SSH). Local CLI sessions start in user EXEC mode. See the "Cisco IOS CLI Modes" section on page 1821 for more information on the different modes that are supported on your networking device. Most of the tasks required to configure and manage a networking device can be done using a remote CLI session. The exceptions are tasks that interact directly with the console port (such as recovering from a corrupted operating system (OS) by uploading a new OS image over the console port) and interacting with the networking device when it is in ROM Monitor Mode.

This document explains how to configure security for remote Telnet sessions. Telnet is the most common method for accessing a remote CLI session on a networking device.

Note SSH is a more secure alternative to Telnet. SSH provides encryption for the session traffic between your local management device such as a PC and the networking device that you are managing. Encrypting the session traffic with SSH prevents hackers that might intercept the traffic from being able to decode it. See *Secure Shell Version 2 Support* (http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_feature_guide09186a00802045dc.html) for more information on using SSH.

Terminal Lines are Used for Local and Remote CLI Sessions

Cisco networking devices use the word lines to refer to the software components that manage local and remote CLI sessions. You use the **line console 0** global configuration command to enter line configuration mode to configure options, such as a password, for the console port.

```
Router# configure terminal
Router(config)# line console 0
Router(config-line)# password password-string
```

Remote CLI sessions use lines that are referred to virtual teletypewriter (VTY) lines. You use the **line vty** ***line-number*** [***ending-line-number***] global configuration command to enter line configuration mode to configure options, such as a password, for remote CLI sessions.

```
Router# configure terminal
Router(config)# line vty 0 4
Router(config-line)# password password-string
```

Protect Access to Cisco IOS EXEC Modes

Cisco IOS provides the ability to configure passwords that protect access to the following:

- Protecting Access to User EXEC Mode, page 1829
- Protecting Access to Privileged EXEC mode, page 1829

Protecting Access to User EXEC Mode

The first step in creating a secure environment for your networking device is protecting access to user EXEC mode by configuring passwords for local and remote CLI sessions.

You protect access to user EXEC mode for local CLI sessions by configuring a password on the console port. See the "Configuring and Verifying a Password for Local CLI Sessions" section on page 1835.

You protect access to user EXEC mode for remote CLI sessions by configuring a password on the virtual terminal lines (VTYs). See the "Configuring and Verifying a Password for Remote CLI Sessions" section on page 1833 for instructions on how to configure passwords for remote CLI sessions.

Protecting Access to Privileged EXEC mode

The second step in creating a secure environment for your networking device is protecting access to privileged EXEC mode with a password. The method for protecting access to privileged EXEC mode is the same for local and remote CLI sessions.

You protect access to privileged EXEC mode by configuring a password for it. This is sometimes referred to as the enable password because the command to enter privileged EXEC mode is **enable**.

Command	Purpose
enable **Example:** `Router> enable` `Password` `Router#`	Enables privileged EXEC mode. • Enter your password if prompted. The password will not be shown in the terminal window. • The ">" at the end of the prompt string is changed to a "#" to indicate that you are in privileged EXEC mode.

Cisco IOS Password Encryption Levels

Some of the passwords that you configure on your networking device are saved in the configuration in plain text. This means that if you store a copy of the configuration file on a disk, anybody with access to the disk can discover the passwords by reading the configuration file. The following password types are stored as plain text in the configuration by default:

- Console passwords for local CLI sessions
- Virtual terminal line passwords for remote CLI sessions
- Username passwords using the default method for configuring the password
- Privileged EXEC mode password when it is configured with the **enable password** *password* command
- Authentication key chain passwords used by RIPv2 and EIGRP
- BGP passwords for authenticating BGP neighbors
- OSPF authentication keys for authenticating OSPF neighbors
- ISIS passwords for authenticating ISIS neighbors

This excerpt from a router configuration file shows examples of passwords and authentication keys that are stored as clear text.

```
!
enable password O9Jb6D
!
username gjones password 0 kV9sIj3
!
key chain trees
 key 1
  key-string willow
!
interface Ethernet1/0.1
 ip address 172.16.6.1 255.255.255.0
 ip router isis
 ip rip authentication key-chain trees
 ip authentication key-chain eigrp 1 trees
 ip ospf authentication-key j7876
 no snmp trap link-status
 isis password u7865k
!
line vty 0 4
 password V9jA5M
!
```

You can encrypt these clear text passwords in the configuration file by using the **service password-encryption** command. This should be considered only a minimal level of security because the encryption algorithm used by the **service password-encryption** command to encrypt passwords creates text strings that be decrypted using tools that are publicly available. You should still protect access to any electronic or paper copies of your configuration files after you use the **service password-encryption** command.

The **service password-encryption** command does not encrypt the passwords when they are sent to the remote device. Anybody with a network traffic analyzer who has access to you network can capture these passwords from the packets as they are transmitted between the devices. See the "Configuring Password Encryption for Clear Text Passwords" section on page 1839 for more information on encrypting clear text passwords in configuration files.

Many of the Cisco IOS features that use clear text passwords can also be configured to use the more secure MD5 algorithm. The MD5 algorithm creates a text string in the configuration file that is much more difficult to decrypt. The MD5 algorithm does not send the password to the remote device. This prevents people using a traffic analyzer to capture traffic on your network from being able to discover your passwords.

You can determine the type of password encryption that has been used by the number that is stored with the password string in the configuration file of the networking device. The number 5 in the configuration excerpt below indicates that the enable secret password has been encrypted using the MD5 algorithm.

```
!
enable secret 5 $1$fGCS$rkYbR6.Z8xo4qCl3vghWQ0
!
```

The number 7 in the excerpt below indicates that the enable password has been encrypted using the less secure algorithm used by the **service password-encryption** command.

```
!
enable password 7 00081204
!
```

Cisco IOS CLI Session Usernames

After you have protected access to user EXEC mode and privileged EXEC mode by configuring passwords for them you can further increase the level of security on your networking device by configuring usernames to limit access to CLI sessions to your networking device to specific users.

Usernames that are intended to be used for managing a networking device can be modified with additional options such as:

- Automatically starting a CLI session at a specific privilege level. See "Configuring and Verifying the Networking Device to Require a Username for the First-Line Technical Support Staff" section on page 1847.
- Running a CLI command automatically. See "Configuring and Verifying a Networking Device to Allow Non Administrative Users to View the Running Configuration Automatically: Example" section on page 1855.

See the *Cisco IOS Security Command Reference*, Release,12.4 (http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124cr/hsec_r/index.htm) for more information on how to configure the **username** command.

Cisco IOS Privilege Levels

The default configuration for Cisco IOS based networking devices uses privilege level 1 for user EXEC mode and privilege level 15 for privileged EXEC. The commands that can be run in user EXEC mode at privilege level 1 are a subset of the commands that can be run in privileged EXEC mode at privilege 15.

The **privilege** command is used to move commands from one privilege level to another. For example, some ISPs allow their first level technical support staff to enable and disable interfaces to activate new customer connections or to restart a connection that has stopped transmitting traffic. See the "Configuring and Verifying a Networking Device to Allow Non Administrative Users to Shutdown and Enable Interfaces: Example" section on page 1855 for an example of how to configure this option.

The **privilege** command can also be used to assign a privilege level to a username so that when a user logs in with the username, the session will run at the privilege level specified by the **privilege** command. For example if you want your technical support staff to view the configuration on a networking device to help them troubleshoot network problems without being able to modify the configuration, you can create a username, configure it with privilege level 15, and configure it to run the **show running-config** command automatically. When a user logs in with the username the running configuration will be displayed automatically. The user's session will be logged out automatically after the user has viewed the last line of the configuration. See the "Configuring and Verifying a Networking Device to Allow Non Administrative Users to View the Running Configuration Automatically: Example" section on page 1855 for an example of how to configure this option.

Cisco IOS Password Configuration

Cisco IOS software does not prompt you to repeat any passwords that you configure to verify that you have entered the passwords exactly as you intended. New passwords, and changes to existing passwords, go into effect immediately after you press the Enter key at the end of a password configuration command string. If you make a mistake when you enter a new password and have saved the configuration on the networking device to its startup configuration file and exited privileged EXEC mode before you realize that you made a mistake, you may find that you are no longer able to manage the device.

The following are common situations that can happen:

- You make a mistake configuring a password for local CLI sessions on the console port.
  - If you have properly configured access to your networking device for remote CLI sessions, you can Telnet to it and reconfigure the password on the console port.
- You make a mistake configuring a password for remote Telnet or SSH sessions.
  - If you have properly configured access to your networking device for local CLI sessions, you can connect a terminal to it and reconfigure the password for the remote CLI sessions.
- You make a mistake configuring a password for privileged EXEC mode (enable password or enable secret password).
  - You will have to perform a lost password recovery procedure.
- You make a mistake configuring your username password, and the networking device requires that you log into it with your username.
  - If you do not have access to another account name, you will have to perform a lost password recovery procedure.

To protect yourself from having to perform a lost password recovery procedure open two CLI sessions to the networking device and keep one of them in privilege EXEC mode while you reset the passwords using the other session. You can use the same device (PC or terminal) to run the two CLI sessions or two different devices. You can use a local CLI session and a remote CLI session or two remote CLI sessions for this procedure. The CLI session that you use to configure the password can also be used to verify that the password was changed properly. The other CLI session that you keep in privileged EXEC mode can be used to change the password again if you made a mistake the first time you configured it.

You should not save password changes that you have made in the running configuration to the startup configuration until you have verified that your password was changed successfully. If you discover that you made a mistake configuring a password, and you were not able to correct the problem using the second CLI session technique described above, you can power cycle the networking device so that it returns to the previous passwords that are stored in the startup configuration.

How To Configure Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices

This section contains the following procedures:

- Protecting Access to User Exec Mode, page 1833
- Protecting Access to Privileged Exec Mode, page 1837
- Configuring Security Options with Passwords, Privilege Levels and usernames to Manage Access to CLI Sessions and CLI Commands, page 1842
- Recovering from a Lost or Misconfigured Password for Local CLI Sessions, page 1850

- Recovering from a Lost or Misconfigured Password for Remote CLI Sessions, page 1851
- Recovering from a Lost or Misconfigured Passwords for Privileged EXEC Mode, page 1852

Protecting Access to User Exec Mode

This section contains the following procedures:

- Configuring and Verifying a Password for Remote CLI Sessions, page 1833
- Configuring and Verifying a Password for Local CLI Sessions, page 1835

Configuring and Verifying a Password for Remote CLI Sessions

This task will assign a password for remote CLI sessions. After you have completed this task the networking device will prompt you for a password the next time that you start a remote CLI session with it.

Cisco IOS based networking devices require that you have a password configured for remote CLI sessions. If you attempt to start a remote CLI session with a device that doesn't have a password configured for remote CLI sessions you will see a message that a password is required and has not been set. The remote CLI session will be terminated by the remote host.

Prerequisites

If you have not previously configured a password for remote CLI sessions, you must perform this task over a local CLI session using a terminal or a PC running a terminal emulation application, attached to the console port.

Your terminal, or terminal emulation application, must be configured with the settings that are used by the console port on the networking device. The console ports on most Cisco networking devices require the following settings: 9600 baud, 8 data bits, 1 stop bit, no parity, and flow control is set to "none." See the documentation for your networking device if these settings do not work for your terminal.

To perform the verification step (Step 6) for this task, your networking device must have an interface that is in an operational state. The interface must have a valid IP address.

Restrictions

If you have not previously configured a password for remote CLI sessions, you must perform this task over a local CLI session using a terminal attached to the console port.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line vty** *line-number* [*ending-line-number*]
4. **password** *password*
5. **end**
6. **telnet ip-address**
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**line vty** *line-number* [*ending-line-number*] **Example:** `Router(config)# line vty 0 4`	Enters line configuration mode.
Step 4	**password** *password* **Example:** `Router(config-line)# password H7x3U8`	The argument *password* is a character string that specifies the line password. The following rules apply to the *password* argument: • The first character cannot be a number. • The string can contain any alphanumeric characters, including spaces, up to 80 characters. You cannot specify the password in the format number-space-anything. • Passwords are case sensitive.
Step 5	**end** **Example:** `Router(config-line)# end`	Exits the current configuration mode and returns to privileged EXEC mode.
Step 6	**telnet** *ip-address* **Example:** `Router# telnet 172.16.1.1`	Start a remote CLI session with the networking device from your current CLI session using the IP address of an interface in the networking device that is in an operational state (interface up, line protocol up). • Enter the password that you configured is step 4 when prompted. **Note** This procedure is often referred to as a starting a recursive Telnet session because you are initiating a remote Telnet session with the networking device from the networking device itself.
Step 7	**exit**	Terminates the remote CLI session (recursive Telnet session) with the networking device.

Troubleshooting Tips

Repeat this task if you made a mistake configuring the remote CLI session password.

What to Do Next

Proceed to the "Configuring and Verifying a Password for Local CLI Sessions" section on page 1835 .

Configuring and Verifying a Password for Local CLI Sessions

This task will assign a password for local CLI sessions over the console port. After you have completed this task, the networking device will prompt you for a password the next time that you start a local CLI session on the console port.

This task can be performed over a local CLI session using the console port or a remote CLI session. If you want to perform the optional step of verifying that you configured the password correctly you should perform this task using a local CLI session using the console port.

Prerequisites

If you want to perform the optional step of verifying the local CLI session password, you must perform this task using a local CLI session. You must have a terminal or a PC running a terminal emulation program, connected to the console port of the networking device. Your terminal must be configured with the settings that are used by the console port on the networking device. The console ports on most Cisco networking devices require the following settings: 9600 baud, 8 data bits, 1 stop bit, no parity, and flow control is set to "none." See the documentation for your networking device if these settings do not work for your terminal.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line console 0**
4. **password** *password*
5. **end**
6. **exit**
7. **Press the Enter key, and enter the password from Step 4 when prompted.**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**line console 0** **Example:** `Router(config)# line console 0`	Enters line configuration mode and selects the console port as the line that you are configuring.
Step 4	**password** *password* **Example:** `Router(config-line)# password Ji8F5Z`	The argument *password* is a character string that specifies the line password. The following rules apply to the *password* argument: • The first character cannot be a number. • The string can contain any alphanumeric characters, including spaces, up to 80 characters. You cannot specify the password in the format number-space-anything. • Passwords are case sensitive.
Step 5	**end** **Example:** `Router(config-line)# end`	Exits the current configuration mode and returns to privileged EXEC mode.
Step 6	**exit** **Example:** `Router# exit`	Exits privileged EXEC mode.
Step 7	Press the Enter key.	(Optional) Initiates the local CLI session on the console port. • Enter the password that you configured is step 4 when prompted to verify that it was configured correctly. **Note** This step can be performed only if you are using a local CLI session to perform this task.

Troubleshooting Tips

If your new password is not accepted proceed to the "Recovering from a Lost or Misconfigured Password for Local CLI Sessions" section on page 1850 for instructions on what to do next.

What to Do Next

Proceed to the "Protecting Access to Privileged Exec Mode" section on page 1837.

Protecting Access to Privileged Exec Mode

This section contains the following procedures:

- Configuring and Verifying the Enable Password, page 1837 (optional)
- Configuring Password Encryption for Clear Text Passwords, page 1839 (optional)
- Configuring and Verifying the Enable Secret Password, page 1840 (recommended)

Configuring and Verifying the Enable Password

Cisco no longer recommends that you use the **enable password** command to configure a password for privileged EXEC mode. The password that you enter with the **enable password** command is stored as plain text in the configuration file of the networking device. You can encrypt the password for the **enable password** command in the configuration file of the networking device using the **service password-encryption** command. However the encryption level used by the **service password-encryption** command can be decrypted using tools available on the Internet.

Instead of using the **enable password** command, Cisco recommends using the **enable secret** command because it encrypts the password that you configure with it with strong encryption . For more information on password encryption issues see the "Cisco IOS Password Encryption Levels" section on page 1829. For information on configuring the **enable secret** command see the "Configuring and Verifying the Enable Secret Password" section on page 1840.

Restrictions

The networking device must not have a password configured by the **enable secret** command in order to perform this task successfully. If you have already configured a password for privileged EXEC mode using the **enable secret** command, the password configured takes precedences over the password that you configure in this task using the **enable password** command.

You cannot use the same password for the **enable secret** command and the **enable** *password* command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **enable password** *password*
4. **end**
5. **exit**
6. **enable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**enable password** *password* **Example:** `Router(config)# enable password t6D77CdKq`	The argument *password* is a character string that specifies the enable password. The following rules apply to the *password* argument: • Must contain from 1 to 25 uppercase and lowercase alphanumeric characters. • Must not have a number as the first character. • Can have leading spaces, but they are ignored. However, intermediate and trailing spaces are recognized. • Can contain the question mark (?) character if you precede the question mark with the key combination Crtl-v when you create the password; for example, to create the password abc?123, do the following: – Enter abc – Type Crtl-v – Enter ?123
Step 4	**end** **Example:** `Router(config)# end`	Exits the current configuration mode and returns to privileged EXEC mode.
Step 5	**exit** **Example:** `Router# exit`	Exits privileged EXEC mode.
Step 6	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter the password you configured in step 3.

Troubleshooting Tips

If your new password is not accepted, proceed to the "Recovering from a Lost or Misconfigured Passwords for Privileged EXEC Mode" section on page 1852 for instructions on what to do next.

What to Do Next

Encrypt the clear text enable password in the configuration file of the networking device using the procedure described in "Configuring Password Encryption for Clear Text Passwords" section on page 1839.

Configuring Password Encryption for Clear Text Passwords

Cisco IOS stores passwords in clear text in network device configuration files for several features such as passwords for local and remote CLI sessions, and passwords for neighbor authentication for routing protocols. Clear text passwords are a security risk because anybody with access to archived copies of the configuration files can discover the passwords that are stored as clear text. The **service password-encryption** command can be used to encrypt clear text commands in the configuration files of networking devices. See the "Cisco IOS Password Encryption Levels" section on page 1829 for more information.

Perform the following steps to configure password encryption for passwords that are stored as clear text in the configuration files of your networking device.

Prerequisites

You must have at least one feature that uses clear text passwords configured on your networking device for this command to have any immediate effect.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **service password-encryption**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**service password-encryption** **Example:** `Router(config)# service password-encryption`	Enables Password encryption for all passwords clear text passwords, including username passwords, authentication key passwords, the privileged command password, console and virtual terminal line access passwords, and Border Gateway Protocol neighbor passwords.
Step 4	**end** **Example:** `Router(config)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Configuring and Verifying the Enable Secret Password

Cisco recommends that you use the **enable secret** command, instead of the **enable** *password* command to configure a password for privileged EXEC mode. The password created by the **enable secret** command is encrypted with the more secure MD5 algorithm.

Restrictions

You cannot use the same password for the **enable secret** command and the **enable** *password* command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **enable secret** *password*
   or
   enable secret *5 previously-encrypted-password*
4. **end**
5. **exit**
6. **enable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**enable secret** *password* or **enable secret** *5 previously-encrypted-password* **Example:** `Router(config)# enable secret t6D77CdKq` or **Example:** `Router(config)# enable secret 5 $1$/x6H$RhnDI3yLC4GA01aJnHLQ4/`	The argument *password* is a character string that specifies the **enable secret** password. The following rules apply to the *password* argument: • Must contain from 1 to 25 uppercase and lowercase alphanumeric characters. • Must not have a number as the first character. • Can have leading spaces, but they are ignored. However, intermediate and trailing spaces are recognized. • Can contain the question mark (?) character if you precede the question mark with the key combination Crtl-v when you create the password; for example, to create the password abc?123, do the following: – Enter abc – Type Crtl-v – Enter ?123 or Sets a previously encrypted password for privileged EXEC mode by entering the number 5 before the previously encrypted string. You must enter an exact copy of a password from a configuration file that was previously encrypted by the **enable secret** command to use this method.
Step 4	**end** **Example:** `Router(config)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router# exit`	Exits privileged EXEC mode.
Step 6	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter the password that you configured in Step 3.

Troubleshooting Tips

If your new password is not accepted proceed to the "Recovering from a Lost or Misconfigured Passwords for Privileged EXEC Mode" section on page 1852 for instructions on what to do next.

What to Do Next

If you have finished configuring passwords for local and remote CLI sessions and you want to configure additional security features, such as usernames, and privilege levels proceed to the "Configuring Security Options with Passwords, Privilege Levels and usernames to Manage Access to CLI Sessions and CLI Commands" section on page 1842.

Configuring Security Options with Passwords, Privilege Levels and usernames to Manage Access to CLI Sessions and CLI Commands

The tasks in this section describe how to configure your networking device to permit the use of a subset of privileged EXEC mode commands by users who should not have access to all of of the commands available in privileged EXEC mode.

These tasks are beneficial for companies that have multiple levels of network support staff and the company wants the staff at each level to have access to a different subset of the privileged EXEC mode commands.

In this task the users who should not have access to all of of the commands available in privileged EXEC mode are referred to as the first-line technical support staff.

This section contains the following procedures:

- Configuring the Networking Device for the First-Line Technical Support Staff, page 1842
- Verifying the Configuration for the First-Line Technical Support Staff, page 1845
- Configuring and Verifying the Networking Device to Require a Username for the First-Line Technical Support Staff, page 1847

Configuring the Networking Device for the First-Line Technical Support Staff

This task describes how to configure the networking device for first-line technical support users. First-line technical support staff are usually not allowed to run all of the commands available in privileged EXEC mode (privilege level 15) on a networking device. They are prevented from running commands that they are not authorized for by not being granted access to the password assigned to privileged EXEC mode or to other roles that have been configured on the networking device.

The **privilege** command is used to move commands from one privilege level to another in order to create the additional levels of administration of a networking device that is required by companies that have different levels of network support staff with different skill levels.

The default configuration of a Cisco IOS device permits two types of users to access the CLI. The first type of user is a person who is only allowed to access user EXEC mode. The second type of user is a person who is allowed access to privileged EXEC mode. A user who is only allowed to access user EXEC mode is not allowed to view or change the configuration of the networking device, or to make any changes to the operational status of the networking device. On the other hand, a user who is allowed access to privileged EXEC mode can make any change to a networking device that is allowed by the CLI.

In this task the two commands that normally run at privilege level 15 are reset to privilege level 7 using the privilege command in order that first-line technical support users will be allowed to run the two commands. The two commands for which the privilege levels will be reset are the **clear counters** command and **reload** command.

- The **clear counters** command is used to reset the counter fields on interfaces for statistics such as packets received, packets transmitted, and errors. When a first-line technical support user is troubleshooting an interface related connectivity issue between networking devices, or with remote users connecting to the network, it is useful to reset the interface statistics to zero and them monitor the interfaces for a period of time to see if the values in the interface statistics counters change.
- The **reload** command is used initiate a reboot sequence for the networking device. One common use of the reload command by first-line technical support staff is to cause the networking device to reboot during a maintenance window so that it loads a new operating system that was previously copied onto the networking device's file system by a user with a higher level of authority.

Any user that is permitted to know the **enable secret** password that is assigned to the first-line technical support user role privilege level can access the networking device as a first-line technical support user. You can add an additional level of security by configuring a username on the networking device and requiring that the users know the username and the password. Configuring a username as an additional level of security is described in the "Configuring and Verifying the Networking Device to Require a Username for the First-Line Technical Support Staff" section on page 1847.

Privilege Command Enhancement

Before Cisco IOS Releases 12.0(22)S and 12.2(13)T, each command in a privilege level had to be specified with a separate **privilege** command. In Cisco IOS Releases 12.0(22)S, 12.2(13)T, and later releases, a "wildcard" option specified by the new keyword **all** was introduced that allows you to configure access to multiple commands with only one **privilege** command. By using the new **all** keyword, you can specify a privilege level for all commands which begin with the string you enter. In other words, the **all** keyword allows you to grant access to all command-line options and suboptions for a specified command.

For example, if you wanted to create a privilege level to allow users to configure all commands which begin with **service-module t1** (such as **service-module t1 linecode** or **service-module t1 clock source**) you can use the **privilege interface all level** *2* ***service-module t1*** command instead of having to specify each **service-module t1** command separately.

If the command specified in the privilege command (used with the **all** keyword) enables a configuration submode, all commands in the submode of that command will also be set to the specified privilege level.

Restrictions

The **all** "wildcard" keyword option for the **privilege** command is not supported in versions of Cisco IOS software prior to Cisco IOS Releases 12.0(22)S and, 12.2(13)T.

You must not have the **aaa new-model** command enabled on the networking device. You must not have the **login local** command configured for the local CLI sessions over the console port or the remote CLI sessions.

Note For clarity, only the arguments and keywords that are relevant for each step are shown in the syntax for the steps in this task. See the Cisco IOS command reference book for your Cisco IOS release for further information on the additional arguments and keywords that can be used with these commands.

Caution Do not use the no form of the **privilege** command to reset the privilege level of a command to its default because it might not return the configuration to the correct default state. Use the **reset** keyword for the **privilege** command instead to return a command to its default privilege level. For example, to remove the **privilege exec level** *reload* command from the configuration and return the **reload** command to its default privilege of 15, use the **privilege exec reset** *reload* command.

SUMMARY STEPS

1. **enable** *password*
2. **configure terminal**
3. **enable secret level** *level password*
4. **privilege exec level** *level command-string*
5. **privilege exec all level** *level command-string*
6. **end**

DETAILED STEPS

Step 1 **enable** *password*

Enters privileged EXEC mode. Enter the password when prompted.

```
Router> enable
```

Step 2 **configure terminal**

Enters global configuration mode.

```
Router# configure terminal
```

Step 3 **enable secret level** *level password*

Configures a new enable secret password for privilege level 7.

```
Router(config)# enable secret level 7 Zy72sKj
```

Step 4 **privilege exec level** *level command-string*

Changes the privilege level of the **clear counters** command from privilege level 15 to privilege level 7.

```
Router(config)# privilege exec level 7 clear counters
```

Step 5 **privilege exec all level** *level command-string*

Changes the privilege level of the **reload** command from privilege level 15 to privilege level 7.

```
Router(config)# privilege exec all level 7 reload
```

Step 6 **end**

Exits global configuration mode.

```
Router(config)# end
```

Verifying the Configuration for the First-Line Technical Support Staff

This task describes how to verify that the network device is configured correctly for the first-line technical support staff.

Prerequisites

The following commands must have been modified to run at privilege level 7 for this task:

- **clear counters**
- **reload**

SUMMARY STEPS

1. **enable** *level password*
2. **show privilege**
3. **clear counters**
4. **clear** *ip route* *
5. **reload in time**
6. **reload cancel**
7. **disable**
8. **show privilege**

DETAILED STEPS

Step 1 **enable** *level password*

Logs the user into the networking device at the privilege level specified for the level argument.

```
Router> enable 7 Zy72sKj
```

Step 2 **show privilege**

Displays the privilege level of the current CLI session

```
Router# show privilege
Current privilege level is 7
```

Step 3 **clear counters**

The clear counters command clears the interface counters. This command has been changed from privilege level 15 to privilege level 7.

```
Router# clear counters
Clear "show interface" counters on all interfaces [confirm]
Router#
02:41:37: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console
```

Step 4 **clear ip route** *

The *ip route* argument string for the **clear** command should not be allowed because it was not changed from privilege level 15 to privilege level 7.

```
Router# clear ip route *
         ^
% Invalid input detected at '^' marker.

Router#
```

Step 5 **reload in time**

The reload command causes the networking device to reboot.

```
Router# reload in 10
Reload scheduled in 10 minutes by console
Proceed with reload? [confirm]
Router#


***
*** --- SHUTDOWN in 0:10:00 ---
***

02:59:50: %SYS-5-SCHEDULED_RELOAD: Reload requested for 23:08:30 PST Sun Mar 20
```

Step 6 **reload cancel**

The **reload cancel** terminates a reload that was previously setup with the the **reload in** *time* command.

```
Router# reload cancel

***
*** --- SHUTDOWN ABORTED ---
***

04:34:08: %SYS-5-SCHEDULED_RELOAD_CANCELLED:  Scheduled reload cancelled at 15:38:46 PST
Sun Mar 27 2005
```

Step 7 **disable**

Exits the current privilege level and returns to privilege level 1.

```
Router# disable
```

Step 8 **show privilege**

Displays the privilege level of the current CLI session

```
Router> show privilege
Current privilege level is 1
```

Troubleshooting Tips

If your configuration does not work the way that you want it to and you want to remove the privilege commands from the configuration, use the **reset** keyword for the **privilege** command to return the commands to their default privilege level. For example, to remove the command **privilege exec level** *reload* command from the configuration and return the **reload** command to its default privilege of 15 use the **privilege exec reset** *reload* command.

What to Do Next

If you want to add an additional level of security by requiring that the first level technical staff use a login name, proceed to the "Configuring and Verifying the Networking Device to Require a Username for the First-Line Technical Support Staff" section on page 1847.

Configuring and Verifying the Networking Device to Require a Username for the First-Line Technical Support Staff

This task configures the networking device to require that the first-line technical support staff login to the networking device with a login name of admin. The admin username configured in this task is assigned the privilege level 0f 7 which will allow users who log in with this name to run the commands that were reassigned to privilege level 7 in the previous task. When a user successfully logs in with the admin username, the CLI session will automatically enter privilege level 7.

Enhanced Username Password Security

Before Cisco IOS Releases 12.0(18)S and 12.2(8)T, two types of passwords were associated with usernames: Type 0, which is a clear text password visible to any user who has access to privileged mode on the router, and type 7, which has a password encrypted by the **service password encryption** command.

In Cisco IOS Releases 12.0(18)S, 12.2(8)T, and later releases, the new **secret** keyword for the **username** command allows you to configure Message Digest 5 (MD5) encryption for username passwords.

Prerequisites

The following commands must have been modified to run at privilege level 7 for this task:

- **clear counters**
- **reload**

See the "Configuring the Networking Device for the First-Line Technical Support Staff" section on page 1842 for instructions on how to change the privilege level for a command.

Restrictions

MD5 encryption for the **username** command is not supported in versions of Cisco IOS software prior to Cisco IOS Releases 12.0(18)S and 12.2(8)T.

You must not have the aaa-new model command enabled on the networking device. You must not have the **login local** command configured for the local CLI sessions over the console port or the remote CLI sessions.

Note For clarity, only the arguments and keywords that are relevant for each step are shown in the syntax for the steps in this task. Refer to the Cisco IOS command reference book for your Cisco IOS release for further information on the additional arguments and keywords that can be used with these commands.

SUMMARY STEPS

1. **enable** *password*
2. **configure terminal**

3. **username *username* privilege** *level* **secret** *password*
4. **end**
5. **disable**
6. **login username** *password*
7. **show privilege**
8. **clear counters**
9. **clear ip route** *
10. **reload in 10**
11. **reload cancel**
12. **disable**
13. **show privilege**

DETAILED STEPS

Step 1 **enable** *t6D77CdKq*

Enters privileged EXEC mode. Enter the password when prompted.

```
Router> enable
```

Step 2 **configure terminal**

Enters global configuration mode.

```
Router# configure terminal
```

Step 3 **username *username* privilege** *level* **secret** *password*

Creates a username and applies MD5 encryption to the *password* text string.

```
Router(config)# username admin privilege 7 secret Kd65xZa
```

Step 4 **end**

Exits global configuration mode.

```
Router(config)# end
```

Step 5 **disable**

Exits the current privilege level and returns to user EXEC mode.

```
Router# disable
```

Step 6 **login** *username*

Logs in the user. Enter the username and password you configured in step 3 when prompted.

```
Router> login admin
```

Step 7 **show privilege**

The **show privilege** command displays the privilege level of the CLI session.

```
Router# show privilege

Current privilege level is 7
```

Step 8 **clear counters**

The **clear counters** command clears the interface counters. This command has been changed from privilege level 15 to privilege level 7.

```
Router# clear counters
Clear "show interface" counters on all interfaces [confirm]
Router#
02:41:37: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console
```

Step 9 **clear** *ip route* *

The *ip route* argument string for the **clear** command is not allowed because it was not changed from privilege level 15 to privilege level 7.

```
Router# clear ip route *
         ^
% Invalid input detected at '^' marker.

Router#
```

Step 10 **reload in** ***time***

The reload command causes the networking device to reboot.

```
Router# reload in 10
Reload scheduled in 10 minutes by console
Proceed with reload? [confirm]
Router#


***
*** --- SHUTDOWN in 0:10:00 ---
***

02:59:50: %SYS-5-SCHEDULED_RELOAD: Reload requested for 23:08:30 PST Sun Mar 20
```

Step 11 **reload cancel**

The **reload cancel** command terminates a reload that was previously setup with the the **reload in** *time* command.

```
Router# reload cancel

***
*** --- SHUTDOWN ABORTED ---
***

04:34:08: %SYS-5-SCHEDULED_RELOAD_CANCELLED:  Scheduled reload cancelled at 15:38:46 PST
Sun Mar 27 2005
```

Step 12 **disable**

Exits the current privilege level and returns to user EXEC mode.

```
Router# disable
```

Step 13 **show privilege**

Displays the privilege level of the current CLI session

```
Router> show privilege
Current privilege level is 1
```

Recovering from a Lost or Misconfigured Password for Local CLI Sessions

There are three methods that can be used to recover from a lost or misconfigured password for local CLI sessions over console port. The method that you will use depends on the current configuration of your networking device.

- Networking Device Is Configured to Allow Remote CLI Sessions, page 1850
- Networking Device Is Not Configured to Allow Remote CLI Sessions and the Local CLI Session Password Has Not Been Saved to the Startup Configuration File, page 1850
- Networking Device is not Configured to Allow Remote CLI Sessions and the Local CLI Session Password Has Been Saved to the Startup Configuration File, page 1850

Networking Device Is Configured to Allow Remote CLI Sessions

The fastest method to recover from a lost, or misconfigured password for local CLI sessions is to establish a remote CLI session with the networking device and repeat the "Configuring and Verifying a Password for Local CLI Sessions" section on page 1835. Your networking device must be configured to allow remote CLI sessions and you must know the remote CLI session password to perform this procedure.

Networking Device Is Not Configured to Allow Remote CLI Sessions and the Local CLI Session Password Has Not Been Saved to the Startup Configuration File

If you cannot establish a remote session to your networking device, and you have not saved the misconfigured local CLI session password to the startup configuration, you can restart the networking device. When the networking device starts up again it will read the startup configuration file. The previous local CLI session password is restored.

Caution Restarting a networking device will cause it to stop forwarding traffic. This will also cause an interruption in any services that are running on the networking device, such as a DHCP server service, to stop. You should only restart a networking device during a period of time that has been allocated for network maintenance.

Networking Device is not Configured to Allow Remote CLI Sessions and the Local CLI Session Password Has Been Saved to the Startup Configuration File

If you can not establish a remote CLI session with the networking device, and you have saved the misconfigured local CLI session password to the startup configuration, or you have lost the local CLI session password, you must perform a password recovery procedure. Password recovery procedures are device specific. You must locate the document that describes the procedure for your type of networking device.

There are three methods for locating a password recovery procedure document for your networking device:

- Many networking devices have a Password Recovery subsection in the Troubleshoot and Alerts section of their product support page on Cisco's Technical Support website (http://www.cisco.com/tac). Navigate to the Troubleshoot and Alerts section of the product support page for your networking device and look for the Password Recovery subsection.

- If you do not find the password recovery document for your networking device on its product support page try searching for the text string **"password recovery"** on Cisco's Technical Support website (http://www.cisco.com/tac). Enclose the text string in double quotes. You can improve the search results by adding an additional text string that matches the model number, series number or platform name for your networking device, For example searching on the string **"password recovery" 12000** will provide search results that give documents with the words **password**, **recovery**" and **12000** in the title a higher ranking.
- If the product support page for your networking device does not have a password recovery document, and you can not find the correct document by searching for it, you can try the Cisco's Network Professionals Connection (http://www.cisco.com/go/netpro).

Recovering from a Lost or Misconfigured Password for Remote CLI Sessions

There are three methods that can be used to recover from a lost, or misconfigured remote CLI session password. The method that you will use depends on the current configuration of your networking device.

- Networking Device Is Configured to Allow Local CLI Sessions, page 1851
- Networking Device Is Not Configured to Allow Local CLI Sessions and the Remote CLI Session Password Has Not Been Saved to the Startup Configuration File, page 1851
- Networking Device Is Not Configured to Allow Local CLI Sessions and the Remote CLI Session Password Has Been Saved to the Startup Configuration File, page 1852

Networking Device Is Configured to Allow Local CLI Sessions

The fastest method to recover from a lost, or misconfigured password for remote CLI sessions is to establish a local CLI session with the networking device and repeat the "Configuring and Verifying a Password for Remote CLI Sessions" section on page 1833. Your networking device must be configured to allow local CLI sessions and you must know the local CLI session password to perform this procedure.

Networking Device Is Not Configured to Allow Local CLI Sessions and the Remote CLI Session Password Has Not Been Saved to the Startup Configuration File

If you cannot establish a local CLI session to your networking device, and you have not saved the misconfigured remote CLI session password to the startup configuration, you can restart the networking device. When the networking device starts up again it will read the startup configuration file. The previous remote CLI session password is restored.

Caution Restarting a networking device will cause it to stop forwarding traffic. This will also cause an interruption in any services that are running on the networking device, such as a DHCP server service, to stop. You should only restart a networking device during a period of time that has been allocated for network maintenance.

Networking Device Is Not Configured to Allow Local CLI Sessions and the Remote CLI Session Password Has Been Saved to the Startup Configuration File

If you can not establish a local CLI session with the networking device, and you have saved the misconfigured remote CLI session password to the startup configuration, or you have lost the remote CLI session password, you must perform a password recovery procedure. Password recovery procedures are device specific. You must locate the document that describes the procedure for your type of networking device.

There are three methods for locating a password recovery procedure document for your networking device:

- Many networking devices have a Password Recovery subsection in the Troubleshoot and Alerts section of their product support page on Cisco's Technical Support website (http://www.cisco.com/tac). Navigate to the Troubleshoot and Alerts section of the product support page for your networking device and look for the Password Recovery subsection.
- If you do not find the password recovery document for your networking device on its product support page try searching for the text string **"password recovery"** on Cisco's Technical Support website (http://www.cisco.com/tac). Enclose the text string in double quotes. You can improve the search results by adding an additional text string that matches the model number, series number or platform name for your networking device, For example searching on the string **"password recovery" 12000** will provide search results that give documents with the words **password**, **recovery**" and **12000** in the title a higher ranking.
- If the product support page for your networking device does not have a password recovery document, and you can not find the correct document by searching for it, you can try the Cisco's Network Professionals Connection (http://www.cisco.com/go/netpro).

Recovering from a Lost or Misconfigured Passwords for Privileged EXEC Mode

There are two methods that can be used to recover from a lost, or misconfigured Privileged EXEC Mode password. The method that you will use depends on the current configuration of your networking device.

- A Misconfigured Privileged EXEC Mode Password Has Not Been Saved to the Startup Configuration File, page 1852
- A Misconfigured Privileged EXEC Mode Password Has Been Saved to the Startup Configuration File, or the Privileged EXEC Mode Password Has Been Lost, page 1853

A Misconfigured Privileged EXEC Mode Password Has Not Been Saved to the Startup Configuration File

If you have not saved the misconfigured privileged EXEC mode password to the startup configuration, you can restart the networking device. When the networking device starts up again it will read the startup configuration file. The previous privileged EXEC mode password is restored.

Caution Restarting a networking device will cause it to stop forwarding traffic. This will also cause an interruption in any services that are running on the networking device, such as a DHCP server service, to stop. You should only restart a networking device during a period of time that has been allocated for network maintenance.

A Misconfigured Privileged EXEC Mode Password Has Been Saved to the Startup Configuration File, or the Privileged EXEC Mode Password Has Been Lost

If you have saved the misconfigured privileged EXEC mode password to the startup configuration, or you have lost the privileged EXEC mode password, you must perform a password recovery procedure. Password recovery procedures are device specific. You must locate the document that describes the procedure for your type of networking device.

There are three methods for locating a password recovery procedure document for your networking device:

- Many networking devices have a Password Recovery subsection in the Troubleshoot and Alerts section of their product support page on Cisco's Technical Support website (http://www.cisco.com/tac). Navigate to the Troubleshoot and Alerts section of the product support page for your networking device and look for the Password Recovery subsection.
- If you do not find the password recovery document for your networking device on its product support page try searching for the text string **"password recovery"** on Cisco's Technical Support website (http://www.cisco.com/tac). Enclose the text string in double quotes. You can improve the search results by adding an additional text string that matches the model number, series number or platform name for your networking device, For example searching on the string **"password recovery" 12000** will provide search results that give documents with the words **password**, **recovery**" and **12000** in the title a higher ranking.
- If the product support page for your networking device does not have a password recovery document, and you can not find the correct document by searching for it, you can try the Cisco's Network Professionals Connection (http://www.cisco.com/go/netpro).

Configuration Examples for Configuring Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices

This section contains the following configuration examples:

- Configuring and Verifying a Networking Device to Allow Non Administrative Users to Clear Remote CLI Sessions: Example, page 1853
- Configuring and Verifying a Networking Device to Allow Non Administrative Users to View the Running Configuration Automatically: Example, page 1855
- Configuring and Verifying a Networking Device to Allow Non Administrative Users to Shutdown and Enable Interfaces: Example, page 1855

Configuring and Verifying a Networking Device to Allow Non Administrative Users to Clear Remote CLI Sessions: Example

The following example shows how to configure a networking device to allow a non administrative user to clear remote CLI session virtual terminal (VTY) lines.

The first section is an an excerpt of the running configuration for this example. The following sections show you how this example is used.

The following section is an excerpt of the running-configuration:

```
!
privilege exec level 7 clear line
!
no aaa new-model
!
!
username admin privilege 7 secret 5 $1$tmIw$1aM7sadKhWMpkVTzxNw1J.
!
privilege exec level 7 clear line
!
! the privilege exec level 7 clear command below is entered automatically
! when you enter the privilege exec level 7 clear line command above, do
! not enter it again
!
privilege exec level 7 clear
!
```

The following section using the **login** command shows the user logging in to the networking device with the username of admin:

```
R1> login
Username: admin
Password:
```

The following section using the **show privilege** command shows that the current privilege level is 7:

```
R1# show privilege
Current privilege level is 7
R1#
```

The following section using the **show user** command shows that two users (admin and root) are currently logged in to the networking device:

```
R1# show user
    Line       User       Host(s)              Idle       Location
*  0 con 0     admin      idle                 00:00:00
   2 vty 0     root       idle                 00:00:17 172.16.6.2

  Interface    User               Mode         Idle     Peer Address
```

The following section using the **clear line** *2* command terminates the remote CLI session in use by the username root:

```
R1# clear line  2
[confirm]
 [OK]
```

The following section using the **show user** command shows that admin is the only user currently logged in to the networking device:

```
R1# show user
    Line       User       Host(s)              Idle       Location
*  0 con 0     admin      idle                 00:00:00

  Interface    User               Mode         Idle     Peer Address
```

Configuring and Verifying a Networking Device to Allow Non Administrative Users to View the Running Configuration Automatically: Example

The following example shows how to configure the networking device to allow a non administrative users (no access to privileged EXEC mode) to view the running configuration automatically. This example requires that the username is configured for privilege level 15 because many of the commands in the configuration file can be viewed only by users who have access to privilege level 15.

The solution is to temporarily allow the user access to privilege level 15 while running the **show running-config** command and then terminating the CLI session when the end of the configuration file has been viewed. In this example the networking device will automatically terminate the CLI session when the end of the configuration file has been viewed. No further configuration steps are required.

Caution You must include the **noescape** keyword for the **username** command to prevent the user from entering an escape character that will terminate viewing the configuration file and leave the session running at privilege level 15.

```
!
!
username viewconf privilege 15 noescape secret 5 $1$zA9C$TDWD/Q0zwp/5xRwRqdgc/.
username viewconf autocommand show running-config
!
```

Configuring and Verifying a Networking Device to Allow Non Administrative Users to Shutdown and Enable Interfaces: Example

The following example shows how to configure a networking device to allow non administrative users to shutdown and enable interfaces.

The first section is an an excerpt of the running configuration for this example. The following sections show you how this example is used.

The following section is an excerpt of the running-configuration:

```
!
no aaa new-model
!
username admin privilege 7 secret 5 $1$tmIw$1aM7sadKhWMpkVTzxNw1J.
!
privilege interface all level 7 shutdown
privilege interface all level 7 no shutdown
privilege configure level 7 interface
privilege exec level 7 configure terminal
!
! the privilege exec level 7 configure command below is entered automatically
! when you enter the privilege exec level 7 configure terminal command above, do
! not enter it again
!
privilege exec level 7 configure
!
```

The following section using the **login** command shows the user logging in to the networking device with the username of admin:

```
R1> login
Username: admin
```

```
Password:
```

The following section using the **show privilege** command shows that the current privilege level is 7:

```
R1# show privilege
Current privilege level is 7
R1#
```

The following section using the **show user** command shows that admin is the only user currently logged in to the networking device:

```
R1# show user
    Line       User       Host(s)              Idle       Location
*  0 con 0     admin      idle                 00:00:00

  Interface    User               Mode         Idle     Peer Address
```

The following section shows that the admin user is permitted to shutdown and enable an interface:

```
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# interface ethernet 1/0
R1(config-if)# shutdown
R1(config-if)# no shutdown
R1(config-if)# exit
R1#
```

Where to Go Next

Once you have established a baseline of security for your networking devices you can consider more advanced options such as:

- Role-Based CLI Access—The role-based CLI access feature offers a more comprehensive set of options than the **privilege** command (described in this document) for network managers who want to allow different levels of technical support staff to have different levels of access to CLI commands.
- AAA Security—Many Cisco networking devices offer an advanced level of security using authentication, authorization and accounting (AAA) features. All of the tasks described in this document, and other - more advanced security features - can be implemented using AAA on the networking device in conjunction with a remote TACACS+ or RADIUS server. For information how to configure AAA security features that can be run locally on a networking device, or for information on how to configure remote AAA security using TACACS+ or RADIUS servers, see the *Cisco IOS Security Configuration Guide,* Release 12.4.

Additional References

The following sections provide references related to Configuring Security with Passwords and, Login Usernames for CLI Sessions on Networking Devices.

Related Documents

Related Topic	Document Title
Managing user access to CLI commands and configuration information.	*Role-Based CLI Access*
AAA Security Features	*Cisco IOS Security Configuration Guide,* Release 12.4
Configuring MD5 secure neighbor authentication for protocols such as OSPF and BGP.	*Neighbor Router Authentication: Overview and Guidelines*

Standards

Standard	Title
No new or modified RFCs are supported by this functionality, and support for existing RFCs has not been modified.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this functionality, and support for existing RFCs has not been modified.	—

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Feature Information for Configuring Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices

Table 70 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Releases 12.2(1) or 12.0(3)S or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For details on when support for specific command was introduced, see the command reference documentation.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Note Table 70 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 70 Feature Information for Configuring Security with Passwords, Privilege Levels and, Login Usernames for CLI Sessions on Networking Devices

Feature Name	Releases	Feature Configuration Information
Enhanced Password Security	12.0(18)S 12.2(8)T	Using the Enhanced Password Security feature, you can configure MD5 encryption for username passwords. MD5 encryption is a one-way hash function that makes reversal of an encrypted password impossible, providing strong encryption protection. Using MD5 encryption, you cannot retrieve clear text passwords. MD5 encrypted passwords cannot be used with protocols that require that the clear text password be retrievable, such as Challenge Handshake Authentication Protocol (CHAP). The following sections provide information about this feature: • Configuring and Verifying the Networking Device to Require a Username for the First-Line Technical Support Staff, page 1847
Privilege Command Enhancement	12.0(22)S 12.2(13)T	The keyword **all** was added to the **privilege** command as a wild card to reduce the number of times you ned to enter the **privilege** command when you are changing the privilege level of several keywords for the same command. The following sections provide information about this feature: • Privilege Command Enhancement, page 1843

No Service Password-Recovery

The No Service Password-Recovery feature is a security enhancement that prevents anyone with console access from accessing the router configuration and clearing the password. It also prevents anyone from changing the configuration register values and accessing NVRAM.

Feature History for the No Service Password-Recovery Feature

Release	Modification
12.3(8)YA	This feature was introduced.
12.3(14)T	This feature was integrated into Cisco IOS Release 12.3(14)T.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for No Service Password-Recovery, page 1861
- Information About No Service Password-Recovery, page 1862
- How to Enable No Service Password-Recovery, page 1862
- Configuration Examples for No Service Password-Recovery, page 1870
- Additional References, page 1872
- Command Reference, page 1873

Prerequisites for No Service Password-Recovery

You are required to download and install ROM monitor (ROMMON) version 12.2(11)YV1 before you can use this feature.

Information About No Service Password-Recovery

To configure the No Service Password-Recovery feature, you should understand the following concepts:

- Cisco Password Recovery Procedure, page 1862
- Configuration Registers and System Boot Configuration, page 1862

Cisco Password Recovery Procedure

The Cisco IOS software provides a password recovery procedure that relies upon gaining access to ROMMON mode using the Break key during system startup. In ROMMON mode, the router software can be reloaded at which time prompting a new system configuration that includes a new password.

The current password recovery procedure enables anyone with console access, the ability to access the router and its network. The No Service Password-Recovery feature prevents the completion of the Break key sequence and the entering of ROMMON mode during system startups and reloads.

Configuration Registers and System Boot Configuration

The lowest four bits of the configuration register (bits 3, 2, 1, and 0) form the boot field. The boot field determines if the router boots manually from ROM or automatically from Flash or the network. For example, when the configuration register boot field value is set to any value from 0x2 to 0xF, the router uses the boot field value to form a default boot filename for autobooting from a network server.

Bit 6, when set, ignores the startup configuration, while bit 8 enables a break. To use this feature, the configuration register must be set to autoboot before it can be enabled. Any other configuration register setting will prevent the feature from being enabled.

Note By default, the no confirm prompt and message are not displayed after reloads.

How to Enable No Service Password-Recovery

This section contains the following procedures:

- Upgrading the ROMMON Version, page 1863 (required)
- Verifying the Upgraded ROMMON Version, page 1865 (optional)
- Enabling No Service Password-Recovery, page 1865 (required)
- Recovering a Device, page 1866 (required)

Upgrading the ROMMON Version

If your router or access server does not find a valid system image to load, the system will enter ROMMON mode. ROMMON mode can also be accessed by interrupting the boot sequence during startup.

Another method for entering ROMMON mode is to set the configuration register so that the router automatically enters ROMMON mode when it boots. For information about setting the configuration register value, refer to the *Cisco IOS Configuration Fundamentals and Network Management Configuration Guide*, Release 12.3.

Perform this task to upgrade your version of ROMMON.

SUMMARY STEPS

1. **reload**
2. **set** *tftp-file ip-address ip-subnet-mask default-gateway tftp-server*
3. **sync**
4. **tftpdnld -u**
5. **boot**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**reload** **Example:** `Router> reload`	Reloads a Cisco IOS image. After issuing this command and responding to the system prompts as necessary, the system will begin reloading the system software image. While the system is reloading, press the Break key or a Break key-combination during the first 60 seconds of system startup. Pressing the Break key interrupts the boot sequence and puts the router into ROMMON mode. **Note** The default Break key combination is Ctrl-C, but this may be configured differently on your system.
Step 2	**set** *tftp-file ip-address ip-subnet-mask default-gateway tftp-server* **Example:** `ROMMON> set tftpabc 10.10.0.0 255.0.0.0 10.1.1.0 10.29.32.0`	Displays all the created variables. The arguments are as follows: • *tftp-file*—Location of the new ROMMON image on the TFTP server. The length of the filename is a maximum of 45 characters. • *ip-address*—IP address on the router to connect to the TFTP server. • *ip-subnet-mask*—IP subnet mask of the router. • *default-gateway*—IP address of the gateway of the TFTP server. • *tftp-server*—IP address of the TFTP server from which the image will be downloaded. **Note** This command is not supported on the Cisco 800 series routers.
Step 3	**sync** **Example:** `ROMMON> sync`	Saves the changes to the image.
Step 4	**tftpdnld -u** **Example:** `ROMMON> tftpdnld -u`	Downloads the new ROMMON image from the TFTP server. Reset if prompted.
Step 5	**boot** **Example:** `ROMMON> boot`	Boots the router with the Cisco IOS image in flash memory.

Verifying the Upgraded ROMMON Version

To verify that you have downloaded a new version of ROMMON, use the **show version** command:

```
Router# show version

Cisco IOS Software, C828 Software (C828-K9OS&6-M), Version 12.3 (20040702:094716)
[userid 168]

Copyright (c) 1986-2004 by Cisco Systems, Inc.

ROM: System Bootstrap, Version 12.2(11)YV1, Release Software (fc1)

Router uptime is 22 minutes
System returned to ROM by reload
.
.
.
```

Enabling No Service Password-Recovery

Perform this task to enable the No Service Password-Recovery feature.

Note As a precaution, a valid Cisco IOS image should reside in flash memory before this feature is enabled.

If you plan to enter the **no service password-recovery** command, we recommend that you save a copy of the system configuration file in a location away from the switch or router. If you are using a switch that is operating in VTP transparent mode, we recommend that you also save a copy of the vlan.dat file in a location away from the switch.

Prerequisites

Always disable the feature before downgrading to an image that does not support this feature, because you cannot reset after the downgrade.

The configuration register boot bit must be enabled so that there is no way to break into ROMMON when this command is configured. Cisco IOS software should prevent the user from configuring the boot field in the config register.

Bit 6, which ignores the startup configuration and bit 8, which enables a break, should be set.

The Break key should be disabled while the router is booting up and disabled in Cisco IOS software when this feature is enabled.

SUMMARY STEPS

1. **enable**
2. **show version**
3. **configure terminal**

4. **config-register** *value*
5. **no service password-recovery**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show version** **Example:** `Router# show version`	Displays information about the system software, including configuration register settings. The configuration register must be set to autoboot before entering the **no service password-recovery** command.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**config-register** *value* **Example:** `Router(config)# config-register 0x2012`	(Optional) Changes the configuration register setting. • If necessary, change the configuration register setting so the router is set to autoboot.
Step 5	**no service password-recovery** **Example:** `Router(config)# no service password-recovery`	Disables password-recovery capability at the system console.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to EXEC mode.

Recovering a Device

To recover a device once the No Service Password-Recovery feature has been enabled, press the Break key within 5 seconds after the image decompresses during the boot. You are prompted to confirm the Break key action. When you confirm the action, the startup configuration is erased, the password-recovery procedure is enabled, and the router boots with the factory default configuration.

If you do not confirm the Break key action, the router boots normally with the No Service Password-Recovery feature enabled.

Examples

This section provides the following examples of the process:

- Confirmed Break, page 1867
- Unconfirmed Break, page 1868

Confirmed Break

```
PASSWORD RECOVERY FUNCTIONALITY IS DISABLED
program load complete, entry point: 0x80013000, size: 0x8396a8
Self decompressing the image :
################################################################################
################################################################################ [OK]
!The 5 second window starts now.

telnet> send break
telnet> send break
telnet> send break

Restricted Rights Legend

Use, duplication, or disclosure by the Government is subject to restrictions as set forth
in subparagraph (c) of the Commercial Computer Software - Restricted Rights clause at FAR
sec. 52.227-19 and subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134-1706

Cisco IOS Software, C831 Software (C831-K9O3SY6-M), Version 12.3(8)YA
Copyright (c) 1986-2004 by Cisco Systems, Inc.
Compiled Fri 13-Aug-04 03:21
Image text-base: 0x80013200, data-base: 0x81020514

PASSWORD RECOVERY IS DISABLED.

Do you want to reset the router to factory default configuration and proceed [y/n] ?
!The user enters "Y" here.

Reset router configuration to factory default.

This product contains cryptographic features and is subject to United States and local
country laws governing import, export, transfer and use. Delivery of Cisco cryptographic
products does not imply third-party authority to import, export, distribute or use
encryption.

Importers, exporters, distributors and users are responsible for compliance with U.S. and
local country laws. By using this product you agree to comply with applicable laws and
regulations. If you are unable to comply with U.S. and local laws, return this product
immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to export@cisco.com.

Cisco C831 (MPC857DSL) processor (revision 0x00) with 46695K/2457K bytes of memory.
Processor board ID 0000 (1314672220), with hardware revision 0000 CPU rev number 7
3 Ethernet interfaces
4 FastEthernet interfaces
128K bytes of NVRAM.
```

```
24576K bytes of processor board System flash (Read/Write)
2048K bytes of processor board Web flash (Read/Write)

         --- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: no
!Start up configuration is erased.

SETUP: new interface FastEthernet1 placed in "up" state
SETUP: new interface FastEthernet2 placed in "up" state
SETUP: new interface FastEthernet3 placed in "up" state
SETUP: new interface FastEthernet4 placed in "up" state

Press RETURN to get started!

Router>
Router> enable
Router# show startup configuration

startup-config is not present

Router# show running-config | incl service

no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption

!The "no service password-recovery" is disabled.
```

Unconfirmed Break

```
PASSWORD RECOVERY FUNCTIONALITY IS DISABLED

telnet> send break
program load complete, entry point: 0x80013000, size: 0x8396a8
Self decompressing the image :
################################################################################################ [OK]

telnet> send break
telnet> send break

Restricted Rights Legend

Use, duplication, or disclosure by the Government is subject to restrictions as set forth
in subparagraph (c) of the Commercial Computer Software - Restricted Rights clause at FAR
sec. 52.227-19 and subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134-1706

Cisco IOS Software, C831 Software (C831-K9O3SY6-M), Version 12.3(8)YA
Copyright (c) 1986-2004 by Cisco Systems, Inc.
Compiled Fri 13-Aug-04 03:21
Image text-base: 0x80013200, data-base: 0x81020514

PASSWORD RECOVERY IS DISABLED.
Do you want to reset the router to factory default configuration and proceed [y/n] ?
!The user enters "N" here.
```

```
This product contains cryptographic features and is subject to United States and local
country laws governing import, export, transfer and use. Delivery of Cisco cryptographic
products does not imply third-party authority to import, export, distribute or use
encryption.

Importers, exporters, distributors and users are responsible for compliance with U.S. and
local country laws. By using this product you agree to comply with applicable laws and
regulations. If you are unable to comply with U.S. and local laws, return this product
immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to export@cisco.com.
Cisco C831 (MPC857DSL) processor (revision 0x00) with 46695K/2457K bytes of memory.
Processor board ID 0000 (1314672220), with hardware revision 0000
CPU rev number 7
3 Ethernet interfaces
4 FastEthernet interfaces
128K bytes of NVRAM.
24576K bytes of processor board System flash (Read/Write)
2048K bytes of processor board Web flash (Read/Write)

Press RETURN to get started!
!The Cisco IOS software boots as if it is not interrupted.

Router> enable
Router#
Router# show startup config

Using 984 out of 131072 bytes
!
version 12.3
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
no service password-recovery
!
hostname Router
!
boot-start-marker
boot-end-marker
!
memory-size iomem 5
!
no aaa new-model
ip subnet-zero
!
ip ips po max-events 100
no ftp-server write-enable
!
interface Ethernet0
 no ip address
 shutdown
!
interface Ethernet1
 no ip address
 shutdown
 duplex auto
!
interface Ethernet2
 no ip address
 shutdown
```

```
!
interface FastEthernet1
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet2
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet3
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet4
 no ip address
 duplex auto
 speed auto
!
ip classless
!
ip http server
no ip http secure-server
!
control-plane
!
line con 0
 no modem enable
 transport preferred all
 transport output all
line aux 0
line vty 0 4
!
scheduler max-task-time 5000
end

Router# show running-config | incl service

no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
no service password-recovery
end
```

Configuration Examples for No Service Password-Recovery

This section provides the following configuration example:

- Disabling Password Recovery: Example, page 1871

Disabling Password Recovery: Example

The following example shows how to obtain the configuration register setting (which is set to autoboot), disable password recovery capability, and then verify that the configuration persists through a system reload:

```
Router# show version

Cisco Internetwork Operating System Software
IOS (tm) 5300 Software (C7200-P-M), Version 12.3(8)YA, RELEASE SOFTWARE (fc1)
TAC Support: http://www.cisco.com/tac
Copyright (c) 1986-2004 by Cisco Systems, Inc.
Compiled Wed 05-Mar-04 10:16 by xxx
Image text-base: 0x60008954, data-base: 0x61964000

ROM: System Bootstrap, Version 12.3(8)YA, RELEASE SOFTWARE (fc1)
.
.
.
125440K bytes of ATA PCMCIA card at slot 0 (Sector size 512 bytes).
8192K bytes of Flash internal SIMM (Sector size 256K).
Configuration register is 0x2102

Router# configure terminal

Router(config)# no service password-recovery

WARNING:
Executing this command will disable the password recovery mechanism.
Do not execute this command without another plan for password recovery.

Are you sure you want to continue? [yes/no]: yes
.
.
.
Router(config)# exit
Router#
Router# reload

Proceed with reload? [confirm] yes

00:01:54: %SYS-5-RELOAD: Reload requested
System Bootstrap, Version 12.3...
Copyright (c) 1994-2004 by cisco Systems, Inc.
C7400 platform with 262144 Kbytes of main memory

PASSWORD RECOVERY FUNCTIONALITY IS DISABLED
.
.
.
```

Additional References

The following sections provide references related to the No Service Password-Recovery feature.

Related Documents

Related Topic	Document Title
Setting, changing, and recovering lost passwords	Refer to the "Configuring Passwords and Privileges" chapter in the *Cisco IOS Security Configuration Guide*, Release 12.3
Loading system images and rebooting	Refer to the "File Management" section in the *Cisco IOS Configuration Fundamentals and Network Management Configuration Guide*, Release 12.3
Security commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Refer to the *Cisco IOS Security Command Reference*, Release 12.3T

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **service password-recovery**

Appendixes

RADIUS Attributes

This part consists of the following:

- RADIUS Attributes Overview and RADIUS IETF Attributes
- RADIUS Vendor-Proprietary Attributes
- RADIUS Vendor-Specific Attributes and RADIUS Disconnect-Cause Attribute Values
- Connect-Info RADIUS Attribute 77
- Encrypted Vendor-Specific Attributes
- Local AAA Server
- Per-User QoS via AAA Policy Name
- RADIUS Attribute 5 (NAS-Port) Format Specified on a Per-Server Group Level
- RADIUS Attribute 8 (Framed-IP-Address) in Access Requests
- RADIUS Attribute 82: Tunnel Assignment ID
- RADIUS Attribute 104
- RADIUS Progress Codes
- RADIUS Timeout Set During Pre-Authentication
- RADIUS Tunnel Attribute Extensions
- V.92 Reporting Using RADIUS Attribute v.92-info

RADIUS Attributes Overview and RADIUS IETF Attributes

Remote Authentication Dial-In User Service (RADIUS) attributes are used to define specific authentication, authorization, and accounting (AAA) elements in a user profile, which is stored on the RADIUS daemon. This appendix lists the RADIUS attributes currently supported.

In This Appendix

This appendix contains the following sections:

- RADIUS Attributes Overview
- RADIUS IETF Attributes
- RADIUS Vendor-Proprietary Attributes
- RADIUS Vendor-Specific Attributes and RADIUS Disconnect-Cause Attribute Values
- RADIUS Disconnect-Cause Attribute Values

RADIUS Attributes Overview

This section contains information important to understanding how RADIUS attributes exchange AAA information between a client and server and includes the following sections:

- IETF Attributes Versus VSAs
- RADIUS Packet Format
- RADIUS Files
- Supporting Documentation

IETF Attributes Versus VSAs

RADIUS Internet Engineering Task Force (IETF) attributes are the original set of 255 standard attributes that are used to communicate AAA information between a client and a server. Because IETF attributes are standard, the attribute data is predefined and well known; thus all clients and servers who exchange AAA information via IETF attributes must agree on attribute data such as the exact meaning of the attributes and the general bounds of the values for each attribute.

RADIUS vendor-specific attributes (VSAs) derived from one IETF attribute—vendor-specific (attribute 26). Attribute 26 allows a vendor to create an additional 255 attributes however they wish. That is, a vendor can create an attribute that does not match the data of any IETF attribute and encapsulate it behind attribute 26; thus, the newly created attribute is accepted if the user accepts attribute 26.

For more information on VSAs, refer to the section "RADIUS Vendor-Specific Attributes and RADIUS Disconnect-Cause Attribute Values" later in this appendix.

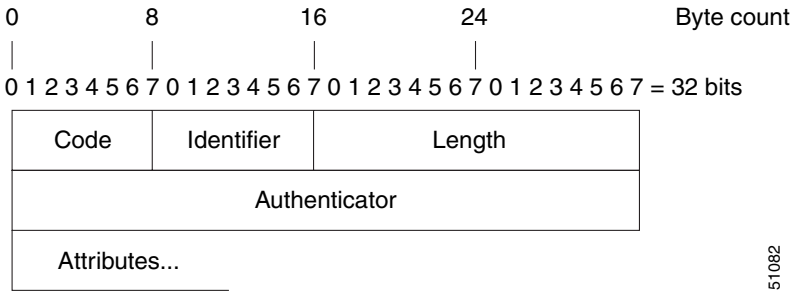
RADIUS Packet Format

The data between a RADIUS server and a RADIUS client is exchanged in RADIUS packets. The data fields are transmitted from left to right.

Figure 125 shows the fields within a RADIUS packet.

Note For a diagram of VSAs, which is an extension of Figure 125, refer to Figure 126.

Figure 125 RADIUS Packet Diagram



Each RADIUS packet contains the following information:

- Code—The code field is one octet; it identifies one of the following types of RADIUS packets:
  - Access-Request (1)
  - Access-Accept (2)
  - Access-Reject (3)
  - Accounting-Request (4)
  - Accounting-Response (5)
- Identifier—The identifier field is one octet; it helps the RADIUS server match requests and responses and detect duplicate requests.
- Length—The length field is two octets; it specifies the length of the entire packet.
- Authenticator—The authenticator field is 16 octets. The most significant octet is transmitted first; it is used to authenticate the reply from the RADIUS server. Two types of authenticators are as follows:
  - Request-Authentication: Available in Access-Request and Accounting-Request packets
  - Response-Authenticator: Available in Access-Accept, Access-Reject, Access-Challenge, and Accounting-Response packets

RADIUS Packet Types

The following list defines the various types of RADIUS packet types that can contain attribute information:

Access-Request—Sent from a client to a RADIUS server. The packet contains information that allows the RADIUS server to determine whether to allow access to a specific network access server (NAS), which will allow access to the user. Any user performing authentication *must* submit an Access-Request packet. Once an Access-Request packet is received, the RADIUS server *must* forward a reply.

Access-Accept—Once a RADIUS server receives an Access-Request packet, it must send an Access-Accept packet if all attribute values in the Access-Request packet are acceptable. Access-Accept packets provide the configuration information necessary for the client to provide service to the user.

Access-Reject—Once a RADIUS server receives an Access-Request packet, it must send an Access-Reject packet if any of the attribute values are not acceptable.

Access-Challenge—Once the RADIUS server receives an Access-Accept packet, it can send the client an Access-Challenge packet, which requires a response. If the client does not know how to respond or if the packets are invalid, the RADIUS server discards the packets. If the client responds to the packet, a new Access-Request packet should be sent with the original Access-Request packet.

Accounting-Request—Sent from a client to a RADIUS accounting server, which provides accounting information. If the RADIUS server successfully records the Accounting-Request packet, it must submit an Accounting Response packet.

Accounting-Response—Sent by the RADIUS accounting server to the client to acknowledge that the Accounting-Request has been received and recorded successfully.

RADIUS Files

Understanding the types of files used by RADIUS is important for communicating AAA information from a client to a server. Each file defines a level of authentication or authorization for the user: The dictionary file defines which attributes the user's NAS can implement; the clients file defines which users are allowed to make requests to the RADIUS server; the users files defines which user requests the RADIUS server will authenticate based on security and configuration data.

- Dictionary File
- Clients File
- Users File

Dictionary File

A dictionary file provides a list of attributes that are dependent upon which attributes your NAS supports. However, you can add your own set of attributes to your dictionary for custom solutions. It defines attribute values, thereby allowing you to interpret attribute output such as parsing requests. A dictionary file contains the following information:

- Name—The ASCII string "name" of the attribute, such as User-Name.
- ID—The numerical "name" of the attribute; for example, User-Name attribute is attribute 1.
- Value type—Each attribute can be specified as one of the following five value types:
  - abinary—0 to 254 octets.
  - date—32-bit value in big endian order. For example, seconds since 00:00:00 GMT, JAN. 1, 1970.

- ipaddr—4 octets in network byte order.
- integer—32-bit value in big endian order (high byte first).
- string—0 to 253 octets.

When the data type for a particular attribute is an integer, you can optionally expand the integer to equate to some string. The follow sample dictionary includes an integer-based attribute and its corresponding values:

```
# dictionary sample of integer entry
#
ATTRIBUTE       Service-Type            6                       integer
VALUE           Service-Type            Login                      1
VALUE           Service-Type            Framed                     2
VALUE           Service-Type            Callback-Login             3
VALUE           Service-Type            Callback-Framed            4
VALUE           Service-Type            Outbound                   5
VALUE           Service-Type            Administrative             6
VALUE           Service-Type            NAS-Prompt                 7
VALUE           Service-Type            Authenticate-Only          8
VALUE           Service-Type            Callback-NAS-Prompt        9
VALUE           Service-Type            Call-Check                 10
VALUE           Service-Type            Callback-Administrative    11
```

Clients File

A clients file is important because it contains a list of RADIUS clients that are allowed to send authentication and accounting requests to the RADIUS server. To receive authentication, the name and authentication key the client sends the server must be an exact match with the data contained in clients file.

The following is an example of a clients file. The key, as shown in this example, must be the same as the **radius-server key** *SomeSecret* command.

```
#Client Name        Key
#----------------  ---------------
10.1.2.3:256        test
nas01               bananas
nas02               MoNkEys
nas07.foo.com       SomeSecret
```

Users File

A RADIUS users file contains an entry for each user that the RADIUS server will authenticate; each entry, which is also referred to as a user profile, establishes an attribute the user can access.

The first line in any user profile is always a “user access” line; that is, the server must check the attributes on the first line before it can grant access to the user. The first line contains the name of the user, which can be up to 252 characters, followed by authentication information such as the password of the user.

Additional lines, which are associated with the user access line, indicate the attribute reply that is sent to the requesting client or server. The attributes sent in the reply must be defined in the dictionary file.

When looking at a user file, please note the the data to the left of the equal (=) character is an attribute defined in the dictionary file, and the data to the right of the equal character is the configuration data.

Note A blank line cannot appear anywhere within a user profile.

The following is an example of a RADIUS user profile (Merit Daemon format). In this example, the user name is cisco.com, the password is cisco, and the user can access five tunnel attributes.

```
# This user profile includes RADIUS tunneling attributes
cisco.com  Password="cisco" Service-Type=Outbound
     Tunnel-Type = :1:L2TP
     Tunnel-Medium-Type = :1:IP
     Tunnel-Server-Endpoint = :1:10.0.0.1
     Tunnel-Password = :1:"welcome"
     Tunnel-Assignment-ID = :1:"nas"
```

Supporting Documentation

For more information on RADIUS IETF and Vendor-Proprietary Attributes, refer to the following documents:

- Cisco AAA Implementation Case Study
- "Configuring RADIUS" "Configuring Authentication," "Configuring Authorization," and "Configuring Accounting" chapters in this book.

  Refer to these chapters for information on how RADIUS is used with AAA.
- IETF RADIUS RFCs
  - RFC 2865, *Remote Authentication Dial In User Service (RADIUS)*
  - RFC 2866, *RADIUS Accounting*
  - RFC 2867, *RADIUS Accounting Modifications for Tunnel Protocol Support*
  - RFC 2868, *RADIUS Attributes for Tunnel Protocol Support*
  - RFC 2869, *RADIUS Extensions*
- RADIUS Vendor-Specific Attributes Voice Implementation Guide

RADIUS IETF Attributes

Note In the Cisco IOS Release 12.2 for RADIUS tunnel attributes, 32 tagged tunnel sets are supported for L2TP.

This section contains the following sections:

- Supported RADIUS IETF Attributes
- Comprehensive List of RADIUS Attribute Descriptions

Supported RADIUS IETF Attributes

Table 71 lists Cisco-supported IETF RADIUS attributes and the Cisco IOS release in which they are implemented. In cases where the attribute has a security server-specific format, the format is specified.

Refer to Table 72 for a description of each listed attribute.

Note Attributes implemented in special (AA) or early development (T) releases will be added to the next mainline image.

Table 71 Supported RADIUS IETF Attributes

Number	IETF Attribute	11.1	11.2	11.3	11.3 AA	11.3T	12.0	12.1	12.2
1	User-Name	yes	yes	yes	yes	yes	yes	yes	yes
2	User-Password	yes	yes	yes	yes	yes	yes	yes	yes
3	CHAP-Password	yes	yes	yes	yes	yes	yes	yes	yes
4	NAS-IP Address	yes	yes	yes	yes	yes	yes	yes	yes
5	NAS-Port	yes	yes	yes	yes	yes	yes	yes	yes
6	Service-Type	yes	yes	yes	yes	yes	yes	yes	yes
7	Framed-Protocol	yes	yes	yes	yes	yes	yes	yes	yes
8	Framed-IP-Address	yes	yes	yes	yes	yes	yes	yes	yes
9	Framed-IP-Netmask	yes	yes	yes	yes	yes	yes	yes	yes
10	Framed-Routing	yes	yes	yes	yes	yes	yes	yes	yes
11	Filter-Id	yes	yes	yes	yes	yes	yes	yes	yes
12	Framed-MTU	yes	yes	yes	yes	yes	yes	yes	yes
13	Framed-Compression	yes	yes	yes	yes	yes	yes	yes	yes
14	Login-IP-Host	yes	yes	yes	yes	yes	yes	yes	yes
15	Login-Service	yes	yes	yes	yes	yes	yes	yes	yes
16	Login-TCP-Port	yes	yes	yes	yes	yes	yes	yes	yes
18	Reply-Message	yes	yes	yes	yes	yes	yes	yes	yes
19	Callback-Number	no	no	no	no	no	no	yes	yes
20	Callback-ID	no	no	no	no	no	no	no	no
22	Framed-Route	yes	yes	yes	yes	yes	yes	yes	yes
23	Framed-IPX-Network	no	no	no	no	no	no	no	no
24	State	yes	yes	yes	yes	yes	yes	yes	yes
25	Class	yes	yes	yes	yes	yes	yes	yes	yes
26	Vendor-Specific	yes	yes	yes	yes	yes	yes	yes	yes
27	Session-Timeout	yes	yes	yes	yes	yes	yes	yes	yes
28	Idle-Timeout	yes	yes	yes	yes	yes	yes	yes	yes
29	Termination-Action	no	no	no	no	no	no	no	no
30	Called-Station-Id	yes	yes	yes	yes	yes	yes	yes	yes
31	Calling-Station-Id	yes	yes	yes	yes	yes	yes	yes	yes
32	NAS-Identifier	no	no	no	no	no	no	no	yes
33	Proxy-State	no	no	no	no	no	no	no	no
34	Login-LAT-Service	yes	yes	yes	yes	yes	yes	yes	yes

Table 71 Supported RADIUS IETF Attributes (continued)

Number	IETF Attribute	11.1	11.2	11.3	11.3 AA	11.3T	12.0	12.1	12.2
35	Login-LAT-Node	no	no	no	no	no	no	no	yes
36	Login-LAT-Group	no	no	no	no	no	no	no	no
37	Framed-AppleTalk-Link	no	no	no	no	no	no	no	no
38	Framed-AppleTalk- Network	no	no	no	no	no	no	no	no
39	Framed-AppleTalk-Zone	no	no	no	no	no	no	no	no
40	Acct-Status-Type	yes	yes	yes	yes	yes	yes	yes	yes
41	Acct-Delay-Time	yes	yes	yes	yes	yes	yes	yes	yes
42	Acct-Input-Octets	yes	yes	yes	yes	yes	yes	yes	yes
43	Acct-Output-Octets	yes	yes	yes	yes	yes	yes	yes	yes
44	Acct-Session-Id	yes	yes	yes	yes	yes	yes	yes	yes
45	Acct-Authentic	yes	yes	yes	yes	yes	yes	yes	yes
46	Acct-Session-Time	yes	yes	yes	yes	yes	yes	yes	yes
47	Acct-Input-Packets	yes	yes	yes	yes	yes	yes	yes	yes
48	Acct-Output-Packets	yes	yes	yes	yes	yes	yes	yes	yes
49	Acct-Terminate-Cause	no	no	no	yes	yes	yes	yes	yes
50	Acct-Multi-Session-Id	no	yes	yes	yes	yes	yes	yes	yes
51	Acct-Link-Count	no	yes	yes	yes	yes	yes	yes	yes
52	Acct-Input-Gigawords	no	no	no	no	no	no	no	no
53	Acct-Output-Gigawords	no	no	no	no	no	no	no	no
55	Event-Timestamp	no	no	no	no	no	no	no	yes
60	CHAP-Challenge	yes	yes	yes	yes	yes	yes	yes	yes
61	NAS-Port-Type	yes	yes	yes	yes	yes	yes	yes	yes
62	Port-Limit	yes	yes	yes	yes	yes	yes	yes	yes
63	Login-LAT-Port	no	no	no	no	no	no	no	no
64	Tunnel-Type[1]	no	no	no	no	no	no	yes	yes
65	Tunnel-Medium-Type[1]	no	no	no	no	no	no	yes	yes
66	Tunnel-Client-Endpoint	no	no	no	no	no	no	yes	yes
67	Tunnel-Server-Endpoint[1]	no	no	no	no	no	no	yes	yes
68	Acct-Tunnel-Connection-ID	no	no	no	no	no	no	yes	yes
69	Tunnel-Password[1]	no	no	no	no	no	no	yes	yes
70	ARAP-Password	no	no	no	no	no	no	no	no
71	ARAP-Features	no	no	no	no	no	no	no	no
72	ARAP-Zone-Access	no	no	no	no	no	no	no	no
73	ARAP-Security	no	no	no	no	no	no	no	no
74	ARAP-Security-Data	no	no	no	no	no	no	no	no
75	Password-Retry	no	no	no	no	no	no	no	no

Table 71 Supported RADIUS IETF Attributes (continued)

Number	IETF Attribute	11.1	11.2	11.3	11.3 AA	11.3T	12.0	12.1	12.2
76	Prompt	no	no	no	no	no	no	yes	yes
77	Connect-Info	no	no	no	no	no	no	no	yes
78	Configuration-Token	no	no	no	no	no	no	no	no
79	EAP-Message	no	no	no	no	no	no	no	no
80	Message-Authenticator	no	no	no	no	no	no	no	no
81	Tunnel-Private-Group-ID	no	no	no	no	no	no	no	no
82	Tunnel-Assignment-ID[1]	no	no	no	no	no	no	yes	yes
83	Tunnel-Preference	no	no	no	no	no	no	no	yes
84	ARAP-Challenge-Response	no	no	no	no	no	no	no	no
85	Acct-Interim-Interval	no	no	no	no	no	no	yes	yes
86	Acct-Tunnel-Packets-Lost	no	no	no	no	no	no	no	no
87	NAS-Port-ID	no	no	no	no	no	no	no	no
88	Framed-Pool	no	no	no	no	no	no	no	no
90	Tunnel-Client-Auth-ID[2]	no	no	no	no	no	no	no	yes
91	Tunnel-Server-Auth-ID	no	no	no	no	no	no	no	yes
200	IETF-Token-Immediate	no	no	no	no	no	no	no	no

1. This RADIUS attribute complies with the following two draft IETF documents: RFC 2868 *RADIUS Attributes for Tunnel Protocol Support* and RFC 2867 *RADIUS Accounting Modifications for Tunnel Protocol Support*.
2. This RADIUS attribute complies withRFC 2865 and RFC 2868.

Comprehensive List of RADIUS Attribute Descriptions

Table 72 lists and describes IETF RADIUS attributes. In cases where the attribute has a security server-specific format, the format is specified.

Table 72 RADIUS IETF Attributes

Number	IETF Attribute	Description
1	User-Name	Indicates the name of the user being authenticated by the RADIUS server.
2	User-Password	Indicates the user's password or the user's input following an Access-Challenge. Passwords longer than 16 characters are encrypted using RFC 2865 specifications.
3	CHAP-Password	Indicates the response value provided by a PPP Challenge-Handshake Authentication Protocol (CHAP) user in response to an Access-Challenge.
4	NAS-IP Address	Specifies the IP address of the network access server that is requesting authentication. The default value is 0.0.0.0/0.

Table 72 RADIUS IETF Attributes (continued)

Number	IETF Attribute	Description
5	NAS-Port	Indicates the physical port number of the network access server that is authenticating the user. The NAS-Port value (32 bits) consists of one or two 16-bit values (depending on the setting of the **radius-server extended-portnames** command). Each 16-bit number should be viewed as a 5-digit decimal integer for interpretation as follows: For asynchronous terminal lines, async network interfaces, and virtual async interfaces, the value is **00ttt**, where **ttt** is the line number or async interface unit number. For ordinary synchronous network interface, the value is **10xxx**. For channels on a primary rate ISDN interface, the value is **2ppcc**. For channels on a basic rate ISDN interface, the value is **3bb0c**. For other types of interfaces, the value is **6nnss**.
6	Service-Type	Indicates the type of service requested or the type of service to be provided. • In a request: Framed for known PPP or SLIP connection. Administrative-user for **enable** command. • In response: Login—Make a connection. Framed—Start SLIP or PPP. Administrative User—Start an EXEC or **enable ok**. Exec User—Start an EXEC session. Service type is indicated by a particular numeric value as follows: • 1: Login • 2: Framed • 3: Callback-Login • 4: Callback-Framed • 5: Outbound • 6: Administrative • 7: NAS-Prompt • 8: Authenticate Only • 9: Callback-NAS-Prompt
7	Framed-Protocol	Indicates the framing to be used for framed access. No other framing is allowed. Framing is indicated by a numeric value as follows: • 1: PPP • 2: SLIP • 3: ARA • 4: Gandalf-proprietary single-link/multilink protocol • 5: Xylogics-proprietary IPX/SLIP

Table 72 RADIUS IETF Attributes (continued)

Number	IETF Attribute	Description
8	Framed-IP-Address	Indicates the IP address to be configured for the user, by sending the IP address of a user to the RADIUS server in the access-request. To enable this command, use the **radius-server attribute 8 include-in-access-req** command in global configuration mode.
9	Framed-IP-Netmask	Indicates the IP netmask to be configured for the user when the user is a router to a network. This attribute value results in a static route being added for Framed-IP-Address with the mask specified.
10	Framed-Routing	Indicates the routing method for the user when the user is a router to a network. Only "None" and "Send and Listen" values are supported for this attribute. Routing method is indicated by a numeric value as follows: • 0: None • 1: Send routing packets • 2: Listen for routing packets • 3: Send routing packets and listen for routing packets
11	Filter-Id	Indicates the name of the filter list for the user and is formatted as follows: %d, %d.in, or %d.out. This attribute is associated with the most recent service-type command. For login and EXEC, use %d or %d.out as the line access list value from 0 to 199. For Framed service, use %d or %d.out as interface output access list, and %d.in for input access list. The numbers are self-encoding to the protocol to which they refer.
12	Framed-MTU	Indicates the maximum transmission unit (MTU) that can be configured for the user when the MTU is not negotiated by PPP or some other means.
13	Framed-Compression	Indicates a compression protocol used for the link. This attribute results in a "/compress" being added to the PPP or SLIP autocommand generated during EXEC authorization. Not currently implemented for non-EXEC authorization. Compression protocol is indicated by a numeric value as follows: • 0: None • 1: VJ-TCP/IP header compression • 2: IPX header compression
14	Login-IP-Host	Indicates the host to which the user will connect when the Login-Service attribute is included. (This begins immediately after login.)
15	Login-Service	Indicates the service that should be used to connect the user to the login host. Service is indicated by a numeric value as follows: • 0: Telnet • 1: Rlogin • 2: TCP-Clear • 3: PortMaster • 4: LAT
16	Login-TCP-Port	Defines the TCP port with which the user is to be connected when the Login-Service attribute is also present.

Table 72 RADIUS IETF Attributes (continued)

Number	IETF Attribute	Description
18	Reply-Message	Indicates text that might be displayed to the user via the RADIUS server. You can include this attribute in user files; however, you cannot exceed a maximum of 16 Replyp-Message entries per profile.
19	Callback-Number	Defines a dialing string to be used for callback.
20	Callback-ID	Defines the name (consisting of one or more octets) of a place to be called, to be interpreted by the network access server.
22	Framed-Route	Provides routing information to be configured for the user on this network access server. The RADIUS RFC format (net/bits [router [metric]]) and the old style dotted mask (net mask [router [metric]]) are supported. If the router field is omitted or 0, the peer IP address is used. Metrics are currently ignored. This attribute is access-request packets.
23	Framed-IPX-Network	Defines the IPX network number configured for the user.
24	State	Allows state information to be maintained between the network access server and the RADIUS server. This attribute is applicable only to CHAP challenges.
25	Class	(Accounting) Arbitrary value that the network access server includes in all accounting packets for this user if supplied by the RADIUS server.
26	Vendor-Specific	Allows vendors to support their own extended attributes not suitable for general use. The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. Cisco's vendor-ID is 9, and the supported option has vendor-type 1, which is named “cisco-avpair.” The value is a string of the format: `protocol : attribute sep value` “Protocol” is a value of the Cisco “protocol” attribute for a particular type of authorization. “Attribute” and “value” are an appropriate AV pair defined in the Cisco TACACS+ specification, and “sep” is “=” for mandatory attributes and “*” for optional attributes. This allows the full set of features available for TACACS+ authorization to also be used for RADIUS. For example: `cisco-avpair= ”ip:addr-pool=first“` `cisco-avpair= ”shell:priv-lvl=15“` The first example causes Cisco’s “multiple named ip address pools” feature to be activated during IP authorization (during PPP’s IPCP address assignment). The second example causes a user logging in from a network access server to have immediate access to EXEC commands. Table 71 lists supported vendor-specific RADIUS attributes (IETF attribute 26). The “TACACS+ Attribute-Value Pairs” appendix provides a complete list of supported TACACS+ attribute-value (AV) pairs that can be used with IETF attribute 26. (RFC 2865)
27	Session-Timeout	Sets the maximum number of seconds of service to be provided to the user before the session terminates. This attribute value becomes the per-user “absolute timeout.”
28	Idle-Timeout	Sets the maximum number of consecutive seconds of idle connection allowed to the user before the session terminates. This attribute value becomes the per-user “session-timeout.”

Table 72 RADIUS IETF Attributes (continued)

Number	IETF Attribute	Description
29	Termination-Action	Termination is indicated by a numeric value as follows: • 0: Default • 1: RADIUS request
30	Called-Station-Id	(Accounting) Allows the network access server to send the telephone number the user called as part of the Access-Request packet (using Dialed Number Identification Service [DNIS] or similar technology). This attribute is only supported on ISDN, and modem calls on the Cisco AS5200 if used with PRI.
31	Calling-Station-Id	(Accounting) Allows the network access server to send the telephone number the call came from as part of the Access-Request packet (using Automatic Number Identification or similar technology). This attribute has the same value as "remote-addr" from TACACS+. This attribute is only supported on ISDN, and modem calls on the Cisco AS5200 if used with PRI.
32	NAS-Identifier	String identifying the network access server originating the Access-Request. Use the **radius-server attribute 32 include-in-access-req** global configuration command to send RADIUS attribute 32 in an Access-Request or Accounting-Request. By default, the FQDN is sent in the attribute when the format is not specified.
33	Proxy-State	Attribute that can be sent by a proxy server to another server when forwarding Access-Requests; this must be returned unmodified in the Access-Accept, Access-Reject or Access-Challenge and removed by the proxy server before sending the response to the network access server.
34	Login-LAT-Service	Indicates the system with which the user is to be connected by LAT. This attribute is only available in the EXEC mode.
35	Login-LAT-Node	Indicates the node with which the user is to be automatically connected by LAT.
36	Login-LAT-Group	Identifies the LAT group codes that this user is authorized to use.
37	Framed-AppleTalk-Link	Indicates the AppleTalk network number that should be used for serial links to the user, which is another AppleTalk router.
38	Framed-AppleTalk-Network	Indicates the AppleTalk network number that the network access server uses to allocate an AppleTalk node for the user.
39	Framed-AppleTalk-Zone	Indicates the AppleTalk Default Zone to be used for this user.
40	Acct-Status-Type	(Accounting) Indicates whether this Accounting-Request marks the beginning of the user service (start) or the end (stop).
41	Acct-Delay-Time	(Accounting) Indicates how many seconds the client has been trying to send a particular record.
42	Acct-Input-Octets	(Accounting) Indicates how many octets have been received from the port over the course of this service being provided.
43	Acct-Output-Octets	(Accounting) Indicates how many octets have been sent to the port in the course of delivering this service.
44	Acct-Session-Id	(Accounting) A unique accounting identifier that makes it easy to match start and stop records in a log file. Acct-Session ID numbers restart at 1 each time the router is power cycled or the software is reloaded. To send this attribute in access-request packets, use the **radius-server attribute 44 include-in-access-req** command in global configuration mode.

Table 72 RADIUS IETF Attributes (continued)

Number	IETF Attribute	Description
45	Acct-Authentic	(Accounting) Indicates how the user was authenticated, whether by RADIUS, the network access server itself, or another remote authentication protocol. This attribute is set to "radius" for users authenticated by RADIUS; "remote" for TACACS+ and Kerberos; or "local" for local, enable, line, and if-needed methods. For all other methods, the attribute is omitted.
46	Acct-Session-Time	(Accounting) Indicates how long (in seconds) the user has received service.
47	Acct-Input-Packets	(Accounting) Indicates how many packets have been received from the port over the course of this service being provided to a framed user.
48	Acct-Output-Packets	(Accounting) Indicates how many packets have been sent to the port in the course of delivering this service to a framed user.
49	Acct-Terminate-Cause	(Accounting) Reports details on why the connection was terminated. Termination causes are indicated by a numeric value as follows: **1.** User request **2.** Lost carrier **3.** Lost service **4.** Idle timeout **5.** Session timeout **6.** Admin reset **7.** Admin reboot **8.** Port error **9.** NAS error **10.** NAS request **11.** NAS reboot **12.** Port unneeded **13.** Port pre-empted **14.** Port suspended **15.** Service unavailable **16.** Callback **17.** User error **18.** Host request **Note** For attribute 49, Cisco IOS supports values 1 to 6, 9, 12, and 15 to 18.
50	Acct-Multi-Session-Id	(Accounting) A unique accounting identifier used to link multiple related sessions in a log file. Each linked session in a multilink session has a unique Acct-Session-Id value, but shares the same Acct-Multi-Session-Id.
51	Acct-Link-Count	(Accounting) Indicates the number of links known in a given multilink session at the time an accounting record is generated. The network access server can include this attribute in any accounting request that might have multiple links.

Table 72 RADIUS IETF Attributes (continued)

Number	IETF Attribute	Description
52	Acct-Input-Gigawords	Indicates how many times the Acct-Input-Octets counter has wrapped around 2^32 over the course of the provided service.
53	Acct-Output-Gigawords	Indicates how many times the Acct-Output-Octets counter has wrapped around 2^32 while delivering service.
55	Event-Timestamp	Records the time that the event occurred on the NAS; the timestamp sent in attribute 55 is in seconds since January 1, 1970 00:00 UTC. To send RADIUS attribute 55 in accounting packets, use the **radius-server attribute 55 include-in-acct-req** command. **Note** Before the Event-Timestamp attribute can be sent in accounting packets, you *must* configure the clock on the router. (For information on setting the clock on your router, refer to section "Performing Basic System Management" in the chapter "System Management" of the *Cisco IOS Configuration Fundamentals Configuration Guide*.) To avoid configuring the clock on the router every time the router is reloaded, you can enable the **clock calendar-valid** command. (For information on this command, refer to the chapter "Basic System Management Commands" in the *Cisco IOS Configuration Fundamentals Command Reference.*
60	CHAP-Challenge	Contains the Challenge Handshake Authentication Protocol challenge sent by the network access server to a PPP CHAP user.
61	NAS-Port-Type	Indicates the type of physical port the network access server is using to authenticate the user. Physical ports are indicated by a numeric value as follows: • 0: Asynchronous • 1: Synchronous • 2: ISDN-Synchronous • 3: ISDN-Asynchronous (V.120) • 4: ISDN-Asynchronous (V.110) • 5: Virtual
62	Port-Limit	Sets the maximum number of ports provided to the user by the NAS.
63	Login-LAT-Port	Defines the port with which the user is to be connected by LAT.
64	Tunnel-Type[1]	Indicates the tunneling protocol(s) used. Cisco IOS software supports two possible values for this attribute: L2TP and L2F. If this attribute is not set, L2F is used as a default.
65	Tunnel-Medium-Type[1]	Indicates the transport medium type to use to create a tunnel. This attribute has only one available value for this release: IP. If no value is set for this attribute, IP is used as the default.

Table 72 RADIUS IETF Attributes (continued)

Number	IETF Attribute	Description
66	Tunnel-Client-Endpoint	Contains the address of the initiator end of the tunnel. It *may* be included in both Access-Request and Access-Accept packets to indicate the address from which a new tunnel is to be initiated. If the Tunnel-Client-Endpoint attribute is included in an Access-Request packet, the RADIUS server should take the value as a hint; the server is not obligated to honor the hint, however. This attribute *should* be included in Accounting-Request packets that contain Acct-Status-Type attributes with values of either Start or Stop, in which case it indicates the address from which the tunnel was initiated. This attribute, along with the Tunnel-Server-Endpoint and Acct-Tunnel-Connection-ID attributes, may be used to provide a globally unique means to identify a tunnel for accounting and auditing purposes. An enhancement has been added for the network access server to accept a value of 127.0.0.X for this attribute such that: 127.0.0.0 would indicate that loopback0 IP address is to be used 127.0.0.1 would indicate that loopback1 IP address is to be used ... 127.0.0.X would indicate that loopbackX IP address is to be used for the actual tunnel client endpoint IP address. This enhancement adds scalability across multiple network access servers.
67	Tunnel-Server-Endpoint[1]	Indicates the address of the server end of the tunnel. The format of this attribute varies depending on the value of Tunnel-Medium-Type. Because this release only supports IP as a tunnel medium type, the IP address or the host name of LNS is valid for this attribute.
68	Acct-Tunnel-Connection-ID	Indicates the identifier assigned to the tunnel session. This attribute *should* be included in Accounting-Request packets that contain an Acct-Status-Type attribute having the value Start, Stop, or any of the values described above. This attribute, along with the Tunnel-Client-Endpoint and Tunnel-Server-Endpoint attributes, may be used to provide a means to uniquely identify a tunnel session for auditing purposes.
69	Tunnel-Password[1]	Defines the password to be used to authenticate to a remote server. This attribute is converted into different AAA attributes based on the value of Tunnel-Type: AAA_ATTR_l2tp_tunnel_pw (L2TP), AAA_ATTR_nas_password (L2F), and AAA_ATTR_gw_password (L2F). By default, all passwords received are encrypted, which can cause authorization failures when a NAS attempts to decrypt a non-encrypted password. To enable attribute 69 to receive non-encrypted passwords, use the **radius-server attribute 69 clear** global configuration command.
70	ARAP-Password	Identifies an Access-Request packet containing a Framed-Protocol of ARAP.
71	ARAP-Features	Includes password information that the NAS should send to the user in an ARAP "feature flags" packet.
72	ARAP-Zone-Access	Indicates how the ARAP zone list for the user should be used.
73	ARAP-Security	Identifies the ARAP Security Module to be used in an Access-Challenge packet.
74	ARAP-Security-Data	Contains the actual security module challenge or response. It can be found in Access-Challenge and Access-Request packets.
75	Password-Retry	Indicates how many times a user may attempt authentication before being disconnected.

Table 72 RADIUS IETF Attributes (continued)

Number	IETF Attribute	Description
76	Prompt	Indicates to the NAS whether it should echo the user's response as it is entered or not echo it. (0=no echo, 1=echo)
77	Connect-Info	Provides additional call information for modem calls. This attribute is generated in start and stop accounting records.
78	Configuration-Token	Indicates a type of user profile to be used. This attribute should be used in large distributed authentication networks based on proxy. It is sent from a RADIUS Proxy Server to a RADIUS Proxy Client in an Access-Accept; it should not be sent to a NAS.
79	EAP-Message	Encapsulates Extended Access Protocol (EAP) packets that allow the NAS to authenticate dial-in users via EAP without having to understand the EAP protocol.
80	Message-Authenticator	Prevents spoofing Access-Requests using CHAP, ARAP, or EAP authentication methods.
81	Tunnel-Private-Group-ID	Indicates the group ID for a particular tunneled session.
82	Tunnel-Assignment-ID[1]	Indicates to the tunnel initiator the particular tunnel to which a session is assigned.
83	Tunnel-Preference	Indicates the relative preference assigned to each tunnel. This attribute should be included if more than one set of tunneling attributes is returned by the RADIUS server to the tunnel initiator.
84	ARAP-Challenge-Response	Contains the response to the challenge of the dial-in client.
85	Acct-Interim-Interval	Indicates the number of seconds between each interim update in seconds for this specific session. This value can only appear in the Access-Accept message.
86	Acct-Tunnel-Packets-Lost	Indicates the number of packets lost on a given link. This attribute should be included in Accounting-Request packets that contain an Acct-Status-Type attribute having the value Tunnel-Link-Stop.
87	NAS-Port-ID	Contains a text string which identifies the port of the NAS that is authenticating the user.
88	Framed-Pool	Contains the name of an assigned address pool that should be used to assign an address for the user. If a NAS does not support multiple address pools, the NAS should ignore this attribute.
90	Tunnel-Client-Auth-ID	Specifies the name used by the tunnel initiator (also known as the NAS) when authenticating tunnel setup with the tunnel terminator. Supports L2F and L2TP protocols.
91	Tunnel-Server-Auth-ID	Specifies the name used by the tunnel terminator (also known as the Home Gateway) when authenticating tunnel setup with the tunnel initiator. Supports L2F and L2TP protocols.
200	IETF-Token-Immediate	Determines how RADIUS treats passwords received from login-users when their file entry specifies a hand-held security card server. The value for this attribute is indicated by a numeric value as follows: • 0: No, meaning that the password is ignored. • 1: Yes, meaning that the password is used for authentication.

1. This RADIUS attribute complies with the following two IETF documents: RFC 2868, *RADIUS Attributes for Tunnel Protocol Support* and RFC 2867, *RADIUS Accounting Modifications for Tunnel Protocol Support*.

RADIUS Vendor-Proprietary Attributes

Although an IETF draft standard for RADIUS specifies a method for communicating vendor-proprietary information between the network access server and the RADIUS server, some vendors have extended the RADIUS attribute set for specific applications.

This section contains the following sections:

- Supported Vendor-Proprietary RADIUS Attributes
- Comprehensive List of Vendor-Proprietary RADIUS Attribute Descriptions

Supported Vendor-Proprietary RADIUS Attributes

Table 73 lists Cisco-supported vendor-proprietary RADIUS attributes and the Cisco IOS release in which they are implemented. In cases where the attribute has a security server-specific format, the format is specified. Refer to Table 74 for a list of descriptions.

Note Attributes implemented in special (AA) or early development (T) releases will be added to the next mainline image.

Table 73 Supported Vendor-Proprietary RADIUS Attributes

Number	Vendor-Proprietary Attribute	11.1	11.2	11.3	11.3AA	11.3T	12.0	12.1	12.2
17	Change-Password	no	no	yes	yes	yes	yes	yes	yes
21	Password-Expiration	no	no	yes	yes	yes	yes	yes	yes
68	Tunnel-ID	no	no	no	no	no	no	no	yes
108	My-Endpoint-Disc-Alias	no	no	no	no	no	no	no	no
109	My-Name-Alias	no	no	no	no	no	no	no	no
110	Remote-FW	no	no	no	no	no	no	no	no
111	Multicast-GLeave-Delay	no	no	no	no	no	no	no	no
112	CBCP-Enable	no	no	no	no	no	no	no	no
113	CBCP-Mode	no	no	no	no	no	no	no	no
114	CBCP-Delay	no	no	no	no	no	no	no	no
115	CBCP-Trunk-Group	no	no	no	no	no	no	no	no

Table 73 Supported Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	11.1	11.2	11.3	11.3AA	11.3T	12.0	12.1	12.2
116	Appletalk-Route	no	no	no	no	no	no	no	no
117	Appletalk-Peer-Mode	no	no	no	no	no	no	no	no
118	Route-Appletalk	no	no	no	no	no	no	no	no
119	FCP-Parameter	no	no	no	no	no	no	no	no
120	Modem-PortNo	no	no	no	no	no	no	no	no
121	Modem-SlotNo	no	no	no	no	no	no	no	no
122	Modem-ShelfNo	no	no	no	no	no	no	no	no
123	Call-Attempt-Limit	no	no	no	no	no	no	no	no
124	Call-Block-Duration	no	no	no	no	no	no	no	no
125	Maximum-Call-Duration	no	no	no	no	no	no	no	no
126	Router-Preference	no	no	no	no	no	no	no	no
127	Tunneling-Protocol	no	no	no	no	no	no	no	no
128	Shared-Profile-Enable	no	no	no	no	no	no	no	no
129	Primary-Home-Agent	no	no	no	no	no	no	no	no
130	Secondary-Home-Agent	no	no	no	no	no	no	no	no
131	Dialout-Allowed	no	no	no	no	no	no	no	no
133	BACP-Enable	no	no	no	no	no	no	no	no
134	DHCP-Maximum-Leases	no	no	no	no	no	no	no	no
135	Primary-DNS-Server	no	no	no	no	yes	yes	yes	yes
136	Secondary-DNS-Server	no	no	no	no	yes	yes	yes	yes
137	Client-Assign-DNS	no	no	no	no	no	no	no	no
138	User-Acct-Type	no	no	no	no	no	no	no	no
139	User-Acct-Host	no	no	no	no	no	no	no	no
140	User-Acct-Port	no	no	no	no	no	no	no	no
141	User-Acct-Key	no	no	no	no	no	no	no	no
142	User-Acct-Base	no	no	no	no	no	no	no	no
143	User-Acct-Time	no	no	no	no	no	no	no	no
144	Assign-IP-Client	no	no	no	no	no	no	no	no
145	Assign-IP-Server	no	no	no	no	no	no	no	no
146	Assign-IP-Global-Pool	no	no	no	no	no	no	no	no
147	DHCP-Reply	no	no	no	no	no	no	no	no
148	DHCP-Pool-Number	no	no	no	no	no	no	no	no
149	Expect-Callback	no	no	no	no	no	no	no	no
150	Event-Type	no	no	no	no	no	no	no	no
151	Session-Svr-Key	no	no	no	yes	no	no	yes	yes

Table 73 Supported Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	11.1	11.2	11.3	11.3AA	11.3T	12.0	12.1	12.2
152	Multicast-Rate-Limit	no	no	no	yes	no	no	yes	yes
153	IF-Netmask	no	no	no	no	no	no	no	no
154	Remote-Addr	no	no	no	no	no	no	no	no
155	Multicast-Client	no	no	no	yes	no	no	yes	yes
156	FR-Circuit-Name	no	no	no	no	no	no	no	no
157	FR-LinkUp	no	no	no	no	no	no	no	no
158	FR-Nailed-Grp	no	no	no	no	no	no	no	no
159	FR-Type	no	no	no	no	no	no	no	no
160	FR-Link-Mgt	no	no	no	no	no	no	no	no
161	FR-N391	no	no	no	no	no	no	no	no
162	FR-DCE-N392	no	no	no	no	no	no	no	no
163	FR-DTE-N392	no	no	no	no	no	no	no	no
164	FR-DCE-N393	no	no	no	no	no	no	no	no
165	FR-DTE-N393	no	no	no	no	no	no	no	no
166	FR-T391	no	no	no	no	no	no	no	no
167	FR-T392	no	no	no	no	no	no	no	no
168	Bridge-Address	no	no	no	no	no	no	no	no
169	TS-Idle-Limit	no	no	no	no	no	no	no	no
170	TS-Idle-Mode	no	no	no	no	no	no	no	no
171	DBA-Monitor	no	no	no	no	no	no	no	no
172	Base-Channel-Count	no	no	no	no	no	no	no	no
173	Minimum-Channels	no	no	no	no	no	no	no	no
174	IPX-Route	no	no	no	no	no	no	no	no
175	FT1-Caller	no	no	no	no	no	no	no	no
176	Backup	no	no	no	no	no	no	no	no
177	Call-Type	no	no	no	no	no	no	no	no
178	Group	no	no	no	no	no	no	no	no
179	FR-DLCI	no	no	no	no	no	no	no	no
180	FR-Profile-Name	no	no	no	no	no	no	no	no
181	Ara-PW	no	no	no	no	no	no	no	no
182	IPX-Node-Addr	no	no	no	no	no	no	no	no
183	Home-Agent-IP-Addr	no	no	no	no	no	no	no	no
184	Home-Agent-Password	no	no	no	no	no	no	no	no
185	Home-Network-Name	no	no	no	no	no	no	no	no
186	Home-Agent-UDP-Port	no	no	no	no	no	no	no	no

Table 73 Supported Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	11.1	11.2	11.3	11.3AA	11.3T	12.0	12.1	12.2
187	Multilink-ID	no	no	no	yes	yes	yes	yes	yes
188	Num-In-Multilink	no	no	no	yes	yes	yes	yes	yes
189	First-Dest	no	no	no	no	no	no	no	no
190	Pre-Input-Octets	no	no	no	yes	yes	yes	yes	yes
191	Pre-Output-Octets	no	no	no	yes	yes	yes	yes	yes
192	Pre-Input-Packets	no	no	no	yes	yes	yes	yes	yes
193	Pre-Output-Packets	no	no	no	yes	yes	yes	yes	yes
194	Maximum-Time	no	no	yes	yes	yes	yes	yes	yes
195	Disconnect-Cause	no	no	yes	yes	yes	yes	yes	yes
196	Connect-Progress	no	no	no	no	no	no	yes	yes
197	Data-Rate	no	no	no	no	yes	yes	yes	yes
198	PreSession-Time	no	no	no	yes	yes	yes	yes	yes
199	Token-Idle	no	no	no	no	no	no	no	no
201	Require-Auth	no	no	no	no	no	no	no	no
202	Number-Sessions	no	no	no	no	no	no	no	no
203	Authen-Alias	no	no	no	no	no	no	no	no
204	Token-Expiry	no	no	no	no	no	no	no	no
205	Menu-Selector	no	no	no	no	no	no	no	no
206	Menu-Item	no	no	no	no	no	no	no	no
207	PW-Warntime	no	no	no	no	no	no	no	no
208	PW-Lifetime	no	no	yes	yes	yes	yes	yes	yes
209	IP-Direct	no	no	no	no	yes	yes	yes	yes
210	PPP-VJ-Slot-Comp	no	no	yes	yes	yes	yes	yes	yes
211	PPP-VJ-1172	no	no	no	no	no	no	no	no
212	PPP-Async-Map	no	no	no	no	no	no	no	no
213	Third-Prompt	no	no	no	no	no	no	no	no
214	Send-Secret	no	no	no	no	no	no	yes	yes
215	Receive-Secret	no	no	no	no	no	no	no	no
216	IPX-Peer-Mode	no	no	no	no	no	no	no	no
217	IP-Pool-Definition	no	no	yes	yes	yes	yes	yes	yes
218	Assign-IP-Pool	no	no	yes	yes	yes	yes	yes	yes
219	FR-Direct	no	no	no	no	no	no	no	no
220	FR-Direct-Profile	no	no	no	no	no	no	no	no
221	FR-Direct-DLCI	no	no	no	no	no	no	no	no
222	Handle-IPX	no	no	no	no	no	no	no	no

Table 73 Supported Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	11.1	11.2	11.3	11.3AA	11.3T	12.0	12.1	12.2
223	Netware-Timeout	no	no	no	no	no	no	no	no
224	IPX-Alias	no	no	no	no	no	no	no	no
225	Metric	no	no	no	no	no	no	no	no
226	PRI-Number-Type	no	no	no	no	no	no	no	no
227	Dial-Number	no	no	no	no	no	no	yes	yes
228	Route-IP	no	no	yes	yes	yes	yes	yes	yes
229	Route-IPX	no	no	no	no	no	no	no	no
230	Bridge	no	no	no	no	no	no	no	no
231	Send-Auth	no	no	no	no	no	no	yes	yes
232	Send-Passwd	no	no	no	no	no	no	no	no
233	Link-Compression	no	no	yes	yes	yes	yes	yes	yes
234	Target-Util	no	no	no	yes	no	yes	yes	yes
235	Maximum-Channels	no	no	yes	yes	yes	yes	yes	yes
236	Inc-Channel-Count	no	no	no	no	no	no	no	no
237	Dec-Channel-Count	no	no	no	no	no	no	no	no
238	Seconds-of-History	no	no	no	no	no	no	no	no
239	History-Weigh-Type	no	no	no	no	no	no	no	no
240	Add-Seconds	no	no	no	no	no	no	no	no
241	Remove-Seconds	no	no	no	no	no	no	no	no
242	Data-Filter	no	no	yes	yes	yes	yes	yes	yes
243	Call-Filter	no	no	no	no	no	no	no	no
244	Idle-Limit	no	no	yes	yes	yes	yes	yes	yes
245	Preempt-Limit	no	no	no	no	no	no	no	no
246	Callback	no	no	no	no	no	no	no	no
247	Data-Svc	no	no	no	no	no	no	yes	yes
248	Force-56	no	no	no	no	no	no	yes	yes
249	Billing Number	no	no	no	no	no	no	no	no
250	Call-By-Call	no	no	no	no	no	no	no	no
251	Transit-Number	no	no	no	no	no	no	no	no
252	Host-Info	no	no	no	no	no	no	no	no
253	PPP-Address	no	no	no	no	no	no	no	no
254	MPP-Idle-Percent	no	no	no	no	no	no	no	no
255	Xmit-Rate	no	no	no	yes	yes	yes	yes	yes

Comprehensive List of Vendor-Proprietary RADIUS Attribute Descriptions

Table 74 lists and describes the known vendor-proprietary RADIUS attributes:

Table 74 Vendor-Proprietary RADIUS Attributes

Number	Vendor-Proprietary Attribute	Description
17	Change-Password	Specifies a request to change the password of a user.
21	Password-Expiration	Specifies an expiration date for a user's password in the user's file entry.
68	Tunnel-ID	(Ascend 5) Specifies the string assigned by RADIUS for each session using CLID or DNIS tunneling. When accounting is implemented, this value is used for accoutning.
108	My-Endpoint-Disc-Alias	(Ascend 5) No description available.
109	My-Name-Alias	(Ascend 5) No description available.
110	Remote-FW	(Ascend 5) No description available.
111	Multicast-GLeave-Delay	(Ascend 5) No description available.
112	CBCP-Enable	(Ascend 5) No description available.
113	CBCP-Mode	(Ascend 5) No description available.
114	CBCP-Delay	(Ascend 5) No description available.
115	CBCP-Trunk-Group	(Ascend 5) No description available.
116	Appletalk-Route	(Ascend 5) No description available.
117	Appletalk-Peer-Mode	(Ascend 5) No description available.
118	Route-Appletalk	(Ascend 5) No description available.
119	FCP-Parameter	(Ascend 5) No description available.
120	Modem-PortNo	(Ascend 5) No description available.
121	Modem-SlotNo	(Ascend 5) No description available.
122	Modem-ShelfNo	(Ascend 5) No description available.
123	Call-Attempt-Limit	(Ascend 5) No description available.
124	Call-Block-Duration	(Ascend 5) No description available.
125	Maximum-Call-Duration	(Ascend 5) No description available.
126	Router-Preference	(Ascend 5) No description available.
127	Tunneling-Protocol	(Ascend 5) No description available.
128	Shared-Profile-Enable	(Ascend 5) No description available.
129	Primary-Home-Agent	(Ascend 5) No description available.
130	Secondary-Home-Agent	(Ascend 5) No description available.
131	Dialout-Allowed	(Ascend 5) No description available.
133	BACP-Enable	(Ascend 5) No description available.
134	DHCP-Maximum-Leases	(Ascend 5) No description available.

Table 74 Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	Description
135	Primary-DNS-Server	Identifies a primary DNS server that can be requested by Microsoft PPP clients from the network access server during IPCP negotiation.
136	Secondary-DNS-Server	Identifies a secondary DNS server that can be requested by Microsoft PPP clients from the network access server during IPCP negotiation.
137	Client-Assign-DNS	No description available.
138	User-Acct-Type	No description available.
139	User-Acct-Host	No description available.
140	User-Acct-Port	No description available.
141	User-Acct-Key	No description available.
142	User-Acct-Base	No description available.
143	User-Acct-Time	No description available.
144	Assign-IP-Client	No description available.
145	Assign-IP-Server	No description available.
146	Assign-IP-Global-Pool	No description available.
147	DHCP-Reply	No description available.
148	DHCP-Pool-Number	No description available.
149	Expect-Callback	No description available.
150	Event-Type	No description available.
151	Session-Svr-Key	No description available.
152	Multicast-Rate-Limit	No description available.
153	IF-Netmask	No description available.
154	Remote-Addr	No description available.
155	Multicast-Client	No description available.
156	FR-Circuit-Name	No description available.
157	FR-LinkUp	No description available.
158	FR-Nailed-Grp	No description available.
159	FR-Type	No description available.
160	FR-Link-Mgt	No description available.
161	FR-N391	No description available.
162	FR-DCE-N392	No description available.
163	FR-DTE-N392	No description available.
164	FR-DCE-N393	No description available.
165	FR-DTE-N393	No description available.
166	FR-T391	No description available.
167	FR-T392	No description available.

Table 74 Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	Description
168	Bridge-Address	No description available.
169	TS-Idle-Limit	No description available.
170	TS-Idle-Mode	No description available.
171	DBA-Monitor	No description available.
172	Base-Channel-Count	No description available.
173	Minimum-Channels	No description available.
174	IPX-Route	No description available.
175	FT1-Caller	No description available.
176	Backup	No description available.
177	Call-Type	No description available.
178	Group	No description available.
179	FR-DLCI	No description available.
180	FR-Profile-Name	No description available.
181	Ara-PW	No description available.
182	IPX-Node-Addr	No description available.
183	Home-Agent-IP-Addr	Indicates the home agent's IP address (in dotted decimal format) when using Ascend Tunnel Management Protocol (ATMP).
184	Home-Agent-Password	With ATMP, specifies the password that the foreign agent uses to authenticate itself.
185	Home-Network-Name	With ATMP, indicates the name of the connection profile to which the home agent sends all packets.
186	Home-Agent-UDP-Port	Indicates the UDP port number the foreign agent uses to send ATMP messages to the home agent.
187	Multilink-ID	Reports the identification number of the multilink bundle when the session closes. This attribute applies to sessions that are part of a multilink bundle. The Multilink-ID attribute is sent in authentication-response packets.
188	Num-In-Multilink	Reports the number of sessions remaining in a multilink bundle when the session reported in an accounting-stop packet closes. This attribute applies to sessions that are part of a multilink bundle. The Num-In-Multilink attribute is sent in authentication-response packets and in some accounting-request packets.
189	First-Dest	Records the destination IP address of the first packet received after authentication.
190	Pre-Input-Octets	Records the number of input octets before authentication. The Pre-Input-Octets attribute is sent in accounting-stop records.
191	Pre-Output-Octets	Records the number of output octets before authentication. The Pre-Output-Octets attribute is sent in accounting-stop records.

Table 74 Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	Description
192	Pre-Input-Packets	Records the number of input packets before authentication. The Pre-Input-Packets attribute is sent in accounting-stop records.
193	Pre-Output-Packets	Records the number of output packets before authentication. The Pre-Output-Packets attribute is sent in accounting-stop records.
194	Maximum-Time	Specifies the maximum length of time (in seconds) allowed for any session. After the session reaches the time limit, its connection is dropped.
195	Disconnect-Cause	Specifies the reason a connection was taken offline. The Disconnect-Cause attribute is sent in accounting-stop records. This attribute also causes stop records to be generated without first generating start records if disconnection occurs before authentication is performed. For more information, refer to the table of Disconnect-Cause Attribute Values and their meanings.
196	Connect-Progress	Indicates the connection state before the connection is disconnected.
197	Data-Rate	Specifies the average number of bits per second over the course of the connection's lifetime. The Data-Rate attribute is sent in accounting-stop records.
198	PreSession-Time	Specifies the length of time, in seconds, from when a call first connects to when it completes authentication. The PreSession-Time attribute is sent in accounting-stop records.
199	Token-Idle	Indicates the maximum amount of time (in minutes) a cached token can remain alive between authentications.
201	Require-Auth	Defines whether additional authentication is required for class that has been CLID authenticated.
202	Number-Sessions	Specifies the number of active sessions (per class) reported to the RADIUS accounting server.
203	Authen-Alias	Defines the RADIUS server's login name during PPP authentication.
204	Token-Expiry	Defines the lifetime of a cached token.
205	Menu-Selector	Defines a string to be used to cue a user to input data.
206	Menu-Item	Specifies a single menu-item for a user-profile. Up to 20 menu items can be assigned per profile.
207	PW-Warntime	(Ascend 5) No description available.
208	PW-Lifetime	Enables you to specify on a per-user basis the number of days that a password is valid.

Table 74 Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	Description
209	IP-Direct	When you include this attribute in a user's file entry, a framed route is installed to the routing and bridging tables. **Note** Packet routing is dependent upon the entire table, not just this newly installed entry. The inclusion of this attribute does not guarantee that all packets should be sent to the specified IP address; thus, this attribute is not fully supported. These attribute limitations occur because the Cisco router cannot bypass all internal routing and bridging tables and send packets to a specified IP address.
210	PPP-VJ-Slot-Comp	Instructs the Cisco router not to use slot compression when sending VJ-compressed packets over a PPP link.
211	PPP-VJ-1172	Instructs PPP to use the 0x0037 value for VJ compression.
212	PPP-Async-Map	Gives the Cisco router the asynchronous control character map for the PPP session. The specified control characters are passed through the PPP link as data and used by applications running over the link.
213	Third-Prompt	Defines a third prompt (after username and password) for additional user input.
214	Send-Secret	Enables an encrypted password to be used in place of a regular password in outdial profiles.
215	Receive-Secret	Enables an encrypted password to be verified by the RADIUS server.
216	IPX-Peer-Mode	(Ascend 5) No description available.
217	IP-Pool-Definition	Defines a pool of addresses using the following format: X a.b.c Z; where X is the pool index number, a.b.c is the pool's starting IP address, and Z is the number of IP addresses in the pool. For example, 3 10.0.0.1 5 allocates 10.0.0.1 through 10.0.0.5 for dynamic assignment.
218	Assign-IP-Pool	Tells the router to assign the user and IP address from the IP pool.
219	FR-Direct	Defines whether the connection profile operates in Frame Relay redirect mode.
220	FR-Direct-Profile	Defines the name of the Frame Relay profile carrying this connection to the Frame Relay switch.
221	FR-Direct-DLCI	Indicates the DLCI carrying this connection to the Frame Relay switch.
222	Handle-IPX	Indicates how NCP watchdog requests will be handled.
223	Netware-Timeout	Defines, in minutes, how long the RADIUS server responds to NCP watchdog packets.
224	IPX-Alias	Allows you to define an alias for IPX routers requiring numbered interfaces.

Table 74 Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	Description
225	Metric	No description available.
226	PRI-Number-Type	No description available.
227	Dial-Number	Defines the number to dial.
228	Route-IP	Indicates whether IP routing is allowed for the user's file entry.
229	Route-IPX	Allows you to enable IPX routing.
230	Bridge	No description available.
231	Send-Auth	Defines the protocol to use (PAP or CHAP) for username-password authentication following CLID authentication.
232	Send-Passwd	Enables the RADIUS server to specify the password that is sent to the remote end of a connection on outgoing calls.
233	Link-Compression	Defines whether to turn on or turn off "stac" compression over a PPP link. Link compression is defined as a numeric value as follows: • 0: None • 1: Stac • 2: Stac-Draft-9 • 3: MS-Stac
234	Target-Util	Specifies the load-threshold percentage value for bringing up an additional channel when PPP multilink is defined.
235	Maximum-Channels	Specifies allowed/allocatable maximum number of channels.
236	Inc-Channel-Count	No description available.
237	Dec-Channel-Count	No description available.
238	Seconds-of-History	No description available.
239	History-Weigh-Type	No description available.
240	Add-Seconds	No description available.
241	Remove-Seconds	No description available.
242	Data-Filter	Defines per-user IP data filters. These filters are retrieved only when a call is placed using a RADIUS outgoing profile or answered using a RADIUS incoming profile. Filter entries are applied on a first-match basis; therefore, the order in which filter entries are entered is important.
243	Call-Filter	Defines per-user IP data filters. On a Cisco router, this attribute is identical to the Data-Filter attribute.
244	Idle-Limit	Specifies the maximum time (in seconds) that any session can be idle. When the session reaches the idle time limit, its connection is dropped.
245	Preempt-Limit	No description available.

Table 74 Vendor-Proprietary RADIUS Attributes (continued)

Number	Vendor-Proprietary Attribute	Description
246	Callback	Allows you to enable or disable callback.
247	Data-Svc	No description available.
248	Force-56	Determines whether the network access server uses only the 56 K portion of a channel, even when all 64 K appear to be available.
249	Billing Number	No description available.
250	Call-By-Call	No description available.
251	Transit-Number	No description available.
252	Host-Info	No description available.
253	PPP-Address	Indicates the IP address reported to the calling unit during PPP IPCP negotiations.
254	MPP-Idle-Percent	No description available.
255	Xmit-Rate	(Ascend 5) No description available.

For more information on vendor-propritary RADIUS attributes, refer to the section "Configuring Router for Vendor-Proprietary RADIUS Server Communication"in the chapter "Configuring RADIUS."

RADIUS Vendor-Specific Attributes and RADIUS Disconnect-Cause Attribute Values

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating vendor-specific information between the network access server and the RADIUS server by using the vendor-specific attribute (attribute 26). Attribute 26 encapsulates vendor specific attributes, thereby, allowing vendors to support their own extended attributes otherwise not suitable for general use.

The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. Cisco's vendor-ID is 9, and the supported option has vendor-type 1, which is named "cisco-avpair." The value is a string of the following format:

```
protocol : attribute sep value *
```

"Protocol" is a value of the Cisco "protocol" attribute for a particular type of authorization; protocols that can be used include IP, IPX, VPDN, VOIP, SHELL, RSVP, SIP, AIRNET, OUTBOUND. "Attribute" and "value" are an appropriate attribute-value (AV) pair defined in the Cisco TACACS+ specification, and "sep" is "=" for mandatory attributes and "*" for optional attributes. This allows the full set of features available for TACACS+ authorization to also be used for RADIUS.

For example, the following AV pair causes Cisco's "multiple named ip address pools" feature to be activated during IP authorization (during PPP's IPCP address assignment):

```
cisco-avpair= ″ip:addr-pool=first″
```

If you insert an "*", the AV pair "ip:addr-pool=first" becomes optional. Note that any AV pair can be made optional.

```
cisco-avpair= ″ip:addr-pool*first″
```

The following example shows how to cause a user logging in from a network access server to have immediate access to EXEC commands:

```
cisco-avpair= ″shell:priv-lvl=15″
```

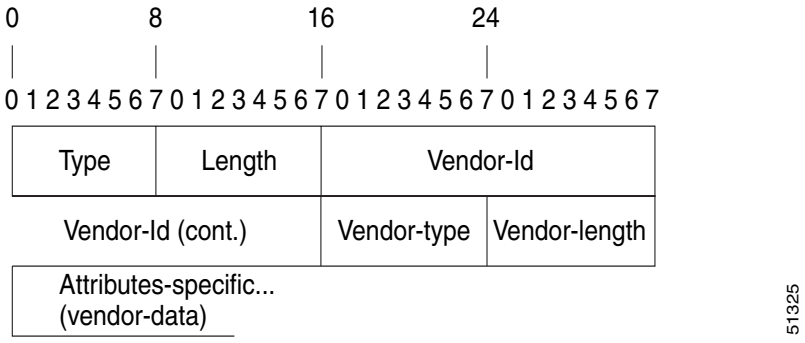
Attribute 26 contains the following three elements:

- Type
- Length
- String (also known as data)
  - Vendor-Id
  - Vendor-Type

- Vendor-Length
- Vendor-Data

Figure 126 shows the packet format for a VSA encapsulated "behind" attribute 26.

Figure 126 VSA Encapsulated Behind Attribute 26



Note It is up to the vendor to specify the format of their VSA. The Attribute-Specific field (also known as Vendor-Data) is dependent on the vendor's definition of that attribute.

Table 76 lists supported vendor-specific RADIUS attributes (IETF attribute 26). Table 75 describes significant fields listed in the Table 76.

Table 75 Vendor-Specific Attributes Table Field Descriptions

Field	Description
Number	All attributes listed in the following table are extensions of IETF attribute 26.
Vendor-Specific Command Codes	A defined code used to identify a particular vendor. Code 9 defines Cisco VSAs, 311 defines Microsoft VSAs, and 529 defines Ascend VSAs.
Sub-Type Number	The attribute ID number. This number is much like the ID numbers of IETF attributes, except it is a "second layer" ID number encapsulated behind attribute 26.
Attribute	The ASCII string name of the attribute.
Description	Description of the attribute.

Table 76 Vendor-Specific RADIUS IETF Attributes

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
MS-CHAP Attributes				
26	311	1	MSCHAP-Response	Contains the response value provided by a PPP MS-CHAP user in response to the challenge. It is only used in Access-Request packets. This attribute is identical to the PPP CHAP Identifier. (RFC 2548)

Table 76 Vendor-Specific RADIUS IETF Attributes (continued)

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	311	11	MSCHAP-Challenge	Contains the challenge sent by a network access server to an MS-CHAP user. It can be used in both Access-Request and Access-Challenge packets. (RFC 2548)
VPDN Attributes				
26	9	1	l2tp-cm-local-window-size	Specifies the maximum receive window size for L2TP control messages. This value is advertised to the peer during tunnel establishment.
26	9	1	l2tp-drop-out-of-order	Respects sequence numbers on data packets by dropping those that are received out of order. This does not ensure that sequence numbers will be sent on data packets, just how to handle them if they are received.
26	9	1	l2tp-hello-interval	Specifies the number of seconds for the hello keepalive interval. Hello packets are sent when no data has been sent on a tunnel for the number of seconds configured here.
26	9	1	l2tp-hidden-avp	When enabled, sensitive AVPs in L2TP control messages are scrambled or hidden.
26	9	1	l2tp-nosession-timeout	Specifies the number of seconds that a tunnel will stay active with no sessions before timing out and shutting down.
26	9	1	l2tp-tos-reflect	Copies the IP ToS field from the IP header of each payload packet to the IP header of the tunnel packet for packets entering the tunnel at the LNS.
26	9	1	l2tp-tunnel-authen	If this attribute is set, it performs L2TP tunnel authentication.
26	9	1	l2tp-tunnel-password	Shared secret used for L2TP tunnel authentication and AVP hiding.
26	9	1	l2tp-udp-checksum	This is an authorization attribute and defines whether L2TP should perform UDP checksums for data packets. Valid values are “yes” and “no.” The default is no.
Store and Forward Fax Attributes				
26	9	3	Fax-Account-Id-Origin	Indicates the account ID origin as defined by system administrator for the **mmoip aaa receive-id** or the **mmoip aaa send-id** commands.
26	9	4	Fax-Msg-Id=	Indicates a unique fax message identification number assigned by Store and Forward Fax.
26	9	5	Fax-Pages	Indicates the number of pages transmitted or received during this fax session. This page count includes cover pages.

Table 76 Vendor-Specific RADIUS IETF Attributes (continued)

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	6	Fax-Coverpage-Flag	Indicates whether or not a cover page was generated by the off-ramp gateway for this fax session. True indicates that a cover page was generated; false means that a cover page was not generated.
26	9	7	Fax-Modem-Time	Indicates the amount of time in seconds the modem sent fax data (x) and the amount of time in seconds of the total fax session (y), which includes both fax-mail and PSTN time, in the form x/y. For example, 10/15 means that the transfer time took 10 seconds, and the total fax session took 15 seconds.
26	9	8	Fax-Connect-Speed	Indicates the modem speed at which this fax-mail was initially transmitted or received. Possible values are 1200, 4800, 9600, and 14400.
26	9	9	Fax-Recipient-Count	Indicates the number of recipients for this fax transmission. Until e-mail servers support Session mode, the number should be 1.
26	9	10	Fax-Process-Abort-Flag	Indicates that the fax session was aborted or successful. True means that the session was aborted; false means that the session was successful.
26	9	11	Fax-Dsn-Address	Indicates the address to which DSNs will be sent.
26	9	12	Fax-Dsn-Flag	Indicates whether or not DSN has been enabled. True indicates that DSN has been enabled; false means that DSN has not been enabled.
26	9	13	Fax-Mdn-Address	Indicates the address to which MDNs will be sent.
26	9	14	Fax-Mdn-Flag	Indicates whether or not message delivery notification (MDN) has been enabled. True indicates that MDN had been enabled; false means that MDN had not been enabled.
26	9	15	Fax-Auth-Status	Indicates whether or not authentication for this fax session was successful. Possible values for this field are success, failed, bypassed, or unknown.
26	9	16	Email-Server-Address	Indicates the IP address of the e-mail server handling the on-ramp fax-mail message.
26	9	17	Email-Server-Ack-Flag	Indicates that the on-ramp gateway has received a positive acknowledgment from the e-mail server accepting the fax-mail message.
26	9	18	Gateway-Id	Indicates the name of the gateway that processed the fax session. The name appears in the following format: hostname.domain-name.
26	9	19	Call-Type	Describes the type of fax activity: fax receive or fax send.
26	9	20	Port-Used	Indicates the slot/port number of the Cisco AS5300 used to either transmit or receive this fax-mail.

Table 76 Vendor-Specific RADIUS IETF Attributes (continued)

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	21	Abort-Cause	If the fax session aborts, indicates the system component that signaled the abort. Examples of system components that could trigger an abort are FAP (Fax Application Process), TIFF (the TIFF reader or the TIFF writer), fax-mail client, fax-mail server, ESMTP client, or ESMTP server.
H323 Attributes				
26	9	23	Remote-Gateway-ID (h323-remote-address)	Indicates the IP address of the remote gateway.
26	9	24	Connection-ID (h323-conf-id)	Identifies the conference ID.
26	9	25	Setup-Time (h323-setup-time)	Indicates the setup time for this connection in Coordinated Universal Time (UTC) formerly known as Greenwich Mean Time (GMT) and Zulu time.
26	9	26	Call-Origin (h323-call-origin)	Indicates the origin of the call relative to the gateway. Possible values are originating and terminating (answer).
26	9	27	Call-Type (h323-call-type)	Indicates call leg type. Possible values are **telephony** and **VoIP**.
26	9	28	Connect-Time (h323-connect-time)	Indicates the connection time for this call leg in UTC.
26	9	29	Disconnect-Time (h323-disconnect-time)	Indicates the time this call leg was disconnected in UTC.
26	9	30	Disconnect-Cause (h323-disconnect-caus)e	Specifies the reason a connection was taken offline per Q.931 specification.
26	9	31	Voice-Quality (h323-voice-quality)	Specifies the impairment factor (ICPIF) affecting voice quality for a call.
26	9	33	Gateway-ID (h323-gw-id)	Indicates the name of the underlying gateway.
Large Scale Dialout Attributes				
26	9	1	callback-dialstring	Defines a dialing string to be used for callback.
26	9	1	data-service	No description available.
26	9	1	dial-number	Defines the number to dial.
26	9	1	force-56	Determines whether the network access server uses only the 56 K portion of a channel, even when all 64 K appear to be available.
26	9	1	map-class	Allows the user profile to reference information configured in a map class of the same name on the network access server that dials out.

Table 76 Vendor-Specific RADIUS IETF Attributes (continued)

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	1	send-auth	Defines the protocol to use (PAP or CHAP) for username-password authentication following CLID authentication.
26	9	1	send-name	PPP name authentication. To apply for PAP, do not configure the **ppp pap sent-name password** command on the interface. For PAP, "preauth:send-name" and "preauth:send-secret" will be used as the PAP username and PAP password for outbound authentication. For CHAP, "preauth:send-name" will be used not only for outbound authentication, but also for inbound authentication. For a CHAP inbound case, the NAS will use the name defined in "preauth:send-name" in the challenge packet to the caller box. **Note** The send-name attribute has changed over time: Initially, it performed the functions now provided by both the send-name and remote-name attributes. Because the remote-name attribute has been added, the send-name attribute is restricted to its current behavior.
26	9	1	send-secret	PPP password authentication. The vendor-specific attributes (VSAs) "preauth:send-name" and "preauth:send-secret" will be used as the PAP username and PAP password for outbound authentication. For a CHAP outbound case, both "preauth:send-name" and "preauth:send-secret" will be used in the response packet.
26	9	1	remote-name	Provides the name of the remote host for use in large-scale dial-out. Dialer checks that the large-scale dial-out remote name matches the authenticated name, to protect against accidental user RADIUS misconfiguration. (For example, dialing a valid phone number but connecting to the wrong router.)
Miscellaneous Attributes				
26	9	2	Cisco-NAS-Port	Specifies additional vendor specific attribute (VSA) information for NAS-Port accounting. To specify additional NAS-Port information in the form an Attribute-Value Pair (AVPair) string, use the **radius-server vsa send** global configuration command. **Note** This VSA is typically used in Accounting, but may also be used in Authentication (Access-Request) packets.
26	9	1	min-links	Sets the minimum number of links for MLP.

Table 76 Vendor-Specific RADIUS IETF Attributes (continued)

Number	Vendor-Specific Company Code	Sub-Type Number	Attribute	Description
26	9	1	proxyacl#<n>	Allows users to configure the downloadable user profiles (dynamic ACLs) by using the authentication proxy feature so that users can have the configured authorization to permit traffic going through the configured interfaces.
26	9	1	spi	Carries the authentication information needed by the home agent to authenticate a mobile node during registration. The information is in the same syntax as the **ip mobile secure host <addr>** configuration command. Basically it contains the rest of the configuration command that follows that string, verbatim. It provides the Security Parameter Index (SPI), key, authentication algorithm, authentication mode, and replay protection timestamp range.

For more information on configuring your NAS to recognize and use VSAs, refer to the section "Configuring Router to Use Vendor-Specific RADIUS Attributes"of the chapter "Configuring RADIUS."

RADIUS Disconnect-Cause Attribute Values

Disconnect-cause attribute values specify the reason a connection was taken offline. The attribute values are sent in Accounting request packets. These values are sent at the end of a session, even if the session fails to be authenticated. If the session is not authenticated, the attribute can cause stop records to be generated without first generating start records.

Table 77 lists the cause codes, values, and descriptions for the Disconnect-Cause (195) attribute.

Note The Disconnect-Cause is incremented by 1000 when it is used in RADIUS AVPairs; for example, disc-cause 4 becomes 1004.

Table 77 Disconnect-Cause Attribute Values

Cause Code	Value	Description
0	No-Reason	No reason is given for the disconnect.
1	No-Disconnect	The event was not disconnected.
2	Unknown	Reason unknown.
3	Call-Disconnect	The call has been disconnected.
4	CLID-Authentication-Failure	Failure to authenticate number of the calling-party.
9	No-Modem-Available	A modem in not available to connect the call.

Table 77 Disconnect-Cause Attribute Values (continued)

Cause Code	Value	Description
10	No-Carrier	No carrier detected. **Note** Codes 10, 11, and 12 can be sent if there is a disconnection during initial modem connection.
11	Lost-Carrier	Loss of carrier.
12	No-Detected-Result-Codes	Failure to detect modem result codes.
20	User-Ends-Session	User terminates a session. **Note** Codes 20, 22, 23, 24, 25, 26, 27, and 28 apply to EXEC sessions.
21	Idle-Timeout	Timeout waiting for user input. Codes 21, 100, 101, 102, and 120 apply to all session types.
22	Exit-Telnet-Session	Disconnect due to exiting Telnet session.
23	No-Remote-IP-Addr	Could not switch to SLIP/PPP; the remote end has no IP address.
24	Exit-Raw-TCP	Disconnect due to exiting raw TCP.
25	Password-Fail	Bad passwords.
26	Raw-TCP-Disabled	Raw TCP disabled.
27	Control-C-Detected	Control-C detected.
28	EXEC-Process-Destroyed	EXEC process destroyed.
29	Close-Virtual-Connection	User closes a virtual connection.
30	End-Virtual-Connection	Virual connected has ended.
31	Exit-Rlogin	User exists Rlogin.
32	Invalid-Rlogin-Option	Invalid Rlogin option selected.
33	Insufficient-Resources	Insufficient resources.
40	Timeout-PPP-LCP	PPP LCP negotiation timed out. **Note** Codes 40 through 49 apply to PPP sessions.
41	Failed-PPP-LCP-Negotiation	PPP LCP negotiation failed.
42	Failed-PPP-PAP-Auth-Fail	PPP PAP authentication failed.
43	Failed-PPP-CHAP-Auth	PPP CHAP authentication failed.
44	Failed-PPP-Remote-Auth	PPP remote authentication failed.
45	PPP-Remote-Terminate	PPP received a Terminate Request from remote end.
46	PPP-Closed-Event	Upper layer requested that the session be closed.
47	NCP-Closed-PPP	PPP session closed because there were no NCPs open.
48	MP-Error-PPP	PPP session closed because of an MP error.
49	PPP-Maximum-Channels	PPP session closed because maximum channels were reached.
50	Tables-Full	Disconnect due to full terminal server tables.
51	Resources-Full	Disconnect due to full internal resources.
52	Invalid-IP-Address	IP address is not valid for Telnet host.
53	Bad-Hostname	Hostname cannot be validated.

Table 77 Disconnect-Cause Attribute Values (continued)

Cause Code	Value	Description
54	Bad-Port	Port number is invalid or missing.
60	Reset-TCP	TCP connection has been reset. **Note** Codes 60 through 67 apply to Telnet or raw TCP sessions.
61	TCP-Connection-Refused	TCP connection has been refused by the host.
62	Timeout-TCP	TCP connection has timed out.
63	Foreign-Host-Close-TCP	TCP connection has been closed.
64	TCP-Network-Unreachable	TCP network is unreachable.
65	TCP-Host-Unreachable	TCP host is unreachable.
66	TCP-Network-Admin Unreachable	TCP network is unreachable for administrative reasons.
67	TCP-Port-Unreachable	TCP port in unreachable.
100	Session-Timeout	Session timed out.
101	Session-Failed-Security	Session failed for security reasons.
102	Session-End-Callback	Session terminated due to callback.
120	Invalid-Protocol	Call refused because the detected protocol is disabled.
150	RADIUS-Disconnect	Disconnected by RADIUS request.
151	Local-Admin-Disconnect	Administrative disconnect.
152	SNMP-Disconnect	Disconnected by SNMP request.
160	V110-Retries	Allowed V.110 retries have been exceeded.
170	PPP-Authentication-Timeout	PPP authentication timed out.
180	Local-Hangup	Disconnected by local hangup.
185	Remote-Hangup	Disconnected by remote end hangup.
190	T1-Quiesced	Disconnected because T1 line was quiesced.
195	Call-Duration	Disconnected because the maximum duration of the call was exceeded.
600	VPN-User-Disconnect	Call disconnected by client (through PPP). Code is sent if the LNS receives a PPP terminate request from the client.
601	VPN-Carrier-Loss	Loss of carrier. This can be the result of a physical line going dead. Code is sent when a client is unable to dial out using a dialer.
602	VPN-No-Resources	No resources available to handle the call. Code is sent when the client is unable to allocate memory (running low on memory).
603	VPN-Bad-Control-Packet	Bad L2TP or L2F control packets. This code is sent when an invalid control packet, such as missing mandatory Attribute-Value pairs (AVP), from the peer is received. When using L2TP, the code will be sent after six retransmits; when using L2F, the number of retransmits is user configurable. **Note** VPN-Tunnel-Shut will be sent if there are active sessions in the tunnel.

Table 77 Disconnect-Cause Attribute Values (continued)

Cause Code	Value	Description
604	VPN-Admin-Disconnect	Administrative disconnect. This can be the result of a VPN soft shutdown, which is when a client reaches maximum session limit or exceeds maximum hopcount. Code is sent when a tunnel is brought down by issuing the **clear vpdn tunnel** command.
605	VPN-Tunnel-Shut	Tunnel teardown or tunnel setup has failed. Code is sent when there are active sessions in a tunnel and the tunnel goes down. **Note** This code is *not* sent when tunnel authentication fails.
606	VPN-Local-Disconnect	Call is disconnected by LNS PPP module. Code is sent when the LNS sends a PPP terminate request to the client. It indicates a normal PPP disconnection initiated by the LNS.
607	VPN-Session-Limit	VPN soft shutdown is enabled. Code is sent when a call has been refused due to any of the soft shutdown restrictions previously mentioned.
608	VPN-Call-Redirect	VPN call redirect is enabled.

For Q.850 cause codes and descriptions, see the section “Internal Cause Codes for SIP and H.323” in the chapter “Cause Codes and Debug Values” of the *Cisco IOS Voice Troubleshooting and Monitoring*.

Connect-Info RADIUS Attribute 77

Feature History

Release	Modification
12.2(11)T	This feature was introduced.

This feature module describes the Connect-Info RADIUS Attribute 77 feature for Cisco IOS Release 12.2(11)T. It includes the following sections:

- Feature Overview, page 1917
- Supported Platforms, page 1918
- Supported Standards, MIBs, and RFCs, page 1919
- Prerequisites, page 1919
- Configuration Tasks, page 1919
- Configuration Examples, page 1920
- Command Reference, page 1920

Feature Overview

The Connect-Info RADIUS Attribute 77 feature introduces support for RADIUS attribute 77 (Connect-Info), which provides information about connection speeds, modulation, and compression for modem dial-in connections via RADIUS accounting “start” and “stop” records.

When the network access server (NAS) sends attribute 77 in accounting “start” and “stop” records, you can measure—across the platform—the connect rates. That is, attribute 77 allows you to record “transmit” speed (the speed at which the NAS modem sends information) and “receive” speed (the speed at which the NAS receives information). These modem speeds for user sessions allow you to determine whether user modem connections renegotiate to lower speeds shortly into a session. If the transmit and receive speeds are different from each other, attribute 77 will report both speeds, which allows you to establish the modem connection speeds each customer gets from his or her session.

Benefits

The Connect-Info RADIUS Attribute 77 feature enables the NAS to report Connect-Info (attribute 77) in accounting "start" and "stop" records that are sent to the RADIUS client. "start" and "stop" records allow you to compare transmit and receive speeds and have a more realistic view of a user session. Comparing transmit and receive speeds is important because many modem speeds are often different at the end of the modem connection (after negotiation).

Related Documents

- "Modem and Dial Shelf Configuration and Management" chapter of the *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2
- *Cisco IOS Dial Technologies Command Reference*, Release 12.2
- *Cisco IOS Security Command Reference*, Release 12.2
- *Cisco IOS Security Configuration Guide*, Release 12.2

Supported Platforms

- Cisco AS5300 series
- Cisco AS5400 series
- Cisco AS5800 series
- Cisco AS5850 series

Determining Platform Support Through Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Feature Navigator. Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image.

To access Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

- RFC 2869, *RADIUS Extensions*

Prerequisites

Before the NAS can send attribute 77 in accounting "start" and "stop" records, you must perform the following tasks:

- Configure your NAS for authentication, authorization, and accounting (AAA) and to accept incoming modem calls.
- Enable AAA accounting by using the **aaa accounting network default start-stop group radius** command in global configuration mode.
- Change the modem poll timer by using the **modem link-info poll time** command in global configuration mode. (Changing the modem poll timer is required on all supported platforms *except* the Cisco AS5400).

Configuration Tasks

None

Verifying Attribute 77

To verify attribute 77 in your accounting "start" and "stop" records, use the **debug radius** privileged EXEC command. The following example shows that Connect-Info appears in the first and last accounting attributes:

```
Router# debug radius

RADIUS: code=Acct-Request id=04 len=0134
                authenticator=BE A2 F3 BD EE CE 89 C7 - 48 19 32 F5 79 84 94 D5
                T=Connect-Info[77]                L=17 V="31200/33600 V34+/LAPM"
                T=Acct-Status-Type[40]            L=06 V=Start               [1]
                ...

RADIUS: code=Acct-Request id=07 len=0226
                authenticator=06 AC 03 10 4A 84 44 A4 - 6F D9 68 AA B3 90 44 CB
                ...
```

```
T=Connect-Info[77]                L=1F V="33600 V34+/LAPM (31200/336"
T=Acct-Status-Type[40]            L=06 V=Stop                  [2]
...
```

Note If the modem negotiation speeds are different, the speeds are shown in a bracket format at the end of the call.

Configuration Examples

This section provides the following configuration example:

- Configure NAS for AAA and Incoming Modem Calls Example

Configure NAS for AAA and Incoming Modem Calls Example

The following example is a sample NAS configuration for AAA and incoming modem calls:

```
interface Serial0:15
  no ip address
  isdn switch-type primary-net5
  isdn incoming-voice modem
!
interface Async1
  ip address 7.0.0.10 255.0.0.0
  encapsulation ppp
  async default routing
  async mode interactive
  no peer default ip address
  ppp authentication chap
!
line 1
  modem InOu
  transport preferred none
  transport input all
  autoselect ppp
!
```

Command Reference

This feature uses no new or modified commands. To see the command pages for the commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

Encrypted Vendor-Specific Attributes

Feature History

Release	Modification
12.2(8)T	This feature was introduced.

This feature module describes the Encrypted Vendor-Specific Attributes feature for Cisco IOS Release 12.2(8)T. It includes the following sections:

- Feature Overview, page 1921
- Supported Platforms, page 1923
- Supported Standards, MIBs, and RFCs, page 1924
- Prerequisites, page 1925
- Configuration Tasks, page 1925
- Configuration Examples, page 1925
- Command Reference, page 1926

Feature Overview

The Encrypted Vendor-Specific Attributes feature introduces support for the following three types of string vendor-specific attributes (VSAs):

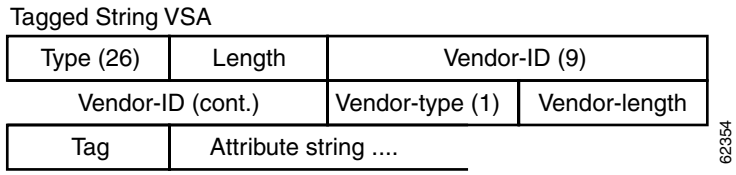
- Tagged String VSA (similar to Cisco VSA type 1 (Cisco:AVPair (1)) except that this new VSA is tagged)
- Encrypted String VSA (similar to Cisco VSA type 1 except that this new VSA is encrypted)
- Tagged and Encrypted String VSA (similar to Cisco VSA type 1 except that this new VSA is tagged and encrypted)

Cisco:AVPairs specify additional authentication and authorization information in the form an Attribute-Value Pair (AVPair) string. When Internet Engineering Task Force (IETF) RADIUS attribute 26 (Vendor-Specific) is transmitted with a vendor-Id number of “9” and a vendor-type value of “1” (which means that it is a Cisco AVPair), the RADIUS user profile format for a Cisco AVPair looks as follows: Cisco:AVPair = “protocol:attribute=value”.

Figure 127, Figure 128, and Figure 129 display the packet formats for each of the newly supported VSAs.

Tagged String VSA

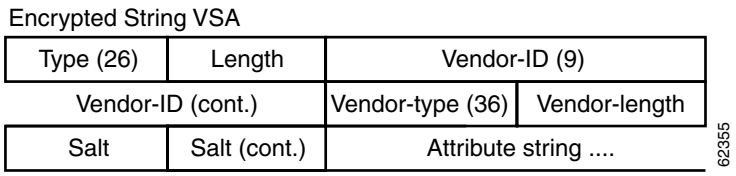
Figure 127 Tagged String VSA Format



To retrieve the correct value, the Tag field must be parsed correctly. The value for this field can range only from 0x01 through 0x1F. If the value is not within the specified range, the RADIUS server will ignore the value and consider the Tag field to be a part of the Attribute String field.

Encrypted String VSA

Figure 128 Encrypted String VSA Format

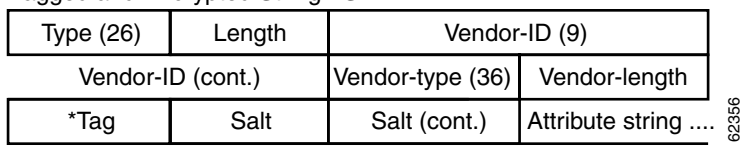


The Salt field ensures the uniqueness of the encryption key that is used to encrypt each instance of the VSA. The first and most significant bit of the Salt field must be set to 1.

Note Vendor-type (36) indicates that the attribute is an encrypted string VSA.

Tagged and Encrypted String VSA

Figure 129 Tagged and Encrypted String VSA Format



This VSA is similar to encrypted string VSAs *except* this VSA has an additional Tag field. If the Tag field is not within the valid range (0x01 0x01 through 0x1F), it is considered to be part of the Salt field.

Benefits

The Encrypted Vendor-Specific Attributes feature provides users with a way to centrally manage filters at a RADIUS server.

Related Documents

- The appendix "RADIUS Attributes" in the *Cisco IOS Security Configuration Guide*, Release 12.2
- RFC 2865, *Remote Authentication Dial In User Service (RADIUS)*
- RFC 2868, *RADIUS Attributes for Tunnel Protocol Support*

Supported Platforms

- Cisco 800 series
- Cisco 805
- Cisco 806
- Cisco 820
- Cisco 828
- Cisco 1400 series
- Cisco 1600 series
- Cisco 1710
- Cisco 1720
- Cisco 1721
- Cisco 1750
- Cisco 1751
- Cisco 2440
- Cisco 2600 series
- Cisco 2691
- Cisco 3620
- Cisco 3631
- Cisco 3640
- Cisco 3660
- Cisco 3725
- Cisco 3745
- Cisco 7100 series
- Cisco 7200 series
- Cisco 7500 series
- Cisco 7700 series

- Cisco uBR7200 series
- Universal Route Module

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

No new or modified RFCs are supported by this feature.

Prerequisites

Before the RADIUS server can accept tagged and encrypted VSAs, you must configure your server for AAA authentication and authorization and to accept PPP calls.

For information on performing these tasks, refer to the chapter "PPP Configuration" in the *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2 and the chapters "Configuring Authentication" and "Configuring Authorization" in the *Cisco IOS Security Configuration Guide*, Release 12.2.

Configuration Tasks

None

Verifying Encrypted VSAs

To verify that RADIUS-tagged and encrypted VSAs are being sent from the RADIUS server, use the following command in privileged EXEC mode:

Command	Purpose
`Router# debug radius`	Displays information associated with RADIUS. The output of this command shows whether tagged and encrypted VSAs are being sent from the RADIUS server.

Configuration Examples

This section provides the following configuration examples:

- NAS Configuration Example
- RADIUS User Profile with a Tagged and Encrypted VSA Example

NAS Configuration Example

The following example shows how to configure your network access server (NAS) with a basic configuration using tagged and encrypted VSAs. (This example assumes that the configuration required to make PPP calls is already enabled.)

```
aaa new-model
aaa authentication ppp default group radius
aaa authorization network default group radius
!
radius-server host 2.2.2.2 auth-port 1645 acct-port 1646
radius-server key cisco
```

RADIUS User Profile with a Tagged and Encrypted VSA Example

The following is an example of user profile on a RADIUS server that supports tagged and encrypted string VSAs:

```
mascot   Password = "cisco"
         Service-Type = NAS-Prompt,
         Framed-Protocol = PPP,
         Cisco:Cisco-Enc = "ip:route=4.0.0.0 255.0.0.0"
         Cisco.attr Cisco-Enc 36 tag-encstr(*,*)
```

Command Reference

This feature uses no new or modified commands. To see the command pages for the commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

Local AAA Server

The Local AAA Server feature allows you to configure your router so that user authentication and authorization attributes currently available on AAA servers are available locally on the router. The attributes can be added to existing framework, such as the local user database or subscriber profile. The local AAA server provides access to the complete dictionary of Cisco IOS supported attributes.

Feature History for Local AAA Server

Release	Modification
12.3(14)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for Local AAA Server, page 1927
- Information About Local AAA Server, page 1928
- How to Configure Local AAA Server, page 1929
- Configuration Examples for Local AAA Server, page 1934
- Additional References, page 1935
- Command Reference, page 1936

Prerequisites for Local AAA Server

- Before using this feature, you must have the **aaa new-model** command enabled.

Information About Local AAA Server

To configure the Local AAA Server feature, you should understand the following concepts:

- Local Authorization Attributes: Overview, page 1928
- Local AAA Attribute Support, page 1928
- AAA Attribute Lists, page 1928
- Validation of Attributes, page 1929

Local Authorization Attributes: Overview

The AAA subsystem (authentication, authorization, and accounting) is responsible for managing all supported attributes that are available to the various services within the Cisco IOS software. As such, it maintains its own local dictionary of all supported attributes. However, prior to Cisco IOS Release 12.3(14)T, most of these authorization options were not available for local (on-box) authorizations.

Local AAA Attribute Support

Effective with Cisco IOS Release 12.3(14)T, you can configure your router so that AAA authentication and authorization attributes currently available on AAA servers are made available on existing Cisco IOS devices. The attributes can be added to existing framework, such as the local user database or subscriber profile. For example, an attribute list can now be added to an existing username, providing the ability for the local user database to act as a local AAA server. For situations in which the local username list is relatively small, this flexibility allows you to provide complete user authentication or authorization locally within the Cisco IOS software without having a AAA server. This ability can allow you to maintain your user database locally or provide a failover local mechanism without having to sacrifice policy options when defining local users.

A subscriber profile allows domain-based clients to have policy applied at the end-user service level. This flexibility allows common policy to be set for all users under a domain in one place and applied there whether or not user authorization is done locally. Effective with Cisco IOS Release 12.3(14)T, an atttribute list can be added to the subscriber profile, allowing the profile to apply all attributes that can be applied to services using AAA servers. Attributes that are configured under the AAA attribute list are merged with the existing attributes that are generated with the existing subscriber profile and passed to the Subscriber Server Switch (SSS) framework for application.

Note Accounting is still done on a AAA server and is not supported by this feature.

AAA Attribute Lists

AAA attribute lists define user profiles that are local to the router. Every attribute that is known to the AAA subsystem is made available for configuration.

The AAA attributes that are defined in the AAA attribute list are standard RADIUS or TACACS+ attributes. However, they are in the Cisco IOS internal format for that attribute. The attributes must be converted from the RADIUS format (for a RADIUS case) to the Cisco IOS AAA interface format. TACACS+ attributes are generally identical to the Cisco IOS AAA interface format.

Converting from RADIUS Format to Cisco IOS AAA Format

You can use the **show aaa attributes protocol radius** command to get the Cisco IOS AAA format of the Internet Engineering Task Force (IETF) RADIUS attribute. The **show** command output provides a complete list of all the AAA attributes that are supported.

Note The conversion from RADIUS to internal AAA is done internally within the AAA framework. RADIUS vendor-specific attributes (VSAs) are usually accurately reflected during conversion. TACACS+ attributes are also usually identical to the local attributes and do not require the conversion process. However, IETF numbered attributes and some special VSAs often require the conversion process.

Validation of Attributes

Attributes are not validated at configuration. The AAA subsystem "knows" only the format that is expected by the services when the service defines a given attribute inside a definition file. However, it cannot validate the attribute information itself. This validation is done by a service when it first uses the attribute. This validation applies whether the AAA server is RADIUS or TACACS+. Thus, if you are not familiar with configuring a AAA server, it is advisable that you test your attribute list on a test device with the service that will be using the list before configuring and using it in a production environment.

How to Configure Local AAA Server

This section contains the following procedures:

- Defining a AAA Attribute List, page 1929 (required)
- Defining a Subscriber Profile, page 1931 (required)
- Monitoring and Troubleshooting a Local AAA Server, page 1932 (optional)

Defining a AAA Attribute List

To define an AAA attribute list, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa attribute list** *list-name*
4. **attribute type** {*name*} {*value*} [**service** *service*] [**protocol** *protocol*]
5. **attribute type** {*name*} {*value*} [**service** *service*] [**protocol** *protocol*]
6. **attribute type** {*name*} {*value*} [**service** *service*] [**protocol** *protocol*]
7. **attribute type** {*name*} {*value*}
8. **attribute type** {*name*} {*value*}
9. **attribute type** {*name*} {*value*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa attribute list** *list-name* **Example:** `Router (config)# aaa attribute list TEST`	Defines a AAA attribute list.
Step 4	**attribute type** {*name*} {*value*} [**service** *service*] [**protocol** *protocol*] **Example:** `Router (config)# attribute type addr-pool "pool name" service ppp protocol ip`	Defines an IP address pool to use.
Step 5	**attribute type** {*name*} {*value*} [**service** *service*] [**protocol** *protocol*] **Example:** `Router (config)# attribute type ip-unnumbered "loopback number" service ppp protocol ip`	Defines the loopback interface to use.
Step 6	**attribute type** {*name*} {*value*} [**service** *service*] [**protocol** *protocol*] **Example:** `Router (config)# attribute type vrf-id "vrf name" service ppp protocol ip`	Defines the virtual route forwarding (VRF) to use.
Step 7	**attribute type** {*name*} {*value*} **Example:** `Router (config)# attribute type ppp-authen-list "aaa list name"`	Defines the AAA authentication list to use.
Step 8	**attribute type** {*name*} {*value*} **Example:** `Router (config)# attribute type ppp-author-list "aaa list name"`	Defines the AAA authorization list to use.
Step 9	**attribute type** {*name*} {*value*} **Example:** `Router (config)# attribute type ppp-acct-list "aaa list name"`	Defines the AAA accounting list to use.

Defining a Subscriber Profile

To define a subscriber profile, perform the following steps.

Note RADIUS users should use the **show aaa attributes** command to map the RADIUS version of the particular attribute to the Cisco IOS AAA version of the string attribute. See the example "Mapping from the RADIUS Version of a Particular Attribute to the Cisco IOS AAA Version: Example."

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **subscriber authorization enable**
4. **subscriber profile** *domain-name*
5. **service local**
6. **exit**
7. **aaa attribute list** *list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**subscriber authorization enable** **Example:** `Router (config)# subscriber authorization enable`	Enables subscriber authorization.
Step 4	**subscriber profile** *domain-name* **Example:** `Router (config)# subscriber profile cisco1.com`	Specifies the username domain that has to be matched and enters subscriber profile configuration mode.
Step 5	**service local** **Example:** `Router (subscriber-profile)# service local`	Specifies that local subscriber authorization should be performed.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Router (subscriber-profile)# exit`	Exits subscriber profile configuration mode.
Step 7	**aaa attribute list** *list-name* **Example:** `Router (config)# aaa attribute list TEST`	Defines the AAA attribute list from which RADIUS attributes are retrieved.

Monitoring and Troubleshooting a Local AAA Server

The following debug commands may be helpful in monitoring and troubleshooting, especially to ensure that domain-based service authorization is being triggered and that location authorization is being called on the local AAA server, which triggers the service.

SUMMARY STEPS

1. **enable**
2. **debug aaa authentication**
3. **debug aaa authorization**
4. **debug aaa per-user**
5. **debug ppp authentication**
6. **debug ppp error**
7. **debug ppp forward**
8. **debug ppp negotiation**
9. **debug radius**
10. **debug sss error**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug aaa authentication** **Example:** `Router# debug aaa authentication`	Displays the methods of authentication being used and the results of these methods.
Step 3	**debug aaa authorization** **Example:** `Router# debug aaa authorization`	Displays the methods of authorization being used and the results of these methods.
Step 4	**debug aaa per-user** **Example:** `Router# debug aaa per-user`	Displays information about PPP session per-user activities.
Step 5	**debug ppp authentication** **Example:** `Router# debug ppp authentication`	Indicates whether a client is passing authentication.
Step 6	**debug ppp error** **Example:** `Router (config)# debug ppp error`	Displays protocol errors and error statistics that are associated with PPP connection negotiation and operation.
Step 7	**debug ppp forward** **Example:** `Router# debug ppp forward`	Displays who is taking control of a session.
Step 8	**debug ppp negotiation** **Example:** `Router# debug ppp negotiation`	Displays PPP packets sent during PPP startup, where PPP options are negotiated.
Step 9	**debug radius** **Example:** `Router# debug radius`	Displays information about the RADIUS server.
Step 10	**debug sss error** **Example:** `Router# debug sss error`	Displays diagnostic information about errors that may occur during SSS call setup.

Configuration Examples for Local AAA Server

This section contains the following configuration examples:

- Local AAA Server: Example, page 1934
- Mapping from the RADIUS Version of a Particular Attribute to the Cisco IOS AAA Version: Example, page 1935

Local AAA Server: Example

The following example shows a Point to Point over Ethernet (PPPoE) group named “bba-group” that is configured for subscriber profile cisco.com (thus, any user with the domain name cisco.com will execute the subscriber profile cisco.com authorization policy). The cisco.com subsciber profile is configured to attach the AAA attribute list “TEST,” which has both “ip vrf forwarding” and “ip unumbered” configured for PPP service under Link Control Protocol (LCP) negotiation. This configuration will essentially cause the named attributes to be applied on the session with the cisco.com domain under the bba-group “pppoe grp1.”

```
aaa authentication ppp template1 local
aaa authorization network template1 local
!
aaa attribute list TEST
   attribute type interface-config "ip unnumbered FastEthernet0" service ppp protocol lcp
   attribute type interface-config "ip vrf forwarding blue" service ppp protocol lcp
!
ip vrf blue
 description vrf blue template1
 rd 1:1
 route-target export 1:1
 route-target import 1:1
!
subscriber authorization enable
!
subscriber profile cisco.com
 service local
 aaa attribute list TEST
!
bba-group pppoe grp1
 virtual-template 1
 service profile cisco.com
!
interface Virtual-Template1
 no ip address
 no snmp trap link-status
 no peer default ip address
 no keepalive
 ppp authentication pap template1
 ppp authorization template1
!
```

Note In some versions of Cisco IOS software, it is better to use the explicit attribute instead of interface-config because it provides better scalability (full VAccess interfaces are not required, and subinterfaces could be used to provide the service). In such a case, you might configure “attribute type ip-unnumbered ‘FastEthernet0’ service ppp protocol ip” instead of “attribute type interface-config ‘ip unnumbered FastEthernet0’ service ppp protocol lcp.”

Mapping from the RADIUS Version of a Particular Attribute to the Cisco IOS AAA Version: Example

The following output example of the **show aaa attributes** command lists RADIUS attributes, which can be used when configuring this feature.

```
Router# show aaa attributes protocol radius

IETF defined attributes:

   Type=4     Name=acl                              Format=Ulong
      Protocol:RADIUS
        Unknown       Type=11    Name=Filter-Id             Format=Binary

Converts attribute 11 (Filter-Id) of type Binary into an internal attribute
named "acl" of type Ulong.  As such, one can configure this attributes locally
by using the attribute type "acl."

Cisco VSA attributes:

  Type=157   Name=interface-config               Format=String

Simply expects a string for the attribute of type "interface-config."
```

Note The **aaa attribute list** command requires the Cisco IOS AAA version of an attribute, which is defined in the "Name" field above.

Additional References

The following sections provide references related to Local AAA Server.

Related Documents

Related Topic	Document Title
AAA, AAA atttribute lists, AAA method lists, and subscriber profiles	The chapter "Configuring Local AAA Server, User Database—Domain to VRF" in "*Cisco 10000 Series Broadband Aggregation and Leased-Line Configuration Guide*
Cisco IOS security commands	*Cisco IOS Security Command Reference*, Release 12.3T
Other Cisco IOS commands	*Cisco IOS Command Reference*, Release 12.3T

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **aaa attribute list**
- **attribute type**

Per-User QoS via AAA Policy Name

The Per-User QoS via AAA Policy Name feature provides the ability to download a policy name that describes quality of service (QoS) parameters for a user session from a RADIUS server and apply them for the particular session.

Feature Specifications for Per-User QoS via AAA Policy Name

Feature History	
Release	**Modification**
12.2(15)B	This feature was introduced.
12.2(15)T	This feature was integrated into Cisco IOS Release 12.2(15)T.
12.2(27)SB	This feature was integrated into Cisco IOS Release 12.2(27)SB.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for Per-User QoS via AAA Policy Name, page 1937
- Information About Per-User QoS via AAA Policy Name, page 1938
- How to Configure Per-User QoS via AAA Policy Name, page 1938
- Configuration Examples for Per-User QoS via AAA Policy Name, page 1939
- Additional References, page 1940
- Command Reference, page 1941
- Glossary, page 1941

Prerequisites for Per-User QoS via AAA Policy Name

Before you configure the Per-User QoS via AAA Policy Name feature, you must locally define on your router the policy whose name is received from the RADIUS server.

Information About Per-User QoS via AAA Policy Name

Effective with Cisco IOS Release 12.2(15)T, separate Cisco vendor-specific attributes (VSAs) are added for the service map.

To configure the Per-User QoS via AAA Policy Name feature, you must understand the following concept:

VSAs Added for Per-User QoS via AAA Policy Name

Two new VSAs have been added for the service map, and the VSAs will bypass the parser while applying the policy for a particular user or session. The new VSAs are as follows:

- vendor-id=9 (Cisco) Vendor type 37 for upstream traffic to input policy name
- vendor-id+9 (Cisco) Vendor type 38 for downstream traffic to output policy name

How to Configure Per-User QoS via AAA Policy Name

This section contains the following procedure:

- Monitoring and Maintaining Per-User QoS via AAA Policy Name, page 1938

To configure per-user QoS, use the authentication, authorization, and accounting (AAA) policy name that you have received from the RADIUS server. To configure QoS policy, refer to the documents listed in the section Related Documents.

Monitoring and Maintaining Per-User QoS via AAA Policy Name

To monitor and maintain per-user QoS using the AAA policy name, use the following **debug** commands:

SUMMARY STEPS

1. **enable**
2. **debug aaa authorization**
3. **debug aaa per-user**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug aaa authorization** **Example:** `Router# debug aaa authorization`	Displays information about AAA/TACACS+ authorization.
Step 3	**debug aaa per-user** **Example:** `Router# debug aaa per-user`	Displays information about per-user QoS parameters.

Configuration Examples for Per-User QoS via AAA Policy Name

This section provides the following configuration example:

- Per-User QoS Using the AAA Policy Name, page 1939

Per-User QoS Using the AAA Policy Name

The following example shows that per-user QoS is being configured using the AAA policy name “policy_class_1_2”:

```
class-map match-all class1
 match access-group 101
class-map match-all class2
 match qos-group 4
 match access-group 101

policy-map policy_class_1_2
 class class1
 bandwidth 3000
 queue-limit 30
 class class2
 bandwidth 2000
 class class-default
 bandwidth 500

peruser_qos_1    Password = "lab"
 Service-Type = Framed,
 Framed-Protocol = PPP,
 Cisco:Cisco-avpair = "ip:sub-policy-In=ssspolicy"
!ssspolicy in the above line is the name of the policy.

peruser_qos_2    Password = "lab"
 Service-Type = Framed,
```

```
Framed-Protocol = PPP,
Cisco:Cisco-avpair = "ip:sub-policy-Out=ssspolicy"
```

Additional References

For additional information related to the Per-User QoS via AAA Policy Name feature, refer to the following references:

Related Documents

Related Topic	Document Title
• AAA per-user and QoS configurations and information about the **policy-map** command	• *Configuring Per-User Configuration* • *Cisco IOS Security Command Reference*, Release 12.2 T

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

This feature uses no new or modified commands. To see the command pages for the commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

Glossary

RADIUS—Remote Authentication Dial-In User Service. RADIUS is a database for authenticating modem and ISDN connections and for tracking connection time.

VSA—vendor-specific attribute. A VSA is an attribute that has been implemented by a particular vendor. It uses the attribute Vendor-Specific to encapsulate the resulting AV pair: essentially, Vendor-Specific = protocol:attribute = value.

Note Refer to the *Internetworking Terms and Acronyms* for terms not included in this glossary.

RADIUS Attribute 5 (NAS-Port) Format Specified on a Per-Server Group Level

The RADIUS Attribute 5 (NAS-Port) Format Specified on a Per-Server Group Level feature allows you to customize configurations for different RADIUS server groups. This flexibility allows customized network access server- (NAS-) port formats to be used instead of global formats.

Feature History for RADIUS Attribute 5 (NAS-Port) Format Specified on a Per-Server Group Level

Release	Modification
12.3(14)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for RADIUS Attribute 5 (NAS-Port) Format Specified on a Per-Server Group Level, page 1944
- Information About RADIUS Attribute 5 (NAS-Port) Format Specified on a Per-Server Group Level, page 1944
- How to Configure RADIUS Attribute 5 (NAS-Port) Format Specified on a Per-Server Group Level, page 1944
- Configuration Examples for RADIUS Attribute 5 (NAS-Port) Format Specified on a Per-Server Group Level, page 1946
- Additional References, page 1947
- Command Reference, page 1948

Prerequisites for RADIUS Attribute 5 (NAS-Port) Format Specified on a Per-Server Group Level

- You must be running a Cisco IOS image that contains the authentication, authorization, and accounting (AAA) component.

Information About RADIUS Attribute 5 (NAS-Port) Format Specified on a Per-Server Group Level

To configure the RADIUS Attribute 5 (NAS-Port) Format Specified on a Per-Server Group Level feature, you should understand the following concept:

- RADIUS Attribute 5 Format Customization, page 1944

RADIUS Attribute 5 Format Customization

Prior to Cisco IOS Release 12.3(14)T, Cisco IOS software allowed RADIUS attributes that were sent in access requests or accounting requests to be customized on a global basis. You could customize how each configurable attribute should function when communicating with a RADIUS server. Since the implementation of server groups, global attribute configurations were not flexible enough to address the different customizations that were required to support the various RADIUS servers with which a router might be interacting. For example, if you configured the global **radius-server attribute nas-port format** command option, every service on the router that interacted with a RADIUS server was used in the same way.

Effective with Cisco IOS Release 12.3(14)T, you can configure your router to support override flexibility for per-server groups. You can configure services to use specific named methods for different service types on a RADIUS server. The service types can be set to use their own respective service groups. This flexibility allows customized NAS-port formats to be used instead of the global formats.

How to Configure RADIUS Attribute 5 (NAS-Port) Format Specified on a Per-Server Group Level

This section contains the following procedures:

- Configuring the RADIUS Attribute 5 Format on a Per-Server Group Level, page 1944
- Monitoring and Maintaining RADIUS Attribute 5 Format on a Per-Server Group Level, page 1946

Configuring the RADIUS Attribute 5 Format on a Per-Server Group Level

To configure your router to support the RADIUS Attribute 5 format on a per-server group level, perform the following steps.

Note To use this per-server group capability, you must actively use a named method list within your services. You can configure one client to use a specific named method while other clients use the default format.

Prerequisites

Before performing these steps, you should first configure method lists for AAA as is applicable for your situation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa group server radius** *group-name*
4. **server** *ip-address* [**auth-port** *port-number*] [**acct-port** *port-number*]
5. **attribute nas-port format** *format-type* [*string*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa group server radius** *group-name* **Example:** `Router (config)# aaa group server radius radius1`	Groups different RADIUS server hosts into distinct lists and distinct methods and enters server-group configuration mode.
Step 4	**server** *ip-address* [**auth-port** *port-number*] [**acct-port** *port-number*] **Example:** `Router (server-group)# server 172.101.159.172 auth-port 1645 acct-port 1646`	Configures the IP address of the RADIUS server for the group server.
Step 5	**attribute nas-port format** *format-type* [*string*] **Example:** `Router (server-group)# attribute nas-port format d`	Configures a service to use specific named methods for different service types. • The service types can be set to use their own respective server groups.

Monitoring and Maintaining RADIUS Attribute 5 Format on a Per-Server Group Level

To monitor and maintain RADIUS Attribute 5 Format on a Per-Server Group Level, perform the following steps (the **debug** commands may be used separately):

SUMMARY STEPS

1. **enable**
2. **debug aaa sg-server selection**
3. **debug radius**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug aaa sg-server selection** **Example:** `Router# debug aaa sg-server selection`	Displays information about why the RADIUS and TACACS+ server group system in a router is choosing a particular server.
Step 3	**debug radius** **Example:** `Router# debug radius`	Displays information showing that a server group has been selected for a particular request.

Configuration Examples for RADIUS Attribute 5 (NAS-Port) Format Specified on a Per-Server Group Level

This section provides the following configuration example:

- RADIUS Attribute 5 Format Specified on a Per-Server Level: Example, page 1946

RADIUS Attribute 5 Format Specified on a Per-Server Level: Example

The following configuration example shows a leased-line PPP client that has chosen to send no RADIUS Attribute 5 while the default is to use format d:

```
interface Serial2/0
 no ip address
 encapsulation ppp
 ppp accounting SerialAccounting
 ppp authentication pap
```

```
aaa accounting network default start-stop group radius
aaa accounting network SerialAccounting start-stop group group1

aaa group server radius group1
 server 10.101.159.172 auth-port 1645 acct-port 1646
 attribute nas-port none

radius-server host 10.101.159.172 auth-port 1645 acct-port 1646
radius-server attribute nas-port format d
```

Additional References

The following sections provide references related to RADIUS Attribute 5 (NAS-Port) Format Specified on a Per-Server Group Level.

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Security Command Reference*, Release 12.3T
Configuring AAA and AAA method lists	"Authentication, Authorization, and Accounting (AAA)" section of *Cisco IOS Security Configuration Guide*, Release 12.3.

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following new command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **attribute nas-port format**

RADIUS Attribute 8 (Framed-IP-Address) in Access Requests

Feature History

Release	Modification
12.2(11)T	RADIUS Attribute 8 (Framed-IP-Address) in Access Requests was introduced.

This feature module describes the RADIUS Attribute 8 (Framed-IP-Address) in Access Requests feature. It includes information on the benefits of the new feature, supported platforms, and related documents.

This document includes the following sections:

- Feature Overview, page 1949
- Supported Platforms, page 1950
- Supported Standards, MIBs, and RFCs, page 1951
- Prerequisites, page 1951
- Configuration Tasks, page 1951
- Configuration Examples, page 1952
- Command Reference, page 1952

Feature Overview

The RADIUS Attribute 8 (Framed-IP-Address) in Access Requests feature makes it possible for a network access server (NAS) to provide the RADIUS server with a hint of the user IP address in advance of user authentication. An application can be run on the RADIUS server to use this hint and build a table (map) of user names and addresses. Using the mapping information, service applications can begin preparing user login information to have available upon successful user authentication.

How It Works

When a network device dials in to a NAS that is configured for RADIUS authentication, the NAS begins the process of contacting the RADIUS server in preparation for user authentication. Typically, the IP address of the dial-in host is not communicated to the RADIUS server until after successful user authentication. Communicating the device IP address to the server in the RADIUS access request allows other applications to begin to take advantage of that information.

As the NAS is setting up communication with the RADIUS server, the NAS assigns an IP address to the dial-in host from a pool of IP addresses configured at the specific interface. The NAS sends the IP address of the dial-in host to the RADIUS server as attribute 8. At that time, the NAS sends other user information, such as the user name, to the RADIUS server.

After the RADIUS server receives the user information from the NAS, it has two options:

- If the user profile on the RADIUS server already includes attribute 8, the RADIUS server can override the IP address sent by the NAS with the IP address defined as attribute 8 in the user profile. The address defined in the user profile is returned to the NAS.
- If the user profile does not include attribute 8, the RADIUS server can accept attribute 8 from the NAS, and the same address is returned to the NAS.

The address returned by the RADIUS server is saved in memory on the NAS for the life of the session. If the NAS is configured for RADIUS accounting, the accounting start packet sent to the RADIUS server includes the same IP address as in attribute 8. All subsequent accounting packets, updates (if configured), and stop packets will also include the same IP address provided in attribute 8.

Benefits

The RADIUS Attribute 8 (Framed-IP-Address) in Access Requests feature makes it possible to run applications on the RADIUS server that build mapping tables of users and IP addresses. The server can then use the mapping table information in other applications, such as preparing customized user login pages in advance of a successful user authentication with the RADIUS server.

Related Documents

- "Configuring Authentication" and "Configuring RADIUS" chapters*, Cisco Security Configuration Guide*, Cisco IOS Release 12.1
- RFC 2138, *Remote Authentication Dial In User Service (RADIUS)*

Supported Platforms

This feature is supported on the following platforms:

- Cisco 2600 series
- Cisco 3600 series
- Cisco AS5300 series
- Cisco AS5400 series

- Cisco AS5800
- Cisco 7200 series

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

No new or modified MIBs are supported by this feature.

For descriptions of supported MIBs and how to use MIBs, see the Cisco MIB web site on Cisco Connection Online (CCO) at http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml.

RFCs

No new or modified RFCs are supported by this feature.

Prerequisites

Sending RADIUS attribute 8 in the RADIUS access requests assumes that the login host has been configured to request its IP address from the NAS server. It also assumes that the login host has been configured to accept an IP address from the NAS.

The NAS must be configured with a pool of network addresses on the interface supporting the login hosts.

Configuration Tasks

See the following section for the configuration task for the RADIUS Attribute 8 (IP-Framed-Address) in Access Requestsfeature: Configuring RADIUS Attribute 8 in Access Requests (required).

Configuring RADIUS Attribute 8 in Access Requests

To send RADIUS attribute 8 in the access request, use the following global configuration command:

Command	Purpose
`Router(config)# radius-server attribute 8 include-in-access-req`	Sends RADIUS attribute 8 in access-request packets.

Verifying RADIUS Attribute 8 in Access Requests

To verify that RADIUS attribute 8 is being sent in access requests, use the following commands in privileged EXEC mode. Attribute 8 should be present in all ppp access requests.

Command	Purpose
`Router# more system:running-config`	Displays the contents of the current running configuration file. (Note that the **more system:running-config** command has replaced the **show running-config** command.)
`Router# debug radius`	Displays information associated with RADIUS. The output of this command shows whether attribute 8 is being sent in access requests.

Configuration Examples

The following example shows a NAS configuration that sends the IP address of the dial-in host to the RADIUS server in the RADIUS access request. The NAS is configured for RADIUS authentication, authorization, and accounting (AAA). A pool of IP addresses (async1-pool) has been configured and applied at interface Async1.

```
aaa new-model
aaa authentication login default group radius
aaa authentication ppp default group radius
aaa authorization network default group radius
aaa accounting network default start-stop group radius
!
ip address-pool local
!
interface Async1
 peer default ip address pool async1-pool
!
ip local pool async1-pool 209.165.200.225 209.165.200.229
!
radius-server host 172.31.71.146 auth-port 1645 acct-port 1646
radius-server retransmit 3
radius-server attribute 8 include-in-access-req
radius-server key radhost
```

Command Reference

The following new command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **radius-server attribute 8 include-in-access-req**

RADIUS Attribute 82: Tunnel Assignment ID

Feature History

Release	Modification
12.2(4)T	This feature was introduced.
12.2(4)T3	Support for the Cisco 7500 series routers was added.
12.2(11)T	This feature was integrated into Cisco IOS Release 12.2(11)T and support was added for the Cisco 1760, Cisco AS5300, Cisco AS5350, Cisco AS5400, Cisco AS5800 and Cisco AS5850 platforms.
12.2(27)SB	This feature was integrated into Cisco IOS Release 12.2(27)SB.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Feature Overview, page 1953
- Supported Platforms, page 1954
- Supported Standards, MIBs, and RFCs, page 1955
- Prerequisites, page 1955
- Configuration Tasks, page 1955
- Configuration Examples, page 1956
- Command Reference, page 1957

Feature Overview

The RADIUS Attribute 82: Tunnel Assignment ID feature allows the Layer 2 Transport Protocol access concentrator (LAC) to group users from different per-user or domain RADIUS profiles into the same active tunnel. Previously, Cisco IOS software assigned a separate virtual private dialup network (VPDN) tunnel for each per-user or domain RADIUS profile, even if tunnels with identical endpoints already

existed. The RADIUS Attribute 82: Tunnel Assignment ID feature defines a new avpair, Tunnel-Assignment-ID, which allows the LAC to group users from different RADIUS profiles into the same tunnel if the chosen endpoint, tunnel type, and Tunnel-Assignment-ID are identical. This feature introduces new software functionality. No new commands are introduced with this feature.

Benefits

The RADIUS Attribute 82: Tunnel Assignment ID feature improves LAC and L2TP network server (LNS) performance by reducing memory usage, because fewer tunnel data structures must be maintained. This feature allows the LAC and LNS to handle a higher volume of users without negatively impacting router performance.

Restrictions

This feature is designed only for VPDN dial-in applications. It does not support VPDN dial-out.

Related Documents

- *Cisco IOS Dial Technologies Configuration Guide,* Release 12.2
- *Cisco IOS Dial Technologies Command Reference*, Release 12.2
- *Cisco IOS Wide-Area Networking Configuration Guide,* Release 12.2
- *Cisco IOS Wide-Area Networking Command Reference,* Release 12.2

Supported Platforms

- Catalyst 4000 Gateway
- Cisco 806
- Cisco 1400 series
- Cisco 1600 series
- Cisco 1700 series
- Cisco 2600 series
- Cisco 3600 series
- Cisco 3700 series
- Cisco 7100 series
- Cisco 7200 series
- Cisco 7500 series
- Cisco AS5300
- Cisco AS5400
- Cisco AS5800
- Cisco IGX 8400 URM

- Cisco MGX 8850
- Cisco ubr7200

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

No new or modified MIBs are supported by this feature.

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

No new or modified RFCs are supported by this feature.

Prerequisites

You must be using a Cisco platform that supports VPDN to use this feature.

Configuration Tasks

None

Verifying RADIUS Attribute 82

To verify that RADIUS attribute 82 is being used by the LAC during tunnel authorization, use the following privileged EXEC command:

Command	Purpose
`Router# debug radius`	Displays information associated with RADIUS. The output of this command shows whether attribute 82 is being sent in access requests.

Configuration Examples

This section provides the following configuration examples:

- LAC Configuration Example
- LNS Configuration Example
- RADIUS Configuration Example

LAC Configuration Example

The following example configures VPDN on the LAC:

```
hostname lac
aaa new-model
aaa authentication ppp default group radius
aaa authorization network default group radius

vpdn enable
 vpdn authen-before-forward

interface Serial2/0:23
 no ip address
 encapsulation ppp
 dialer-group 1
 isdn switch-type primary-5ess
 no fair-queue

dialer-list 1 protocol ip permit

radius-server host lac-radiusd auth-port 1645 acct-port 1646
radius-server retransmit 3
radius-server key rad123
```

LNS Configuration Example

The following example configures VPDN on the LNS:

```
hostname lns
aaa new-model
aaa authentication ppp default group radius
aaa authorization network default group radius

vpdn enable

vpdn-group 1
 accept-dialin
  protocol any
  virtual-template 1
```

```
interface Loopback0
 ip address 10.1.1.3 255.255.255.0

interface Virtual-Template1
 ip unnumbered Loopback0
 no keepalive
 peer default ip address pool mypool
 ppp authentication chap

ip local pool mypool 10.1.1.10 10.1.1.50

radius-server host lns-radiusd  auth-port 1645 acct-port 1646
radius-server retransmit 3
radius-server key cisco
```

RADIUS Configuration Example

The following examples configure the RADIUS server to group sessions in a tunnel:

Per-User Configuration

```
user@router.com Password = "cisco" Service-Type = Outbound,
       Tunnel-Type = :1:L2TP,
       Tunnel-Server-Endpoint = :1:"10.14.10.54",
       Tunnel-Assignment-Id = :1:"router"

client@router.com Password = "cisco" Service-Type = Outbound,
       Tunnel-Type = :1:L2TP,
       Tunnel-Server-Endpoint = :1:"10.14.10.54",
       Tunnel-Assignment-Id = :1:"router"
```

Domain Configuration

```
eng.router.com Password = "cisco" Service-Type = Outbound,
        Tunnel-Type = :1:L2TP,
        Tunnel-Server-Endpoint = :1:"10.14.10.54",
        Tunnel-Assignment-Id = :1:"router"

sales.router.com Password = "cisco" Service-Type = Outbound,
        Tunnel-Type = :1:L2TP,
        Tunnel-Server-Endpoint = :1:"10.14.10.54",
        Tunnel-Assignment-Id = :1:"router"
```

Command Reference

This feature uses no new or modified commands. To see the command pages for the commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

RADIUS Attribute 104

The RADIUS Attribute 104 feature allows you to specify private routes (attribute 104) in your RADIUS authorization profile. The private routes affect only packets that are received on an individual interface. The routes are stored apart from the global routing table and are not injected into any routing protocols for redistribution.

Feature History for RADIUS Attribute 104

Release	Modification
12.3(7)T	This feature was introduced.
12.3(14)T	Support for the map display extension functionality was added: The **detailed** keyword was added to the **show route-map** command.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for RADIUS Attribute 104, page 1959
- Restrictions for RADIUS Attribute 104, page 1960
- Information About RADIUS Attribute 104, page 1960
- How to Apply RADIUS Attribute 104, page 1961
- Configuration Examples for RADIUS Attribute 104, page 1964
- Additional References, page 1965
- Command Reference, page 1966

Prerequisites for RADIUS Attribute 104

- You must be using a Cisco RADIUS server.
- You should be familiar with configuring RADIUS.

- You should be familiar with policy-based routing (PBR) and private routes.
- You should be familiar with configuring access control lists (ACLs).
- Before using the RADIUS Attribute 104 feature, you must configure RADIUS AAA authorization and RADIUS route download.
- The following memory bytes are required:
  - One route map—50 bytes.
  - One match-set clause—600 bytes.
  - One extended ACL—366 bytes.
  - For N number of attribute 104s, the memory requirement is (600+366)*N+50=1000*N(approximate) per user.

Restrictions for RADIUS Attribute 104

- If you already have PBR locally (statically) configured under the interface, and you specify attribute 104, the locally configured PBR will be disabled.
- If a pseudo next-hop address is involved, there must be a route available in the routing table for the next-hop address. If a route is not available, the packet will not be policy routed.
- Policy routing does not order the match-set clauses and relies on the first match, so you should specify the attributes in the order in which you want them to be matched.
- Metric numbers cannot be used in the attribute.

Information About RADIUS Attribute 104

Before using the RADIUS Attribute 104 feature, you should understand the following concepts:

- Policy-Based Routing: Background, page 1960
- Attribute 104 and the Policy-Based Route Map, page 1961

Policy-Based Routing: Background

PBR provides a mechanism for the forwarding, or routing of, data packets on the basis of defined policies. The policies are not wholly dependent on the destination address but rather on other factors, such as type of service, source address, precedence, port numbers, or protocol type.

Policy-based routing is applied to incoming packets. All packets that are received on an interface that has policy-based routing enabled are considered for policy-based routing. The router passes the packets through enhanced packet filters called route maps. On the basis of the criteria that are defined in the route maps, the packets are forwarded to the appropriate next hop.

Each entry in a route map statement contains a combination of match clauses and set clauses or commands. The match clauses define the criteria for whether appropriate packets meet the particular policy (that is, whether the conditions are met). The set clauses provide instruction for how the packets should be routed after they have met the match criteria. The match clause specifies which set of filters a packet must match for the corresponding set clause to be applied.

Attribute 104 and the Policy-Based Route Map

This section discusses the attribute 104 feature and how it works with policy-based route maps.

RADIUS Attribute 104 Overview

Using the the RADIUS Attribute 104 feature, you can specify private routes in your RADIUS authorization profile. The private routes you specify will affect only packets that are received on an individual interface. The routes are stored apart from the global routing table and are not injected into any routing protocols for redistribution.

Permit Route Map

Route map statements can be marked as "permit" or "deny." If the statement is marked "permit," the set clause is applied to the packets that match the match criteria. For attribute 104, when you are configuring the route map, you need to mark the route map as "permit," as follows. (To configure a route map, see the chapter "Configuring Policy-Based Routing" in the *Cisco IOS Quality of Service Solutions Configuration Guide.*)

```
route-map map-tag permit sequence-number
```

Default Private Route

The policy routing process proceeds through the route map until a match is found. If no match is found in the route map, the global routing table is consulted. If you have specified a default route in your user profile, any further routes beyond the default route are effectively ignored.

Route Map Order

You need to specify route maps on the server in the order that you want them to be applied.

How to Apply RADIUS Attribute 104

This section contains the following procedures:

- Applying RADIUS Attribute 104 to Your User Profile, page 1961
- Verifying Route Maps, page 1962
- Troubleshooting the RADIUS Profile, page 1963

Applying RADIUS Attribute 104 to Your User Profile

You can apply RADIUS attribute 104 to your user profile by adding the following to the RADIUS server database.

SUMMARY STEPS

1. Apply RADIUS attribute 104 to your user profile.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Apply RADIUS attribute 104 to your user profile.	`Ascend-Private-Route="dest_addr/netmask next_hop"` The destination network address of the router is "dest_addr/netmask," and the address of the next-hop router is "next_hop."

Examples

The following is a sample user profile that creates three private routes that are associated with the caller:

```
username Password="ascend"; User-Service=Framed-User

   Framed-Protocol=PPP,
   Framed-Address=10.1.1.1,
   Framed-Netmask=255.0.0.0,
   Ascend-Private-Route="172.1.0.0/16 10.10.10.1"
   Ascend-Private-Route="192.1.1.1/32 10.10.10.2"
   Ascend-Private-Route="10.20.0.0/16 10.10.10.3"
   Ascend-Private-Route="0.0.0.0/0 10.10.10.4"
```

Using the above profile, the private routing table for the connection contains the following routes, including a default route:

```
Destination/Mask     Gateway
172.1.0.0/16         10.10.10.1
192.1.1.1/32         10.10.10.2
10.20.20.20/16       10.10.10.3
0.0.0.0/0            10.10.10.4
```

Verifying Route Maps

You can use the following **show** commands to verify the route maps that have been configured.

SUMMARY STEPS

1. **enable**
2. **show ip policy**
3. **show route-map** [*map-name* | **dynamic** [*dynamic-map-name* | **application** [*application-name*]] | **all**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip policy** **Example:** `Router# show ip policy`	Displays the route map that is used for policy routing.
Step 3	**show route-map** [*map-name* | **dynamic** [*dynamic-map-name* | **application** [*application-name*]] | **all**] **Example:** `Router# show route-map`	Displays all route maps that are configured or only the one that is specified.

Troubleshooting the RADIUS Profile

If your private route configuration is not working properly, you may want to reread the section "Policy-Based Routing: Background." This section may help you determine what is happening to the packets. In addition, the following **debug** commands can be used to troubleshoot your RADIUS profile.

SUMMARY STEPS

1. **enable**
2. **debug radius**
3. **debug aaa per-user**
4. **debug ip policy**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug radius** **Example:** `Router# debug radius`	Displays information associated with RADIUS.
Step 3	**debug aaa per-user** **Example:** `Router# debug aaa per-user`	Displays the attributes that are applied to each user as the user authenticates.
Step 4	**debug ip policy** **Example:** `Router# debug ip policy`	Displays IP routing packet activity.

Configuration Examples for RADIUS Attribute 104

This section includes the following configuration example:

• Route-Map Configuration in Which Attribute 104 Has Been Applied: Example, page 1964

Route-Map Configuration in Which Attribute 104 Has Been Applied: Example

The following output is a typical route-map configuration to which attribute 104 has been applied:

```
Router# show route-map dynamic

route-map AAA-01/08/04-14:13:59.542-1-AppSpec, permit, sequence 0, identifier 1639994476
  Match clauses:
    ip address (access-lists): PBR#1 PBR#2
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes
route-map AAA-01/08/04-14:13:59.542-1-AppSpec, permit, sequence 1, identifier 1640264784
  Match clauses:
    ip address (access-lists): PBR#3 PBR#4
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes
route-map AAA-01/08/04-14:13:59.542-1-AppSpec, permit, sequence 2, identifier 1645563704
  Match clauses:
    ip address (access-lists): PBR#5 PBR#6
    length 10 100
  Set clauses:
    ip next-hop 10.1.1.1
    ip gateway10.1.1.1
  Policy routing matches: 0 packets, 0 bytes
 Current active dynamic routemaps = 1
```

Additional References

The following sections provide references related to RADIUS Attribute 104.

Related Documents

Related Topic	Document Title
Configuring RADIUS	The "Configuring RADIUS" chapter of the "Security Server Protocols" section of the *Cisco IOS Security Configuration Guide*
Configuring policy-based routing	"Configuring Policy-Based Routing" chapter of the "Classification" section of the *Cisco IOS Quality of Service Configuration Guide*
Configuring access control lists	• The "Access Control Lists: Overview and Guidelines" chapter of the "Traffic Filtering and Firewalls" section of the *Cisco IOS Security Configuration Guide* • *IP Access List Entry Sequence Numbering*, Release 12.3(2)T
Configuring RADIUS AAA authorization and RADIUS route download	*RADIUS Route Download*, Release 12.2(8)T
Security commands	*Cisco IOS Security Command Reference*, Release 12.3 T
Quality of Service (QoS) commands (for policy-based routing commands)	*Cisco IOS Quality of Service Solutions Command Reference*, Release 12.3 T

Standards

Standards	Title
There are no new or modified standards associated with this feature.	—

MIBs

MIBs	MIBs Link
There are no new or modified MIBs associated with this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
There are no new or modified RFCs associated with this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following modified commands are pertinent to this feature. To see the command pages for these commands and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **show ip policy**
- **show route-map**

RADIUS Progress Codes

Feature History

Release	Modification
12.2(11)T	This feature was introduced.
12.2(27)SB	This feature was integrated into Cisco IOS Release 12.2(27)SB.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

- Feature Overview, page 1967
- Supported Platforms, page 1968
- Supported Standards, MIBs, and RFCs, page 1969
- Prerequisites, page 1969
- Configuration Tasks, page 1969
- Configuration Examples, page 1969
- Command Reference, page 1970
- Glossary, page 1971

Feature Overview

The RADIUS Progress Codes feature adds additional progress codes—10, 30, 33, 41, 60, 65, 67—to RADIUS attribute 196 (Ascend-Connect-Progress), which indicates the connection state before the call is disconnected via progress codes.

Attribute 196 is sent in network, exec, and resource accounting “start” and “stop” records. This attribute can facilitate call failure debugging because each progress code identifies accounting information relevant to the connection state of a call. The attribute is activated by default; when an accounting “start” or “stop” accounting record is requested, authentication, authorization, and accounting (AAA) will add attribute 196 into the record as part of the standard attribute list.

Note In accounting "start" records, attribute 196 does not have a value.

The newly supported progress codes are defined in Table 78.

Table 78 Newly Supported Progress Codes for Attribute 196

Code	Description
10	Modem allocation and negotiation is complete; the call is up.
30	The modem is up.
33	The modem is waiting for result codes.
41	The max TNT is establishing the TCP connection by setting up a TCP clear call.
60	Link control protocol (LCP) is the open state with PPP and IP Control Protocol (IPCP) negotiation; the LAN session is up.
65	PPP negotiation occurs and, initially, the LCP negotiation occurs; LCP is in the open state.
67	After PPP negotiation with LCP in the open state occurs, IPCP negotiation begins.

Note Progress codes 33, 30, and 67 are generated and seen via debugs on the NAS; all other codes are generated and seen via debugs and the accounting record on the RADIUS server.

Benefits

The RADIUS Progress Codes feature adds support for the following progress codes to RADIUS attribute 196 (Ascend-Connect-Progress): 10, 30, 33, 41, 60, 65, 67. Attribute 196 is valuable because the progress codes, which are sent in accounting "start" and "stop" records, facilitate the debugging of call failures.

Related Documents

- *Cisco IOS Security Command Reference*, Release 12.2
- "Configuring Accounting" chapter in *Cisco IOS Security Configuration Guide*, Release 12.2
- "RADIUS Attributes" appendix in the *Cisco IOS Security Configuration Guide*, Release 12.2

Supported Platforms

- Cisco AS5300 series
- Cisco AS5400 series
- Cisco AS5800 series
- Cisco AS5850 series

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

No new or modified RFCs are supported by this feature.

Prerequisites

Before attribute 196 (Ascend-Connect-Progress) can be sent in accounting "start" and "stop" records, you must perform the following tasks:

- Enable AAA.
- Enable exec, network, or resource accounting.

For information on completing these tasks, refer to the AAA sections of the *Cisco IOS Security Configuration Guide*, Release 12.2.

When these tasks are completed, attribute 196 is active by default.

Configuration Tasks

None

Verifying Attribute 196

To verify attribute 196 in accounting "start" and "stop records, use one of the following commands in privileged EXEC mode:

Command	Purpose
`Router# debug aaa accounting`	Displays information on accountable events as they occur.
`Router# show radius statistics`	Displays the RADIUS statistics for accounting and authentication packets.

Configuration Examples

This section provides the following configuration example:

- Sample Debug Output Example

Sample Debug Output Example

The following example is a sample debug output from the **debug ppp negotiation** command. This debug output is used to verify that accounting "stop" records have been generated and that attribute 196 (Ascend-Connect-Progress) has a value of 65.

```
Tue Aug 7 06:21:03 2001
        NAS-IP-Address = 10.0.58.62
        NAS-Port = 20018
        Vendor-Specific = ""
        NAS-Port-Type = ISDN
        User-Name = "peer_16a"
        Called-Station-Id = "5213124"
        Calling-Station-Id = "5212175"
        Acct-Status-Type = Stop
        Acct-Authentic = RADIUS
        Service-Type = Framed-User
        Acct-Session-Id = "00000014"
        Framed-Protocol = PPP
        Framed-IP-Address = 60.0.0.2
        Acct-Input-Octets = 3180
        Acct-Output-Octets = 3186
        Acct-Input-Packets = 40
        Acct-Output-Packets = 40
        Ascend-Connect-Pr = 65
        Acct-Session-Time = 49
        Acct-Delay-Time = 0
        Timestamp = 997190463
        Request-Authenticator = Unverified
```

Command Reference

This feature uses no new or modified commands. To see the command pages for the commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

Glossary

AAA—authentication, authorization, and accounting. Suite of network security services that provide the primary framework through which access control can be set up on your Cisco router or access server.

attribute—RADIUS Internet Engineering Task Force (IETF) attributes are the original set of 255 standard attributes that are used to communicate AAA information between a client and a server. Because IETF attributes are standard, the attribute data is predefined and well known; thus all clients and servers who exchange AAA information via IETF attributes must agree on attribute data such as the exact meaning of the attributes and the general bounds of the values for each attribute.

EXEC accounting—Provides information about user EXEC terminal sessions of the network access server.

IPCP—IP Control Protocol. A protocol that establishes and configures IP over PPP.

LCP—link control protocol. A protocol that establishes, configures, and tests data-link connections for use by PPP.

network accounting—Provides information for all PPP, Serial Line Internet Protocol (SLIP), or AppleTalk Remote Access Protocol (ARAP) sessions, including packet and byte counts.

PPP—Point-to-Point Protocol. Successor to SLIP that provides router-to-router and host-to-network connections over synchronous and asynchronous circuits. Whereas SLIP was designed to work with IP, PPP was designed to work with several network layer protocols, such as IP, IPX, and ARA. PPP also has built-in security mechanisms, such as CHAP and PAP. PPP relies on two protocols: LCP and NCP.

RADIUS—Remote Authentication Dial-In User Service. RADIUS is a distributed client/server system that secures networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco routers and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.

resource accounting—Provides “start” and “stop” records for calls that have passed user authentication, and provides “stop” records for calls that fail to authenticate.

RADIUS Timeout Set During Pre-Authentication

Some call sessions for Internet service provider (ISP) subscribers are billed through authentication, authorization, and accounting (AAA) messages in a prepaid time model. When these subscribers are preauthenticated, a RADIUS server checks for any remaining credit in the prepaid time model and sets a session timeout based on the credit available. The RADIUS Timeout Set During Pre-Authentication feature is useful in situations where the PPP authentication that follows the preauthentication phase of these call sessions does not return the Session-Timeout value (RADIUS attribute 27), and therefore allows the ISP to add call setup time to the subscriber's bill.

Feature Specifications for the RADIUS Timeout Set During Pre-Authentication Feature

Feature History	
Release	**Modification**
12.2(15)T	This feature was introduced.
12.2(27)SB	This feature was integrated into Cisco IOS Release 12.2(27)SB.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for the RADIUS Timeout Set During Pre-Authentication Feature, page 1974
- Information About the RADIUS Timeout Set During Pre-Authentication Feature, page 1974
- How to Configure the RADIUS Timeout Set During Pre-Authentication Feature, page 1974
- Additional References, page 1975
- Command Reference, page 1976

Prerequisites for the RADIUS Timeout Set During Pre-Authentication Feature

- This feature is specific to RADIUS. Basic AAA authentication and preauthentication must be configured.
- Preauthentication and normal PPP authentication are required for legacy functionality.

Information About the RADIUS Timeout Set During Pre-Authentication Feature

You need to understand the following concept about the RADIUS Timeout Set During Pre-Authentication feature:

- RADIUS Attribute 27 and the PPP Authentication Phase, page 1974

RADIUS Attribute 27 and the PPP Authentication Phase

The RADIUS Timeout Set During Pre-Authentication feature was developed for ISPs that want to bill dial-in subscribers for call setup time and the entire duration of the call session. These subscribers are billed through AAA messages in a prepaid time model. When the subscribers are preauthenticated, a RADIUS server checks for any remaining credit in the prepaid time model and sets a session timeout (in minutes or seconds) based on the credit available. This time can range from a few seconds for ISDN users, to much longer for asynchronous dial-up subscribers.

Until the RADIUS Timeout Set During Pre-Authentication feature was developed, the value of RADIUS attribute 27, which is returned during the preauthentication phase of a call, was either ignored or overwritten during the PPP authentication phase. Even when the PPP authentication phase did not return a value for attribute 27, the old value obtained during the preauthentication phase was being ignored.

With the RADIUS Timeout Set During Pre-Authentication feature introduced for Cisco IOS Release 12.2(15)T, if the PPP authentication phase does not return a value for attribute 27, the old value that was returned during the preauthentication phase is saved and used to time out the session; attribute 27 is saved in a preauthentication database for future use. However, if the PPP authentication user profile has a session timeout configured and PPP authentication succeeds, the new value downloaded during PPP authentication overwrites the old attribute 27 value. By setting the session timeout value in the preauthentication phase itself, the service provider can bill the subscriber for the call setup time and the call duration.

How to Configure the RADIUS Timeout Set During Pre-Authentication Feature

No new configuration is required. The RADIUS Timeout Set During Pre-Authentication feature is included in all Cisco platforms that support preauthentication, and that have RADIUS attribute 27, Session-Timeout, specified in a preauthentication user profile.

Additional References

For additional information related to the RADIUS Timeout Set During Pre-Authentication feature, refer to the following sections:

- Related Documents, page 1975
- Standards, page 1975
- MIBs, page 1975
- RFCs, page 1976
- Technical Assistance, page 1976

Related Documents

Related Topic	Document Title
RADIUS attributes and user profiles	*Cisco IOS Security Configuration Guide*, Release 12.2. Refer to "RADIUS Attributes" in the Appendixes.

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

This feature uses no new or modified commands. To see the command pages for the commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

RADIUS Tunnel Attribute Extensions

Feature History

Release	Modification
12.1(5)T	This feature was introduced.
12.2(4)B3	This feature was integrated into Cisco IOS Release 12.2(4)B3.
12.2(13)T	This feature was integrated into Cisco IOS Release 12.2(13)T.

This feature module describes the RADIUS Tunnel Attribute Extensions feature. It includes the following sections:

- Feature Overview, page 1977
- Supported Platforms, page 1979
- Supported Standards, MIBs, and RFCs, page 1980
- Prerequisites, page 1980
- Configuration Tasks, page 1980
- Configuration Examples, page 1981
- Command Reference, page 1982
- Glossary, page 1983

Feature Overview

The RADIUS Tunnel Attribute Extensions feature introduces RADIUS attribute 90 (Tunnel-Client-Auth-ID) and RADIUS attribute 91 (Tunnel-Server-Auth-ID). Both attributes help support the provision of compulsory tunneling in virtual private networks (VPNs) by allowing the user to specify authentication names for the network access server (NAS) and the RADIUS server.

How It Works

Once a NAS has set up communication with a RADIUS server, you can enable a tunneling protocol. Some applications of tunneling protocols are voluntary, but others involve compulsory tunneling; that is, a tunnel is created without any action from the user and without allowing the user any choice in the matter. In those cases, new RADIUS attributes are needed to carry the tunneling information from the NAS to the RADIUS server to establish authentication. These new RADIUS attributes are listed in Table 79.

Note In compulsory tunneling, any security measures in place apply only to traffic between the tunnel endpoints. Encryption or integrity protection of tunneled traffic must not be considered as a replacement for end-to-end security.

Table 79 RADIUS Tunnel Attributes

Number	IETF RADIUS Tunnel Attribute	Equivalent TACACS+ Attribute	Supported Protocols	Description
90	Tunnel-Client-Auth-ID	tunnel-id	• Layer 2 Forwarding (L2F) • Layer 2 Tunneling Protocol (L2TP)	Specifies the name used by the tunnel initiator (also known as the NAS[1]) when authenticating tunnel setup with the tunnel terminator.
91	Tunnel-Server-Auth-ID	gw-name	• Layer 2 Forwarding (L2F) • Layer 2 Tunneling Protocol (L2TP)	Specifies the name used by the tunnel terminator (also known as the Home Gateway[2]) when authenticating tunnel setup with the tunnel initiator.

1. When L2TP is used, the NAS is referred to as an L2TP access concentrator (LAC).
2. When L2TP is used, the Home Gateway is referred to as an L2TP network server (LNS).

RADIUS attribute 90 and RADIUS attribute 91 are included in the following situations:

- If the RADIUS server accepts the request and the desired authentication name is different from the default, they must be included it.
- If an accounting request contains Acct-Status-Type attributes with values of either start or stop and pertains to a tunneled session, they should be included in.

Benefits

The RADIUS Tunnel Attribute Extensions feature allows you to specify a name (other than the default) of the tunnel initiator and the tunnel terminator. Thus, you can establish a higher level of security when setting up VPN tunneling.

Restrictions

Your RADIUS server must support tagged attributes to use RADIUS tunnel attributes 90 and 91.

Related Documents

The following documents provide information related to the RADIUS Tunnel Attribute Extensions feature:

- The chapters “Configuring Authentication” and “Configuring RADIUS” in the *Cisco IOS Security Configuration Guide*, Release 12.2
- The appendix “RADIUS Attributes” in the *Cisco IOS Security Configuration Guide*, Release 12.2

- The chapter "Configuring Virtual Private Networks" in the *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2
- RFC 2868, *RADIUS Attributes for Tunnel Protocol Support*

Supported Platforms

Cisco IOS Release 12.1(5)T Only

- AS5300
- AS5800

Cisco IOS Releases 12.2(4)B3 and 12.2(13)T Only

Cisco 6400-NRP-1

Cisco 6400-NRP-2

Cisco 6400-NRP-2SV

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that are supported on specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

- RFC 2868, *RADIUS Attributes for Tunnel Protocol Support*

Prerequisites

To use RADIUS attributes 90 and 91, you must complete the following tasks:

- Configure your NAS to support AAA.
- Configure your NAS to support RADIUS.
- Configure your NAS to support VPN.

Configuration Tasks

None

Verifying RADIUS Attribute 90 and RADIUS Attribute 91

To verify that RADIUS attribute 90 and RADIUS attribute 91 are being sent in access accepts and accounting requests, use the following command in privileged EXEC mode:

Command	Purpose
`Router# debug radius`	Displays information associated with RADIUS. The output of this command shows whether attribute 90 and attribute 91 are being sent in access accepts and accounting requests.

Configuration Examples

This section provides the following configuration examples:

- L2TP Network Server (LNS) Configuration Example
- RADIUS User Profile with RADIUS Tunneling Attributes 90 and 91 Example

L2TP Network Server (LNS) Configuration Example

The following example shows how to configure the LNS with a basic L2F and L2TP configuration using RADIUS tunneling attributes 90 and 91:

```
aaa new-model
aaa authentication login default none
aaa authentication login console none
aaa authentication ppp default local group radius
aaa authorization network default group radius if-authenticated
!
username l2f-cli-auth-id password 0 l2f-cli-pass
username l2f-svr-auth-id password 0 l2f-svr-pass
username l2tp-svr-auth-id password 0 l2tp-tnl-pass
!
vpdn enable
vpdn search-order domain
!
vpdn-group 1
accept-dialin
protocol l2f
virtual-template 1
terminate-from hostname l2f-cli-auth-id
local name l2f-svr-auth-id
!
vpdn-group 2
accept-dialin
protocol l2tp
virtual-template 2
terminate-from hostname l2tp-cli-auth-id
local name l2tp-svr-auth-id
!
interface Ethernet1/0
ip address 10.0.0.3 255.255.255.0
no ip route-cache
no ip mroute-cache
```

```
!
interface Virtual-Template1
ip unnumbered Ethernet1/0
ppp authentication pap
!
interface Virtual-Template2
ip unnumbered Ethernet1/0
ppp authentication pap
!
radius-server host 1.1.1.1 auth-port 1645 acct-port 1646
radius-server key <deleted>
!
```

RADIUS User Profile with RADIUS Tunneling Attributes 90 and 91 Example

The following is an example of a RADIUS user profile that includes RADIUS tunneling attributes 90 and 91. This entry supports two tunnels, one for L2F and the other for L2TP. The tag entries with :1 support L2F tunnels, and the tag entries with :2 support L2TP tunnels.

```
cisco.com Password = "cisco", Service-Type = Outbound
    Service-Type = Outbound,
    Tunnel-Type = :1:L2F,
    Tunnel-Medium-Type = :1:IP,
    Tunnel-Client-Endpoint = :1:"10.0.0.2",
    Tunnel-Server-Endpoint = :1:"10.0.0.3",
    Tunnel-Client-Auth-Id = :1:"l2f-cli-auth-id",
    Tunnel-Server-Auth-Id = :1:"l2f-svr-auth-id",
    Tunnel-Assignment-Id = :1:"l2f-assignment-id",
    Cisco-Avpair = "vpdn:nas-password=l2f-cli-pass",
    Cisco-Avpair = "vpdn:gw-password=l2f-svr-pass",
    Tunnel-Preference = :1:1,
    Tunnel-Type = :2:L2TP,
    Tunnel-Medium-Type = :2:IP,
    Tunnel-Client-Endpoint = :2:"10.0.0.2",
    Tunnel-Server-Endpoint = :2:"10.0.0.3",
    Tunnel-Client-Auth-Id = :2:"l2tp-cli-auth-id",
    Tunnel-Server-Auth-Id = :2:"l2tp-svr-auth-id",
    Tunnel-Assignment-Id = :2:"l2tp-assignment-id",
    Cisco-Avpair = "vpdn:l2tp-tunnel-password=l2tp-tnl-pass",
    Tunnel-Preference = :2:2
```

Command Reference

This feature uses no new or modified commands. To see the command pages for the commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

Glossary

Layer 2 Forwarding (L2F)—A Layer 2 tunneling protocol that enables an ISP or other access service to create a virtual tunnel to link customer remote sites or remote users with corporate home networks. In particular, a network access server (NAS) at the ISP point of presence (POP) exchanges PPP messages with the remote users and communicates by L2F or L2TP requests and responses with the customer tunnel server to set up tunnels.

Layer 2 Tunnel Protocol (L2TP)—A Layer 2 tunneling protocol that enables an ISP or other access service to create a virtual tunnel to link customer remote sites or remote users with corporate home networks. In particular, a network access server (NAS) at the ISP point of presence (POP) exchanges PPP messages with the remote users and communicates by L2F or L2TP requests and responses with the customer tunnel server to set up tunnels.

L2TP access concentrator (LAC)—A network access server (NAS) to which the client directly connects and through which PPP frames are tunneled to the L2TP network server (LNS). The LAC need only implement the media over which L2TP is to operate to pass traffic to one or more LNSs. The LAC may tunnel any protocol carried within PPP. The LAC initiates incoming calls and receives outgoing calls. A LAC is analogous to an L2F network access server.

L2TP network server (LNS)—A termination point for L2TP tunnels, and an access point where PPP frames are processed and passed to higher-layer protocols. An LNS can operate on any platform that terminates PPP. The LNS handles the server side of the L2TP protocol. L2TP relies only on the single medium over which L2TP tunnels arrive. The LNS initiates outgoing calls and receives incoming calls. An LNS is analogous to a home gateway in L2F technology.

network access server (NAS)—A Cisco platform, or collection of platforms, such as an AccessPath system, that interfaces between the packet world (such as the Internet) and the circuit-switched world (such as the PSTN).

tunnel—A virtual pipe between the L2TP access concentrator (LAC) and L2TP network server (LNS) that can carry multiple PPP sessions.

virtual private network (VPN)—A system that permits dial-in networks to exist remotely to home networks, while giving the appearance of being directly connected. VPNs use L2TP and L2F to terminate the Layer 2 and higher parts of the network connection at the L2TP network server (LNS) instead of the L2TP access concentrator (LAC).

V.92 Reporting Using RADIUS Attribute v.92-info

The V.92 Reporting Using RADIUS Attribute v.92-info feature provides the ability to track V.92 call information, such as V.92 features that are supported, the Quick Connect feature set that was attempted, the duration for which the original call was put on hold, and how many times Modem On Hold was initiated. The vendor-specific attribute (VSA) v.92-info is included in accounting "start" and "stop" records when modems negotiate a V.92 connection.

Feature Specifications for the V.92 Reporting Using RADIUS Attribute v.92-info Feature

Feature History	
Release	**Modification**
12.3(1)	This feature was introduced.
Supported Platforms	
Cisco AS5300, Cisco AS5350, Cisco AS5400, Cisco AS5800, Cisco AS5850	

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- Prerequisites for V.92 Reporting Using RADIUS Attribute v.92-info, page 1986
- Restrictions for V.92 Reporting Using RADIUS Attribute v.92-info, page 1986
- Information About V.92 Reporting Using RADIUS Attribute v.92-info, page 1986
- Monitoring V.92 Call Information, page 1987
- Verifying V.92 Call Information, page 1995
- Additional References, page 1999
- Command Reference, page 2000

Prerequisites for V.92 Reporting Using RADIUS Attribute v.92-info

Before the network access server (NAS) can send attribute v.92-info information in accounting "start" and "stop" records, you must perform the following tasks:

- Configure your NAS for authentication, authorization, and accounting (AAA) and to accept incoming modem calls.
- Enable AAA accounting by using the **aaa accounting network default start-stop group radius** command in global configuration mode.
- Familiarize yourself with the V.92 Quick Connect feature. Refer to the following document:
  - *V.92 Quick Connect for Cisco AS5300 and Cisco AS5800 Universal Access Servers*
- Familiarize yourself with the V.92 Modem on Hold feature. Refer to the following document:
  - *V.92 Modem on Hold for Cisco AS5300 and Cisco AS5800 Universal Access Servers*

Restrictions for V.92 Reporting Using RADIUS Attribute v.92-info

- If V.92 is not negotiated on your server, V.92 information will not be included in the accounting record.
- Because the attribute v.92-info information is sent as a Cisco VSA, if you configure your RADIUS server as nonstandard (using a non-Cisco server), the V.92 call information will not be sent by default. However, you can still get the V.92 call information by first configuring the **radius-server vsa send** command with the **accounting** keyword (that is, **radius-server vsa send accounting**).

Information About V.92 Reporting Using RADIUS Attribute v.92-info

Before you use the V.92 Reporting Using RADIUS Attribute v.92-info feature, you must understand the following concepts:

- V.92 Standard Overview, page 1986
- VSA v.92-info, page 1987

V.92 Standard Overview

The International Telecommunication Union Telecommunication Standardization Sector (ITU-T) V.92 standard encompasses a number of specifications, including Quick Connect (QC), which dramatically improves how quickly users can connect with their Internet service provider (ISP), and Modem on Hold (MoH), which enables users to suspend and reactivate their dial-up connection to either receive or initiate a telephone call. V.92 also includes pulse code modulation (PCM) upstream, which boosts the upstream data rates from the user to the ISP to reduce transfer times for large files and e-mail attachments sent by the user.

VSA v.92-info

The VSA v.92-info information in RADIUS accounting "start" and "stop" records can help you track V.92 feature set information. The VSA is enabled by default for all sessions that reside over a modem call that is connected using V.92 model modulation.

The VSA information is displayed in the "start" and "stop" records as follows:

v92-info=<V.92 features supported>/<QC Exchange>/<Total MOH time>/<MOH count>

The VSA v92-info has the following four subfields:

- V.92 features supported—All features that are available for the V.92 modem user who is dialing in. These features include QC, MoH, and PCM Upstream.
- QC Exchange—If QC was initiated, this subfield states what feature set (within QC) was attempted.
- Total MOH time—If MoH was initiated, this subfield indicates the duration for which the original call was put on hold.
- MOH count—If MOH was initiated, this field indicates how many times the MOH was initiated.

The following is an example of VSA v92-info information displayed in an accounting record:

v92-info=V.92 QC MOH/QC Requested/60/1

How to Monitor and Verify V.92 Call Information

The following sections include tasks to help you monitor and verify V.92 call information:

- Monitoring V.92 Call Information, page 1987
- Verifying V.92 Call Information, page 1995

Monitoring V.92 Call Information

To monitor the V.92 information in the accounting "start" and "stop" records, you can perform the following task using some or all of the debug commands that are listed:

SUMMARY

1. **enable**
2. **debug aaa accounting**
3. **debug aaa authentication**
4. **debug aaa authorization**
5. **debug isdn event**
6. **debug modem csm** [*slot/port* | **group** *group-number*]
7. **debug ppp** {**negotiation** | **authentication**}
8. **debug radius**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug aaa accounting** **Example:** `Router# debug aaa accounting`	Displays information about accountable events as they occur.
Step 3	**debug aaa authentication** **Example:** `Router# debug aaa authentication`	Displays information about AAA authentication.
Step 4	**debug aaa authorization** **Example:** `Router# debug aaa authorization`	Displays information about AAA and TACACS+ authorization.
Step 5	**debug isdn event** **Example:** `Router# debug isdn event`	Displays ISDN events occurring on the user side (on the router) of the ISDN interface.
Step 6	**debug modem csm** [*slot/port* | **group** *group-number*] **Example:** `Router# debug modem csm 1/0 group 1`	Displays call switching module (CSM) modem call information.
Step 7	debug ppp {**negotiation** | **authentication**} **Example:** `Router# debug ppp authentication`	Displays information on traffic and exchanges in an internetwork that is implementing the PPP.
Step 8	**debug radius** **Example:** `Router# debug radius`	Displays information associated with RADIUS.

Examples

The following sample debug outputs display information about a V.92 reporting situation:

Debug Output 1

```
01:39:19: ISDN Se7/6:23: RX <-  SETUP pd = 8  callref = 0x42A0
01:39:19:         Bearer Capability i = 0x9090A2
01:39:19:         Channel ID i = 0xA18396
01:39:19:         Progress Ind i = 0x8183 - Origination address is non-ISDN
01:39:19:         Calling Party Number i = 0xA1, '60112', Plan:ISDN, Type:National
```

```
01:39:19:         Called Party Number i = 0xA1, '50138', Plan:ISDN, Type:National
01:39:19:         Locking Shift to Codeset 6
01:39:19:         Codeset 6 IE 0x28  i = 'ANALOG,savitha'
01:39:19: ISDN Se7/6:23: Incoming call id = 0x0038, dsl 0
01:39:19: ISDN Se7/6:23: NegotiateBchan: bchan 22 intid 0 serv_st 0 chan_st 0 callid
0x0000 ev 0x90 n/w? 0
01:39:19: Negotiated int_id 0 bchan 0 cr=0xC2A0 callid=0x0038 lo_chan 22 final
int_id/bchan 0/22 cause 0x0
01:39:19: ISDN Se7/6:23: LIF_EVENT: ces/callid 1/0x38 CALL_INCOMING
01:39:19: ISDN Se7/6:23: CALL_INCOMING dsl 0 bchan 21
01:39:19: voice_parse_intf_name: Using the old NAS_PORT string
01:39:19: AAA/ACCT/EVENT/(00000007): CALL START
01:39:19: AAA/ACCT(00000000): add node, session 9
01:39:19: AAA/ACCT/NET(00000007): add, count 1
01:39:19: AAA/ACCT/EVENT/(00000007): ATTR REPLACE
01:39:19: ISDN Se7/6:23: CALL_INCOMING: call type is VOICE ULAW, bchan = 21
01:39:19: ISDN Se7/6:23: Event:  Received a VOICE call from 60112 on B21 at 64 Kb/s Tone
Value 0
01:39:19: AAA/ACCT/DS0: channel=21, ds1=6, t3=0, slot=7, ds0=117465109
01:39:19: AAA/ACCT/DS0: channel=21, ds1=6, t3=0, slot=7, ds0=117465109
01:39:19: VDEV_ALLOCATE: 1/5 is allocated
01:39:19: ISDN Se7/6:23: RM returned call_type 1 resource type 0 response 2
01:39:19: EVENT_FROM_ISDN: dchan_idb=0x63B3D334, call_id=0x38, ces=0x0
   bchan=0x15, event=0x1, cause=0x0
01:39:19:  dev in call to isdn : set dnis_collected & fap_notify
01:39:19: EVENT_FROM_ISDN:(0038): DEV_INCALL at slot 1 and port 5
01:39:19: EVENT_FROM_ISDN: decode:calling oct3 0xA1, called oct3 0xA1, oct3a 0x0,mask 0x3D
01:39:19: EVENT_FROM_ISDN: csm_call_info:calling oct3 0xA1, called oct3 0xA1, oct3a
0x0,mask 0x3D
01:39:19: CSM_PROC_IDLE: CSM_EVENT_ISDN_CALL at slot 1, port 5
01:39:19: CSM DSPLIB(1/5/csm_flags=0x12): np_dsplib_prepare_modem
01:39:19: csm_connect_pri_vdev: TS allocated at bp_stream 0, bp_Ch 5, vdev_common
0x62EAD8F4 1/5
01:39:19: ISDN Se7/6:23: EVENT to CSM:DEV_INCALL: calltype=VOICE, bchan=21
01:39:19: ISDN Se7/6:23: TX ->  CALL_PROC pd = 8  callref = 0xC2A0
01:39:19:         Channel ID i = 0xA98396
01:39:19: ISDN Se7/6:23: TX ->  ALERTING pd = 8  callref = 0xC2A0
01:39:19: CSM DSPLIB(1/5):DSPLIB_MODEM_INIT: Modem session transition to IDLE
01:39:19: CSM DSPLIB(1/5): Modem went offhook
01:39:19: CSM_PROC_IC2_RING: CSM_EVENT_MODEM_OFFHOOK at slot 1, port 5
01:39:19: ISDN Se7/6:23: VOICE_ANS Event:  call id 0x38, bchan 21, ces 0
01:39:19: ISDN Se7/6:23: isdn_send_connect(): msg 74, call id 0x38, ces 0 bchan 21, call
type VOICE
01:39:19: ISDN Se7/6:23: TX ->  CONNECT pd = 8  callref = 0xC2A0
01:39:19: ISDN Se7/6:23: RX <-  CONNECT_ACK pd = 8  callref = 0x42A0
01:39:19: ISDN Se7/6:23: LIF_EVENT: ces/callid 1/0x38 CALL_PROGRESS
01:39:19: ISDN Se7/6:23: event CALL_PROGRESS dsl 0
01:39:19: ISDN Se7/6:23: CALL_PROGRESS: CALL_CONNECTED call id 0x38, bchan 21, dsl 0
01:39:19: EVENT_FROM_ISDN: dchan_idb=0x63B3D334, call_id=0x38, ces=0x0
   bchan=0x15, event=0x4, cause=0x0
01:39:19: EVENT_FROM_ISDN:(0038): DEV_CONNECTED at slot 1 and port 5
01:39:19: CSM_PROC_IC6_WAIT_FOR_CONNECT: CSM_EVENT_ISDN_CONNECTED at slot 1, port 5
01:39:19: CSM DSPLIB(1/5): np_dsplib_call_accept
01:39:19: ISDN Se7/6:23: EVENT to CSM:DEV_CONNECTED: calltype=VOICE, bchan=21
01:39:19: CSM DSPLIB(1/5):DSPLIB_MODEM_WAIT_ACTIVE: Modem session transition to ACTIVE
01:39:19: CSM DSPLIB(1/5): Modem state changed to (CONNECT_STATE)
01:39:22: CSM DSPLIB(1/5): Modem state changed to (V8BIS_EXCHANGE_STATE)
01:39:24: CSM DSPLIB(1/5): Modem state changed to (LINK_STATE)
01:39:28: CSM DSPLIB(1/5): Modem state changed to (RANGING_STATE)
01:39:30: CSM DSPLIB(1/5): Modem state changed to (HALF_DUPLEX_TRAIN_STATE)
01:39:45: CSM DSPLIB(1/5): Modem state changed to (TRAINUP_STATE)
01:39:45: CSM DSPLIB(1/5): Modem state changed to (EC_NEGOTIATING_STATE)
01:39:46: CSM DSPLIB(1/5): Modem state changed to (STEADY_STATE)
01:39:46: TTY1/05: DSR came up
```

```
01:39:46: tty1/05: Modem: IDLE->(unknown)
01:39:46: TTY1/05: EXEC creation
01:39:46: CHAT1/05: Attempting line activation script
01:39:46: CHAT1/05: Asserting DTR
01:39:50: voice_parse_intf_name: Using the old NAS_PORT string
01:39:50: voice_parse_intf_name: Using the old NAS_PORT string
01:39:50: AAA/AUTHEN/LOGIN (00000007): Pick method list 'default'
01:39:50: RADIUS/ENCODE(00000007): ask "Username: "
01:39:50: RADIUS/ENCODE(00000007): send packet; GET_USER
01:39:50: TTY1/05: set timer type 10, 30 seconds
01:39:50: TTY1/05: Autoselect(2) sample 7E
01:39:50: TTY1/05: Autoselect(2) sample 7EFF
01:39:50: TTY1/05: Autoselect(2) sample 7EFF7D
01:39:50: TTY1/05: Autoselect(2) sample 7EFF7D23
01:39:50: TTY1/05 Autoselect cmd:  ppp negotiate
01:39:50: TTY1/05: EXEC creation
01:39:50: CHAT1/05: Attempting line activation script
01:39:50: CHAT1/05: Asserting DTR
01:39:54: voice_parse_intf_name: Using the old NAS_PORT string
01:39:54: voice_parse_intf_name: Using the old NAS_PORT string
01:39:54: TTY1/05: no timer type 1 to destroy
01:39:54: TTY1/05: no timer type 0 to destroy
01:39:54: As1/05 LCP: I CONFREQ [Closed] id 0 len 50
01:39:54: As1/05 LCP:    ACCM 0x00000000 (0x020600000000)
01:39:54: As1/05 LCP:    MagicNumber 0x00002EB8 (0x050600002EB8)
01:39:54: As1/05 LCP:    PFC (0x0702)
01:39:54: As1/05 LCP:    ACFC (0x0802)
01:39:54: As1/05 LCP:    Callback 6  (0x0D0306)
01:39:54: As1/05 LCP:    MRRU 1614 (0x1104064E)
01:39:54: As1/05 LCP:    EndpointDisc 1 Local
01:39:54: As1/05 LCP:     (0x131701CC7F60A0E7A211D6B549000102)
01:39:54: As1/05 LCP:     (0x2BC43900000000)
01:39:54: As1/05 LCP: Lower layer not up, Fast Starting
01:39:54: voice_parse_intf_name: Using the old NAS_PORT string
01:39:54: voice_parse_intf_name: Using the old NAS_PORT string
01:39:54: As1/05 PPP: Treating connection as a callin
01:39:54: As1/05 PPP: Phase is ESTABLISHING, Passive Open
01:39:54: As1/05 LCP: State is Listen
01:39:54: As1/05 PPP: Authorization required
01:39:54: As1/05 LCP: O CONFREQ [Listen] id 1 len 25
01:39:54: As1/05 LCP:    ACCM 0x000A0000 (0x0206000A0000)
01:39:54: As1/05 LCP:    AuthProto CHAP (0x0305C22305)
01:39:54: As1/05 LCP:    MagicNumber 0x099EBCBA (0x0506099EBCBA)
01:39:54: As1/05 LCP:    PFC (0x0702)
01:39:54: As1/05 LCP:    ACFC (0x0802)
01:39:54: As1/05 LCP: O CONFREJ [Listen] id 0 len 11
01:39:54: As1/05 LCP:    Callback 6  (0x0D0306)
01:39:54: As1/05 LCP:    MRRU 1614 (0x1104064E)
01:39:54: As1/05 LCP: I CONFACK [REQsent] id 1 len 25
01:39:54: As1/05 LCP:    ACCM 0x000A0000 (0x0206000A0000)
01:39:54: As1/05 LCP:    AuthProto CHAP (0x0305C22305)
01:39:54: As1/05 LCP:    MagicNumber 0x099EBCBA (0x0506099EBCBA)
01:39:54: As1/05 LCP:    PFC (0x0702)
01:39:54: As1/05 LCP:    ACFC (0x0802)
01:39:54: As1/05 LCP: I CONFREQ [ACKrcvd] id 1 len 43
01:39:54: As1/05 LCP:    ACCM 0x00000000 (0x020600000000)
01:39:54: As1/05 LCP:    MagicNumber 0x00002EB8 (0x050600002EB8)
01:39:54: As1/05 LCP:    PFC (0x0702)
01:39:54: As1/05 LCP:    ACFC (0x0802)
01:39:54: As1/05 LCP:    EndpointDisc 1 Local
01:39:54: As1/05 LCP:     (0x131701CC7F60A0E7A211D6B549000102)
01:39:54: As1/05 LCP:     (0x2BC43900000000)
01:39:54: As1/05 LCP: O CONFACK [ACKrcvd] id 1 len 43
01:39:54: As1/05 LCP:    ACCM 0x00000000 (0x020600000000)
```

```
01:39:54: As1/05 LCP:    MagicNumber 0x00002EB8 (0x050600002EB8)
01:39:54: As1/05 LCP:    PFC (0x0702)
01:39:54: As1/05 LCP:    ACFC (0x0802)
01:39:54: As1/05 LCP:    EndpointDisc 1 Local
01:39:54: As1/05 LCP:     (0x131701CC7F60A0E7A211D6B549000102)
01:39:54: As1/05 LCP:     (0x2BC43900000000)
01:39:54: As1/05 LCP: State is Open
01:39:54: As1/05 PPP: Phase is AUTHENTICATING, by this end
01:39:54: As1/05 CHAP: O CHALLENGE id 1 len 26 from "s5400"
01:39:54: As1/05 LCP: I IDENTIFY [Open] id 2 len 18 magic 0x00002EB8 MSRASV4.00
01:39:54: As1/05 LCP: I IDENTIFY [Open] id 3 len 23 magic 0x00002EB8 MSRAS-1-PTE-PC1
01:39:54: As1/05 CHAP: I RESPONSE id 1 len 34 from "Administrator"
01:39:54: As1/05 PPP: Phase is FORWARDING, Attempting Forward
01:39:54: As1/05 PPP: Phase is AUTHENTICATING, Unauthenticated User
01:39:54: AAA/AUTHEN/PPP (00000007): Pick method list 'default'
01:39:54: As1/05 PPP: Sent CHAP LOGIN Request
01:39:54: RADIUS/ENCODE(00000007): Unsupported AAA attribute parent-interface
01:39:54: RADIUS/ENCODE(00000007): Unsupported AAA attribute parent-interface-type
01:39:54: RADIUS/ENCODE(00000007): acct_session_id: 9
01:39:54: RADIUS(00000007): sending
01:39:54: RADIUS: Send to unknown id 2 10.107.164.120:1645, Access-Request, len 128
01:39:54: RADIUS:  authenticator 13 E4 F2 9F BC 3E CE 52 - CC 93 0C E0 01 0C 73 7B
01:39:54: RADIUS:  Framed-Protocol     [7]   6   PPP                       [1]
01:39:54: RADIUS:  User-Name           [1]   15  "Administrator"
01:39:54: RADIUS:  CHAP-Password       [3]   19  *
01:39:54: RADIUS:  Called-Station-Id   [30]  7   "50138"
01:39:54: RADIUS:  Calling-Station-Id  [31]  7   "60112"
01:39:54: RADIUS:  Vendor, Cisco       [26]  30
01:39:54: RADIUS:   cisco-nas-port     [2]   24  "Async1/05*Serial7/6:21"
01:39:54: RADIUS:  NAS-Port            [5]   6   221
01:39:54: RADIUS:  NAS-Port-Type       [61]  6   Async                      [0]
01:39:54: RADIUS:  Service-Type        [6]   6   Framed                     [2]
01:39:54: RADIUS:  NAS-IP-Address      [4]   6   10.0.58.107
01:39:54: RADIUS: Received from id 2 10.107.164.120:1645, Access-Accept, len 62
01:39:54: RADIUS:  authenticator EF 45 A3 D4 A7 EE D0 65 - 03 50 B4 3E 07 87 2E 2F
01:39:54: RADIUS:  Vendor, Cisco       [26]  30
01:39:54: RADIUS:   cisco-nas-port     [2]   24  "Async1/05*Serial7/6:21"
01:39:54: RADIUS:  Service-Type        [6]   6   Framed                     [2]
01:39:54: RADIUS:  Framed-Protocol     [7]   6   PPP                       [1]
01:39:54: RADIUS: Received from id 7
01:39:54: As1/05 PPP: Received LOGIN Response PASS
01:39:54: As1/05 PPP/AAA: Check Attr: interface
01:39:54: As1/05 PPP/AAA: Check Attr: service-type
01:39:54: As1/05 PPP/AAA: Check Attr: Framed-Protocol
01:39:54: As1/05 PPP: Phase is FORWARDING, Attempting Forward
01:39:54: As1/05 PPP: Phase is AUTHENTICATING, Authenticated User
01:39:54: As1/05 AAA/AUTHOR/LCP: Process Author
01:39:54: As1/05 AAA/AUTHOR/LCP: Process Attr: service-type
01:39:54: As1/05 CHAP: O SUCCESS id 1 len 4
01:39:54: AAA/ACCT/NET(00000007): Pick method list 'default'
01:39:54: AAA/ACCT/SETMLIST(00000007): Handle FFFFFFFF, mlist 630B11E4, Name default
01:39:54: AAA/ACCT/EVENT/(00000007): NET UP
01:39:54: AAA/ACCT/NET(00000007): Queueing record is START
01:39:54: As1/05 PPP: Phase is UP
01:39:54: As1/05 AAA/AUTHOR/IPCP: FSM authorization not needed
01:39:54: As1/05 AAA/AUTHOR/FSM: We can start IPCP
01:39:54: As1/05 IPCP: O CONFREQ [Closed] id 1 len 10
01:39:54: As1/05 IPCP:    Address 10.1.1.2 (0x030646010102)
01:39:54: AAA/ACCT(00000007): Accouting method=radius (radius)
01:39:54: RADIUS/ENCODE(00000007): Unsupported AAA attribute timezone
01:39:54: RADIUS/ENCODE(00000007): Unsupported AAA attribute parent-interface
01:39:54: RADIUS/ENCODE(00000007): Unsupported AAA attribute parent-interface-type
01:39:54: RADIUS(00000007): sending
01:39:54: RADIUS: Send to unknown id 8 10.107.164.120:1646, Accounting-Request, len 243
```

```
01:39:54: RADIUS:  authenticator 41 87 FA 03 EB F9 94 62 - B2 3A 24 B8 27 4C A4 BC
01:39:54: RADIUS:  Acct-Session-Id     [44]  10  "00000009"
01:39:54: RADIUS:  Framed-Protocol     [7]   6   PPP                         [1]
01:39:54: RADIUS:  Connect-Info        [77]  26  "52000/28800 V90/V44/LAPM"
01:39:54: RADIUS:  Vendor, Cisco       [26]  48
01:39:54: RADIUS:   Cisco AVpair       [1]   42  "v92-info=V.92 QC MOH/No QC
Requested/0/0"
01:39:54: RADIUS:  Vendor, Cisco       [26]  32
01:39:54: RADIUS:   Cisco AVpair       [1]   26  "connect-progress=Call Up"
01:39:54: RADIUS:  Authentic           [45]  6   RADIUS                      [1]
01:39:54: RADIUS:  User-Name           [1]   15  "Administrator"
01:39:54: RADIUS:  Acct-Status-Type    [40]  6   Start                       [1]
01:39:54: RADIUS:  Called-Station-Id   [30]  7   "50138"
01:39:54: RADIUS:  Calling-Station-Id  [31]  7   "60112"
01:39:54: RADIUS:  Vendor, Cisco       [26]  30
01:39:54: RADIUS:   cisco-nas-port     [2]   24  "Async1/05*Serial7/6:21"
01:39:54: RADIUS:  NAS-Port            [5]   6   221
01:39:54: RADIUS:  NAS-Port-Type       [61]  6   Async                       [0]
01:39:54: RADIUS:  Service-Type        [6]   6   Framed                      [2]
01:39:54: RADIUS:  NAS-IP-Address      [4]   6   10.0.58.107
01:39:54: RADIUS:  Acct-Delay-Time     [41]  6   0
01:39:54: RADIUS: Received from id 8 10.107.164.120:1646, Accounting-response, len 20
01:39:54: RADIUS:  authenticator E5 5C D3 69 88 D5 2E 8E - 49 AF 63 22 01 53 33 7B
01:39:54: AAA/ACCT/NET(00000007): START protocol reply PASS
01:39:54: As1/05 CCP: I CONFREQ [Not negotiated] id 4 len 211
01:39:54: As1/05 CCP:    Type254
01:39:54: As1/05 CCP:     (0xFEC901000000000000000000000000000000)
01:39:54: As1/05 CCP:     (0x000074FFC7000000000068000000A000)
01:39:54: As1/05 CCP:     (0x00006C20563905000000C0000000A400)
01:39:54: As1/05 CCP:     (0x0000BC0000001864000007000E80018C8)
01:39:54: As1/05 CCP:     (0x130017CCF17700000000001000000E8FE)
01:39:54: As1/05 CCP:     (0xC70076CDF17706000000000000000000)
01:39:54: As1/05 CCP:     (0x00000000000000000000000000000000)
01:39:54: As1/05 CCP:     (0x00000000000000000000000000000000)
01:39:54: As1/05 CCP:     (0x00000000000000000000000000000000)
01:39:54: As1/05 CCP:     (0x00000000000000000000220020000001)
01:39:54: As1/05 CCP:     (0x8000000000005016B1CBA2E7D611B549)
01:39:54: As1/05 CCP:     (0x0001022BC439C800000000000000C800)
01:39:54: As1/05 CCP:     (0x00004D000000281FB8)
01:39:54: As1/05 CCP:    MS-PPC supported bits 0x00000006 (0x120600000006)
01:39:54: As1/05 LCP: O PROTREJ [Open] id 2 len 217 protocol CCP
01:39:54: As1/05 LCP:  (0x80FD010400D3FEC90100000000000000)
01:39:54: As1/05 LCP:  (0x00000000000000074FFC70000000000)
01:39:54: As1/05 LCP:  (0x68000000A00000006C20563905000000)
01:39:54: As1/05 LCP:  (0xC0000000A4000000BC00000018640000)
01:39:54: As1/05 LCP:  (0x7000E80018C8130017CCF17700000000)
01:39:54: As1/05 LCP:  (0x01000000E8FEC70076CDF17706000000)
01:39:54: As1/05 LCP:  (0x00000000000000000000000000000000)
01:39:54: As1/05 LCP:  (0x00000000000000000000000000000000)
01:39:54: As1/05 LCP:  (0x00000000000000000000000000000000)
01:39:54: As1/05 LCP:  (0x00000000000000000000000000000000)
01:39:54: As1/05 LCP:  (0x2200200000010800000000005016B1CB)
01:39:54: As1/05 LCP:  (0xA2E7D611B5490001022BC439C8000000)
01:39:54: As1/05 LCP:  (0x00000000C80000004D000000281FB812)
01:39:54: As1/05 LCP:  (0x0600000006)
01:39:54: As1/05 IPCP: I CONFREQ [REQsent] id 5 len 34
01:39:54: As1/05 IPCP:    Address 0.0.0.0 (0x030600000000)
01:39:54: As1/05 IPCP:    PrimaryDNS 0.0.0.0 (0x810600000000)
01:39:54: As1/05 IPCP:    PrimaryWINS 0.0.0.0 (0x820600000000)
01:39:54: As1/05 IPCP:    SecondaryDNS 0.0.0.0 (0x830600000000)
01:39:54: As1/05 IPCP:    SecondaryWINS 0.0.0.0 (0x840600000000)
01:39:54: As1/05 AAA/AUTHOR/IPCP: Start.  Her address 0.0.0.0, we want 10.2.2.6
01:39:54: As1/05 AAA/AUTHOR/IPCP: Authorization succeeded
01:39:54: As1/05 AAA/AUTHOR/IPCP: Done.  Her address 0.0.0.0, we want 10.2.2.6
```

```
01:39:54: As1/05 AAA/AUTHOR/IPCP: no author-info for primary dns
01:39:54: As1/05 AAA/AUTHOR/IPCP: no author-info for primary wins
01:39:54: As1/05 AAA/AUTHOR/IPCP: no author-info for seconday dns
01:39:54: As1/05 AAA/AUTHOR/IPCP: no author-info for seconday wins
01:39:54: As1/05 IPCP: O CONFREJ [REQsent] id 5 len 28
01:39:54: As1/05 IPCP:    PrimaryDNS 0.0.0.0 (0x8106000000000)
01:39:54: As1/05 IPCP:    PrimaryWINS 0.0.0.0 (0x8206000000000)
01:39:54: As1/05 IPCP:    SecondaryDNS 0.0.0.0 (0x8306000000000)
01:39:54: As1/05 IPCP:    SecondaryWINS 0.0.0.0 (0x8406000000000)
01:39:54: As1/05 IPCP: I CONFACK [REQsent] id 1 len 10
01:39:54: As1/05 IPCP:    Address 70.1.1.2 (0x030646010102)
01:39:54: As1/05 IPCP: I CONFREQ [ACKrcvd] id 6 len 10
01:39:54: As1/05 IPCP:    Address 0.0.0.0 (0x030600000000)
01:39:54: As1/05 IPCP: O CONFNAK [ACKrcvd] id 6 len 10
01:39:54: As1/05 IPCP:    Address 70.2.2.6 (0x030646020206)
01:39:55: As1/05 IPCP: I CONFREQ [ACKrcvd] id 7 len 10
01:39:55: As1/05 IPCP:    Address 70.2.2.6 (0x030646020206)
01:39:55: As1/05 IPCP: O CONFACK [ACKrcvd] id 7 len 10
01:39:55: As1/05 IPCP:    Address 70.2.2.6 (0x030646020206)
01:39:55: As1/05 IPCP: State is Open
01:39:55: AAA/ACCT/EVENT/(00000007): IPCP_PASS
01:39:55: As1/05 IPCP: Install route to 10.2.2.6
01:39:55: As1/05 IPCP: Add link info for cef entry 10.2.2.6
```

Debug Output 2

```
01:40:50: ISDN Se7/6:23: RX <-  DISCONNECT pd = 8  callref = 0x42A0
01:40:50:         Cause i = 0x8190 - Normal call clearing
01:40:50: ISDN Se7/6:23: LIF_EVENT: ces/callid 1/0x38 CALL_DISC
01:40:50: EVENT_FROM_ISDN: dchan_idb=0x63B3D334, call_id=0x38, ces=0x0
   bchan=0x15, event=0x0, cause=0x10
01:40:50: EVENT_FROM_ISDN:(0038): DEV_IDLE at slot 1 and port 5
01:40:50: CSM_PROC_IC7_OC6_CONNECTED: CSM_EVENT_ISDN_DISCONNECTED at slot 1, port 5
01:40:50: CSM DSPLIB(1/5): np_dsplib_call_hangup reason 14
01:40:50: CSM(1/5): Enter csm_enter_disconnecting_state
01:40:50: VDEV_DEALLOCATE: slot 1 and port 5 is deallocated

01:40:50: ISDN Se7/6:23: EVENT to CSM:DEV_IDLE: calltype=VOICE, bchan=21
01:40:50: ISDN Se7/6:23: process_disc_ack(): call id 0x38, ces 0, call type VOICE cause
0x10
01:40:50: ISDN Se7/6:23: TX ->  RELEASE pd = 8  callref = 0xC2A0
01:40:50: AAA/ACCT/EVENT/(00000007): CALL STOP
01:40:50: AAA/ACCT/CALL STOP(00000007): Sending stop requests
01:40:50: AAA/ACCT(00000007): Send all stops
01:40:50: AAA/ACCT/NET(00000007): STOP
01:40:50: AAA/ACCT/NET(00000007): Queueing record is STOP osr 1
01:40:50: AAA/ACCT(00000007): Accouting method=radius (radius)
01:40:50: RADIUS/ENCODE(00000007): Unsupported AAA attribute timezone
01:40:50: RADIUS/ENCODE(00000007): Unsupported AAA attribute parent-interface
01:40:50: RADIUS/ENCODE(00000007): Unsupported AAA attribute parent-interface-type
01:40:50: RADIUS(00000007): sending
01:40:50: RADIUS: Send to unknown id 9 10.107.164.120:1646, Accounting-Request, len 315
01:40:50: RADIUS:  authenticator 2E 6A 04 D0 04 9A D3 D5 - F7 DD 99 E0 C3 99 27 60
01:40:50: RADIUS:  Acct-Session-Id     [44]  10  "00000009"
01:40:50: RADIUS:  Framed-Protocol     [7]   6   PPP                       [1]
01:40:50: RADIUS:  Framed-IP-Address   [8]   6   70.2.2.6
01:40:50: RADIUS:  Acct-Terminate-Cause[49]  6   lost-carrier              [2]
01:40:50: RADIUS:  Vendor, Cisco       [26]  33
01:40:50: RADIUS:   Cisco AVpair       [1]   27  "disc-cause-ext=No Carrier"
01:40:50: RADIUS:  Vendor, Cisco       [26]  35
01:40:50: RADIUS:   Cisco AVpair       [1]   29  "connect-progress=LAN Ses Up"
01:40:50: RADIUS:  Acct-Session-Time   [46]  6   56
01:40:50: RADIUS:  Connect-Info        [77]  26  "52000/28800 V90/V44/LAPM"
01:40:50: RADIUS:  Vendor, Cisco       [26]  48
```

```
01:40:50: RADIUS:   Cisco AVpair       [1]   42  "v92-info=V.92 QC MOH/No QC
Requested/0/0"
01:40:50: RADIUS:  Acct-Input-Octets   [42]  6   285
01:40:50: RADIUS:  Acct-Output-Octets  [43]  6   295
01:40:50: RADIUS:  Acct-Input-Packets  [47]  6   5
01:40:50: RADIUS:  Acct-Output-Packets [48]  6   5
01:40:50: RADIUS:  User-Name           [1]   15  "Administrator"
01:40:50: RADIUS:  Acct-Status-Type    [40]  6   Stop                      [2]
01:40:50: RADIUS:  Called-Station-Id   [30]  7   "50138"
01:40:50: RADIUS:  Calling-Station-Id  [31]  7   "60112"
01:40:50: RADIUS:  Vendor, Cisco       [26]  30
01:40:50: RADIUS:   cisco-nas-port     [2]   24  "Async1/05*Serial7/6:21"
01:40:50: RADIUS:  NAS-Port            [5]   6   221
01:40:50: RADIUS:  NAS-Port-Type       [61]  6   Async                     [0]
01:40:50: RADIUS:  Service-Type        [6]   6   Framed                    [2]
01:40:50: RADIUS:  NAS-IP-Address      [4]   6   10.0.58.107
01:40:50: RADIUS:  Acct-Delay-Time     [41]  6   0
01:40:50: RADIUS: Received from id 9 10.107.164.120:1646, Accounting-response, len 20
01:40:50: RADIUS:  authenticator D0 3F 32 D7 7C 8C 5E 22 - 9A 69 EF 17 AC 32 81 21
01:40:50: AAA/ACCT/NET(00000007): STOP protocol reply PASS
01:40:50: AAA/ACCT/NET(00000007): Cleaning up from Callback osr 0
01:40:50: AAA/ACCT(00000007): del node, session 9
01:40:50: AAA/ACCT/NET(00000007): free_rec, count 0
01:40:50: AAA/ACCT/NET(00000007) reccnt 0, csr TRUE, osr 0
01:40:50: AAA/ACCT/NET(00000007): Last rec in db, intf not enqueued
01:40:50: ISDN Se7/6:23: RX <-  RELEASE_COMP pd = 8  callref = 0x42A0
01:40:50: ISDN Se7/6:23: CCPRI_ReleaseCall(): bchan 22, call id 0x38, call type VOICE
01:40:50: CCPRI_ReleaseChan released b_dsl 0 B_Chan 22
01:40:50: ISDN Se7/6:23: LIF_EVENT: ces/callid 1/0x38 CALL_CLEARED
01:40:50: ISDN Se7/6:23: received CALL_CLEARED  call_id 0x38
01:40:50: no resend setup, no redial
01:40:50: no resend setup, no redial
01:40:50: AAA/ACCT/DS0: channel=21, ds1=6, t3=0, slot=7, ds0=117465109
01:40:50: EVENT_FROM_ISDN: dchan_idb=0x63B3D334, call_id=0x38, ces=0x1
   bchan=0x15, event=0x0, cause=0x0
01:40:50: ISDN Se7/6:23: EVENT to CSM:DEV_IDLE: calltype=VOICE, bchan=21
01:40:51: CSM DSPLIB(1/5): Modem state changed to (TERMINATING_STATE)
01:40:51: CSM DSPLIB(1/5): Modem went onhook
01:40:51: CSM_PROC_IC8_OC8_DISCONNECTING: CSM_EVENT_MODEM_ONHOOK at slot 1, port 5
01:40:51: CSM(1/5): Enter csm_enter_idle_state
01:40:51: CSM DSPLIB(1/5):DSPLIB_IDLE: Modem session transition to FLUSHING
01:40:51: CSM DSPLIB(1/5):DSPLIB_IDLE: Modem session transition to IDLE
01:40:51: TTY1/05: DSR was dropped
01:40:51: tty1/05: Modem: READY->(unknown)
01:40:52: TTY1/05: dropping DTR, hanging up
01:40:52: DSPLIB(1/5): np_dsplib_process_dtr_notify()
01:40:52: CSM DSPLIB(1/5): Modem went onhook
01:40:52: CSM_PROC_IDLE: CSM_EVENT_MODEM_ONHOOK at slot 1, port 5
01:40:52: TTY1/05: Async Int reset: Dropping DTR
01:40:52: tty1/05: Modem: HANGUP->(unknown)
01:40:52: AAA/ACCT/EVENT/(00000007): NET DOWN
01:40:52: As1/05 IPCP: Remove link info for cef entry 70.2.2.6
01:40:52: As1/05 IPCP: State is Closed
01:40:52: As1/05 PPP: Phase is TERMINATING
01:40:52: As1/05 LCP: State is Closed
01:40:52: As1/05 PPP: Phase is DOWN
01:40:52: As1/05 IPCP: Remove route to 70.2.2.6
01:40:52: As1/05 LCP: State is Closed
01:40:53: TTY1/05: cleanup pending. Delaying DTR
01:40:54: TTY1/05: cleanup pending. Delaying DTR
01:40:55: TTY1/05: cleanup pending. Delaying DTR
01:40:56: TTY1/05: cleanup pending. Delaying DTR
01:40:57: TTY1/05: no timer type 0 to destroy
01:40:57: TTY1/05: no timer type 1 to destroy
```

```
01:40:57: TTY1/05: no timer type 3 to destroy
01:40:57: TTY1/05: no timer type 4 to destroy
01:40:57: TTY1/05: no timer type 2 to destroy
01:40:57: Async1/05: allowing modem_process to continue hangup
01:40:57: TTY1/05: restoring DTR
01:40:57: TTY1/05: autoconfigure probe started
01:40:57: As1/05 LCP: State is Closed
```

Verifying V.92 Call Information

To verify that the V.92 call was correctly established, use the following **show** commands:

SUMMARY

- **show modem** [*slot*/*port* | **group** *number*]
- **show port modem log** [**reverse** *slot*/*port*] [*slot* | *slot*/*port*]
- **show users** [**all**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show modem** [*slot*/*port* | **group** *number*] **Example:** `Router# show modem 1/0 group 1`	Displays a high-level performance report for all the modems or a single modem inside Cisco access servers.
Step 2	**show port modem log** [**reverse** *slot*/*port*] [*slot* | *slot*/*port*] **Example:** `Router# show port modem log`	Displays the events generated by the modem sessions.
Step 3	**show users** [**all**] **Example:** `Router# show users`	Displays information about the active lines on the router.

Examples

The following V.92 reporting outputs are from the **show port modem log** and **show users** commands:

Show Output 1

```
Router# show port modem log 1/05

Port 1/05 Events Log
  01:46:19: Service Type: DATA_FAX_MODEM
  01:46:19: Service Mode: DATA_FAX_MODEM
  01:46:19: Session State: IDLE
  01:46:19: incoming caller number: 60112
  01:46:19: incoming called number: 50138
  01:46:19: Service Type: DATA_FAX_MODEM
  01:46:19: Service Mode: DATA_FAX_MODEM
```

```
  01:46:19: Session State: IDLE
  01:46:19: Service Type: DATA_FAX_MODEM
  01:46:19: Service Mode: DATA_FAX_MODEM
  01:46:19: Session State: ACTIVE
  01:46:19: Modem State event:
            State: Connect
  01:46:20: Modem State event:
            State: V.8bis Exchange
  01:46:20: Modem State event:
            State: Link
  01:46:20: Modem State event:
            State: Ranging
  01:46:20: Modem State event:
            State: Half Duplex Train
  01:46:20: Modem State event:
            State: Train Up
  01:46:20: Modem State event:
            State: EC Negotiating
  01:46:20: Modem State event:
            State: Steady
  01:46:20: Modem Static event:
    Connect Protocol                        :   LAP-M
    Compression                             :   V.44
    Connected Standard                      :   V.90
    TX,RX Symbol Rate                       :   8000, 3200
    TX,RX Carrier Frequency                 :   0, 1829
    TX,RX Trellis Coding                    :   16/No trellis
    Frequency Offset                        :   0  Hz
    Round Trip Delay                        :   0  msecs
    TX,RX Bit Rate                          :   52000, 28800
    Robbed Bit Signalling (RBS) pattern     :   255
    Digital Pad                             :   6 dB
    Digital Pad Compensation                :   Enabled
    MNP10EC                                 :   Off-None
    QC Exchange                             :   No QC Requested
    TX,RX Negotiated String Length          :   255, 255
    DC TX,RX Negotiated Codewords           :   1024, 1024
    DC TX,RX Negotiated History Size        :   4096, 5120
01:46:21: ISDN Se7/6:23: RX <-  SERVICE pd = 3  callref = 0x0000
01:46:21:         Change Status i = 0xC0 - in-service
01:46:21:         Channel ID i = 0xA98381
01:46:21: ISDN Se7/6:23: Incoming call id = 0x003A, dsl 0
01:46:21: ISDN Se7/6:23: LIF_EVENT: ces/callid 1/0x0 CHAN_STATUS
01:46:21: ISDN Se7/6:23: CHAN_STATUS B-chan=1, action=2; Maintenance.
01:46:21: ISDN Se7/6:23: TX ->  SERVICE ACKNOWLEDGE pd = 3  callref = 0x8000
01:46:21:         Change Status i = 0xC0 - in-service
01:46:21:         Channel ID i =        1
s5400#sh port modem log 1/05
Port 1/05 Events Log
  01:46:30: Service Type: DATA_FAX_MODEM
  01:46:30: Service Mode: DATA_FAX_MODEM
  01:46:30: Session State: IDLE
  01:46:30: incoming caller number: 60112
  01:46:30: incoming called number: 50138
  01:46:30: Service Type: DATA_FAX_MODEM
  01:46:30: Service Mode: DATA_FAX_MODEM
  01:46:30: Session State: IDLE
  01:46:30: Service Type: DATA_FAX_MODEM
  01:46:30: Service Mode: DATA_FAX_MODEM
  01:46:30: Session State: ACTIVE
  01:46:30: Modem State event:
            State: Connect
  01:46:30: Modem State event:
            State: V.8bis Exchange
```

```
01:46:30: Modem State event:
          State: Link
01:46:30: Modem State event:
          State: Ranging
01:46:30: Modem State event:
          State: Half Duplex Train
01:46:30: Modem State event:
          State: Train Up
01:46:31: Modem State event:
          State: EC Negotiating
01:46:31: Modem State event:
          State: Steady
01:46:31: Modem Static event:
  Connect Protocol                          :   LAP-M
  Compression                               :   V.44
  Connected Standard                        :   V.90
  TX,RX Symbol Rate                         :   8000, 3200
  TX,RX Carrier Frequency                   :   0, 1829
  TX,RX Trellis Coding                      :   16/No trellis
  Frequency Offset                          :   0  Hz
  Round Trip Delay                          :   0  msecs
  TX,RX Bit Rate                            :   52000, 28800
  Robbed Bit Signalling (RBS) pattern       :   255
  Digital Pad                               :   6 dB
  Digital Pad Compensation                  :   Enabled
  MNP10EC                                   :   Off-None
  QC Exchange                               :   No QC Requested
  TX,RX Negotiated String Length            :   255, 255
  DC TX,RX Negotiated Codewords             :   1024, 1024
  DC TX,RX Negotiated History Size          :   4096, 5120
  Diagnostic Code                           :   00 00 00 00 00 00 00 00
  V.92 Status                               :   V.92 QC MOH
01:46:32: Modem Dynamic event:
  Sq Value                                  :   6
  Signal Noise Ratio                        :   38  dB
  Receive Level                             :   -11  dBm
  Phase Jitter Frequency                    :   0  Hz
  Phase Jitter Level                        :   0  degrees
  Far End Echo Level                        :   0  dBm
  Phase Roll                                :   0  degrees
  Total Retrains                            :   0
  EC Retransmission Count                   :   0
  Characters transmitted, received          :   0, 0
  Characters received BAD                   :   0
  PPP/SLIP packets transmitted, received    :   0, 0
  PPP/SLIP packets received (BAD/ABORTED)   :   0
  EC packets transmitted, received OK       :   0, 0
  EC packets (Received BAD/ABORTED)         :   0
  Total Speedshifts                         :   0
  Total MOH Time                            :   0  secs
  Current MOH Time                          :   0  secs
  MOH Status                                :   Modem is Not on Hold
  MOH Count                                 :   0
  MOH Request Count                         :   0
  Retrains due to Call Waiting              :   0
  DC Encoder,Decoder State                  :   compressed/compressed
  DC TX,RX Compression Ratio                :   not calculated/not calculated
  DC TX,RX Dictionary Reset Count           :   0, 0
  Diagnostic Code                           :   00 00 00 00 00 00 00 00
01:46:35: Modem State event:
          State: Terminate
01:46:35: Service Type: DATA_FAX_MODEM
01:46:35: Service Mode: DATA_FAX_MODEM
01:46:35: Session State: FLUSHING
```

```
01:46:35: Service Type: DATA_FAX_MODEM
01:46:35: Service Mode: DATA_FAX_MODEM
01:46:35: Session State: IDLE
01:46:35: Modem End Connect event:
  Call Timer                              :   65  secs
  Disconnect Reason Info                  :   0x220
      Type (=0  ):  <unknown>
     Class (=2  ):  EC condition - locally detected
    Reason (=32 ):  received DISC frame -- normal LAPM termination
  Total Retrains                          :   0
  EC Retransmission Count                 :   0
  Characters transmitted, received        :   677, 817
  Characters received BAD                 :   0
  PPP/SLIP packets transmitted, received  :   10, 10
  PPP/SLIP packets received (BAD/ABORTED) :   0
  EC packets transmitted, received OK     :   10, 21
  EC packets (Received BAD/ABORTED)       :   0
  TX,RX Bit Rate                          :   52000, 28800
  Total Speedshifts                       :   0
  Total MOH Time                          :   0  secs
  Current MOH Time                        :   0  secs
  MOH Status                              :   Modem is Not on Hold
  MOH Count                               :   0
  MOH Request Count                       :   0
  Retrains due to Call Waiting            :   0
  DC Encoder,Decoder State                :   compressed/compressed
  DC TX,RX Compression Ratio              :   1.67:1/1.65:1
  DC TX,RX Dictionary Reset Count         :   0, 1
  Diagnostic Code                         :   00 00 00 00 00 00 00 00
01:46:37:Modem Link Rate event:
```

Show Output 2

```
Router# show users

    Line       User       Host(s)              Idle       Location
*  0 con 0                 idle                 00:00:00
  tty 1/05     Administra Async interface      00:00:29   PPP: 70.2.2.6

  Interface      User        Mode                    Idle     Peer Address
```

Troubleshooting Tips

If you see that V.92 call information is not being reported by AAA, ensure that the call is a V.92 call by using the **show modem** command or by looking at the modem logs by using the **show modem log** command.

Additional References

For additional information related to the V.92 Reporting Using RADIUS Attribute v.92-info feature, refer to the following references:

Related Documents

Related Topic	Document Title
AAA accounting	The chapters "AAA Overview" and "Configuring Accounting" in the "Authentication, Authorization, and Accounting" section of the *Cisco IOS Security Configuration Guide,* Release 12.3.
AAA accounting commands	The *Cisco IOS Security Command Reference*, Release 12.3.
V.92 Quick Connect feature	*V.92 Quick Connect for Cisco AS5300 and Cisco AS5800 Universal Access Servers*
V.92 Modem on Hold feature	*V.92 Modem on Hold for Cisco AS5300 and Cisco AS5800 Universal Access Servers*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

This feature uses no new or modified commands. To see the command pages for the commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

TACACS+ Attribute-Value Pairs

Terminal Access Controller Access Control System Plus (TACACS+) attribute-value (AV) pairs are used to define specific authentication, authorization, and accounting elements in a user profile that is stored on the TACACS+ daemon. This appendix lists the TACACS+ AV pairs currently supported.

How to Use This Appendix

This appendix is divided into two sections:

- TACACS+ Authentication and Authorization AV Pairs
- TACACS+ Accounting AV Pairs

The first section lists and describes the supported TACACS+ authentication and authorization AV pairs, and it specifies the Cisco IOS release in which they are implemented. The second section lists and describes the supported TACACS+ accounting AV pairs, and it specifies the Cisco IOS release in which they are implemented.

TACACS+ Authentication and Authorization AV Pairs

Table 80 lists and describes the supported TACACS+ authentication and authorization AV pairs and specifies the Cisco IOS release in which they are implemented.

Table 80 Supported TACACS+ Authentication and Authorization AV Pairs

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2
acl=x	ASCII number representing a connection access list. Used only when service=shell.	yes	yes	yes	yes	yes	yes	yes
addr=x	A network address. Used with service=slip, service=ppp, and protocol=ip. Contains the IP address that the remote host should use when connecting via SLIP or PPP/IP. For example, addr=10.2.3.4.	yes	yes	yes	yes	yes	yes	yes

Table 80 Supported TACACS+ Authentication and Authorization AV Pairs (continued)

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2
addr-pool=x	Specifies the name of a local pool from which to get the address of the remote host. Used with service=ppp and protocol=ip. Note that **addr-pool** works in conjunction with local pooling. It specifies the name of a local pool (which must be preconfigured on the network access server). Use the **ip-local pool** command to declare local pools. For example: `ip address-pool local` `ip local pool boo 10.0.0.1 10.0.0.10` `ip local pool moo 10.0.0.1 10.0.0.20` You can then use TACACS+ to return addr-pool=boo or addr-pool=moo to indicate the address pool from which you want to get this remote node's address.	yes	yes	yes	yes	yes	yes	yes
autocmd=x	Specifies an autocommand to be executed at EXEC startup (for example, autocmd=telnet example.com). Used only with service=shell.	yes	yes	yes	yes	yes	yes	yes
callback-dialstring	Sets the telephone number for a callback (for example: callback-dialstring= 408-555-1212). Value is NULL, or a dial-string. A NULL value indicates that the service might choose to get the dial string through other means. Used with service=arap, service=slip, service=ppp, service=shell. Not valid for ISDN.	no	yes	yes	yes	yes	yes	yes
callback-line	The number of a TTY line to use for callback (for example: callback-line=4). Used with service=arap, service=slip, service=ppp, service=shell. Not valid for ISDN.	no	yes	yes	yes	yes	yes	yes
callback-rotary	The number of a rotary group (between 0 and 100 inclusive) to use for callback (for example: callback-rotary=34). Used with service=arap, service=slip, service=ppp, service=shell. Not valid for ISDN.	no	yes	yes	yes	yes	yes	yes
cmd-arg=x	An argument to a shell (EXEC) command. This indicates an argument for the shell command that is to be run. Multiple cmd-arg attributes can be specified, and they are order dependent. **Note** This TACACS+ AV pair cannot be used with RADIUS attribute 26.	yes	yes	yes	yes	yes	yes	yes

Table 80 Supported TACACS+ Authentication and Authorization AV Pairs (continued)

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2
cmd=x	A shell (EXEC) command. This indicates the command name for a shell command that is to be run. This attribute must be specified if service equals "shell." A NULL value indicates that the shell itself is being referred to. **Note** This TACACS+ AV pair cannot be used with RADIUS attribute 26.	yes	yes	yes	yes	yes	yes	yes
data-service	Used with the service=outbound and protocol=ip.	no	no	no	no	no	yes	yes
dial-number	Defines the number to dial. Used with the service=outbound and protocol=ip.	no	no	no	no	no	yes	yes
dns-servers=	Identifies a DNS server (primary or secondary) that can be requested by Microsoft PPP clients from the network access server during IPCP negotiation. To be used with service=ppp and protocol=ip. The IP address identifying each DNS server is entered in dotted decimal format.	no	no	no	yes	yes	yes	yes
force-56	Determines whether the network access server uses only the 56 K portion of a channel, even when all 64 K appear to be available. To turn on this attribute, use the "true" value (force-56=true). Any other value is treated as false. Used with the service=outbound and protocol=ip.	no	no	no	no	no	yes	yes
gw-password	Specifies the password for the home gateway during the L2F tunnel authentication. Used with service=ppp and protocol=vpdn.	no	no	yes	yes	yes	yes	yes
idletime=x	Sets a value, in minutes, after which an idle session is terminated. A value of zero indicates no timeout.	no	yes	yes	yes	yes	yes	yes
inacl#<n>	ASCII access list identifier for an input access list to be installed and applied to an interface for the duration of the current connection. Used with service=ppp and protocol=ip, and service service=ppp and protocol =ipx. Per-user access lists do not currently work with ISDN interfaces.	no	no	no	yes	yes	yes	yes
inacl=x	ASCII identifier for an interface input access list. Used with service=ppp and protocol=ip. Per-user access lists do not currently work with ISDN interfaces.	yes	yes	yes	yes	yes	yes	yes

Table 80 Supported TACACS+ Authentication and Authorization AV Pairs (continued)

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2
interface-config#<n>	Specifies user-specific AAA interface configuration information with Virtual Profiles. The information that follows the equal sign (=) can be any Cisco IOS interface configuration command. Multiple instances of the attributes are allowed, but each instance must have a unique number. Used with service=ppp and protocol=lcp. **Note** This attribute replaces the "interface-config=" attribute.	no	no	no	yes	yes	yes	yes
ip-addresses	Space-separated list of possible IP addresses that can be used for the end-point of a tunnel. Used with service=ppp and protocol=vpdn.	no	no	yes	yes	yes	yes	yes
l2tp-busy-disconnect	If a vpdn-group on an LNS uses a virtual-template that is configured to be pre-cloned, this attribute will control the disposition of a new L2TP session that finds no pre-cloned interface to which to connect. If the attribute is true (the default), the session will be disconnected by the LNS. Otherwise, a new interface will be cloned from the virtual-template. Used with service=ppp and protocol=vpdn.	no	no	no	no	no	yes	yes
l2tp-cm-local-window-size	Specifies the maximum receive window size for L2TP control messages. This value is advertised to the peer during tunnel establishment. Used with service=ppp and protocol=vpdn.	no	no	no	no	no	yes	yes
l2tp-drop-out-of-order	Respects sequence numbers on data packets by dropping those that are received out of order. This does not ensure that sequence numbers will be sent on data packets, just how to handle them if they are received. Used with service=ppp and protocol=vpdn.	no	no	no	no	no	yes	yes
l2tp-hello-interval	Specifies the number of seconds for the hello keepalive interval. Hello packets are sent when no data has been sent on a tunnel for the number of seconds configured here. Used with service=ppp and protocol=vpdn.	no	no	no	no	no	yes	yes
l2tp-hidden-avp	When enabled, sensitive AVPs in L2TP control messages are scrambled or hidden. Used with service=ppp and protocol=vpdn.	no	no	no	no	no	yes	yes
l2tp-nosession-timeout	Specifies the number of seconds that a tunnel will stay active with no sessions before timing out and shutting down. Used with service=ppp and protocol=vpdn.	no	no	no	no	no	yes	yes

Table 80 Supported TACACS+ Authentication and Authorization AV Pairs (continued)

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2
l2tp-tos-reflect	Copies the IP ToS field from the IP header of each payload packet to the IP header of the tunnel packet for packets entering the tunnel at the LNS. Used with service=ppp and protocol=vpdn.	no	no	no	no	no	yes	yes
l2tp-tunnel-authen	If this attribute is set, it performs L2TP tunnel authentication. Used with service=ppp and protocol=vpdn.	no	no	no	no	no	yes	yes
l2tp-tunnel-password	Shared secret used for L2TP tunnel authentication and AVP hiding. Used with service=ppp and protocol=vpdn.	no	no	no	no	no	yes	yes
l2tp-udp-checksum	This is an authorization attribute and defines whether L2TP should perform UDP checksums for data packets. Valid values are “yes” and “no.” The default is no. Used with service=ppp and protocol=vpdn.	no	no	no	no	no	yes	yes
link-compression=	Defines whether to turn on or turn off “stac” compression over a PPP link. Used with service=ppp. Link compression is defined as a numeric value as follows: • 0: None • 1: Stac • 2: Stac-Draft-9 • 3: MS-Stac	no	no	no	yes	yes	yes	yes
load-threshold=<n>	Sets the load threshold for the caller at which additional links are either added to or deleted from the multilink bundle. If the load goes above the specified value, additional links are added. If the load goes below the specified value, links are deleted. Used with service=ppp and protocol=multilink. The range for <n> is from 1 to 255.	no	no	no	yes	yes	yes	yes
map-class	Allows the user profile to reference information configured in a map class of the same name on the network access server that dials out. Used with the service=outbound and protocol=ip.	no	no	no	no	no	yes	yes
max-links=<n>	Restricts the number of links that a user can have in a multilink bundle. Used with service=ppp and protocol=multilink. The range for <n> is from 1 to 255.	no	no	no	yes	yes	yes	yes
min-links	Sets the minimum number of links for MLP. Used with service=ppp and protocol=multilink, protocol=vpdn.	no	no	no	no	no	yes	yes

Table 80 Supported TACACS+ Authentication and Authorization AV Pairs (continued)

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2
nas-password	Specifies the password for the network access server during the L2F tunnel authentication. Used with service=ppp and protocol=vpdn.	no	no	yes	yes	yes	yes	yes
nocallback-verify	Indicates that no callback verification is required. The only valid value for this parameter is 1 (for example, nocallback-verify=1). Used with service=arap, service=slip, service=ppp, service=shell. There is no authentication on callback. Not valid for ISDN.	no	yes	yes	yes	yes	yes	yes
noescape=x	Prevents user from using an escape character. Used with service=shell. Can be either true or false (for example, noescape=true).	yes	yes	yes	yes	yes	yes	yes
nohangup=x	Used with service=shell. Specifies the nohangup option, which means that after an EXEC shell is terminated, the user is presented with another login (username) prompt. Can be either true or false (for example, nohangup=false).	yes	yes	yes	yes	yes	yes	yes
old-prompts	Allows providers to make the prompts in TACACS+ appear identical to those of earlier systems (TACACS and Extended TACACS). This allows administrators to upgrade from TACACS or Extended TACACS to TACACS+ transparently to users.	yes	yes	yes	yes	yes	yes	yes
outacl#<n>	ASCII access list identifier for an interface output access list to be installed and applied to an interface for the duration of the current condition. Used with service=ppp and protocol=ip, and service service=ppp and protocol=ipx. Per-user access lists do not currently work with ISDN interfaces.	no	no	no	yes	yes	yes	yes
outacl=x	ASCII identifier for an interface output access list. Used with service=ppp and protocol=ip, and service service=ppp and protocol=ipx. Contains an IP output access list for SLIP or PPP/IP (for example, outacl=4). The access list itself must be preconfigured on the router. Per-user access lists do not currently work with ISDN interfaces.	yes (PPP /IP only)	yes	yes	yes	yes	yes	yes
pool-def#<n>	Defines IP address pools on the network access server. Used with service=ppp and protocol=ip.	no	no	no	yes	yes	yes	yes

Table 80 Supported TACACS+ Authentication and Authorization AV Pairs (continued)

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2
pool-timeout=	Defines (in conjunction with pool-def) IP address pools on the network access server. During IPCP address negotiation, if an IP pool name is specified for a user (see the addr-pool attribute), a check is made to see if the named pool is defined on the network access server. If it is, the pool is consulted for an IP address. Used with service=ppp and protocol=ip.	no	no	yes	yes	yes	yes	yes
port-type	Indicates the type of physical port the network access server is using to authenticate the user. Physical ports are indicated by a numeric value as follows: • 0: Asynchronous • 1: Synchronous • 2: ISDN-Synchronous • 3: ISDN-Asynchronous (V.120) • 4: ISDN- Asynchronous (V.110) • 5: Virtual Used with service=any and protocol=aaa.	no	no	no	no	no	yes	yes
ppp-vj-slot-compression	Instructs the Cisco router not to use slot compression when sending VJ-compressed packets over a PPP link.	no	no	no	yes	yes	yes	yes
priv-lvl=x	Privilege level to be assigned for the EXEC. Used with service=shell. Privilege levels range from 0 to 15, with 15 being the highest.	yes	yes	yes	yes	yes	yes	yes
protocol=x	A protocol that is a subset of a service. An example would be any PPP NCP. Currently known values are **lcp**, **ip**, **ipx**, **atalk**, **vines**, **lat**, **xremote**, **tn3270**, **telnet**, **rlogin**, **pad**, **vpdn**, **osicp**, **deccp**, **ccp**, **cdp**, **bridging**, **xns**, **nbf**, **bap**, **multilink**, and **unknown**.	yes	yes	yes	yes	yes	yes	yes
proxyacl#<n>	Allows users to configure the downloadable user profiles (dynamic ACLs) by using the authentication proxy feature so that users can have the configured authorization to permit traffic going through the configured interfaces. Used with the service=shell and protocol=exec.	no	no	no	no	no	yes	yes

Table 80 Supported TACACS+ Authentication and Authorization AV Pairs (continued)

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2
route	Specifies a route to be applied to an interface. Used with service=slip, service=ppp, and protocol=ip. During network authorization, the route attribute can be used to specify a per-user static route, to be installed by TACACS+ as follows: `route="`*dst_address mask* `[`*gateway*`]"` This indicates a temporary static route that is to be applied. The *dst_address*, *mask*, and *gateway* are expected to be in the usual dotted-decimal notation, with the same meanings as in the familiar **ip route** configuration command on a network access server. If *gateway* is omitted, the peer's address is the gateway. The route is expunged when the connection terminates.	no	yes	yes	yes	yes	yes	yes
route#<n>	Like the route AV pair, this specifies a route to be applied to an interface, but these routes are numbered, allowing multiple routes to be applied. Used with service=ppp and protocol=ip, and service=ppp and protocol=ipx.	no	no	no	yes	yes	yes	yes
routing=x	Specifies whether routing information is to be propagated to and accepted from this interface. Used with service=slip, service=ppp, and protocol=ip. Equivalent in function to the /routing flag in SLIP and PPP commands. Can either be true or false (for example, routing=true).	yes	yes	yes	yes	yes	yes	yes
rte-fltr-in#<n>	Specifies an input access list definition to be installed and applied to routing updates on the current interface for the duration of the current connection. Used with service=ppp and protocol=ip, and with service=ppp and protocol=ipx.	no	no	no	yes	yes	yes	yes
rte-fltr-out#<n>	Specifies an output access list definition to be installed and applied to routing updates on the current interface for the duration of the current connection. Used with service=ppp and protocol=ip, and with service=ppp and protocol=ipx.	no	no	no	yes	yes	yes	yes
sap#<n>	Specifies static Service Advertising Protocol (SAP) entries to be installed for the duration of a connection. Used with service=ppp and protocol=ipx.	no	no	no	yes	yes	yes	yes

Table 80 Supported TACACS+ Authentication and Authorization AV Pairs (continued)

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2
sap-fltr-in#<n>	Specifies an input SAP filter access list definition to be installed and applied on the current interface for the duration of the current connection. Used with service=ppp and protocol=ipx.	no	no	no	yes	yes	yes	yes
sap-fltr-out#<n>	Specifies an output SAP filter access list definition to be installed and applied on the current interface for the duration of the current connection. Used with service=ppp and protocol=ipx.	no	no	no	yes	yes	yes	yes
send-auth	Defines the protocol to use (PAP or CHAP) for username-password authentication following CLID authentication. Used with service=any and protocol=aaa.	no	no	no	no	no	yes	yes
send-secret	Specifies the password that the NAS needs to respond to a chap/pap request from the remote end of a connection on an outgoing call. Used with service=ppp and protocol=ip.	no	no	no	no	no	yes	yes
service=x	The primary service. Specifying a service attribute indicates that this is a request for authorization or accounting of that service. Current values are **slip**, **ppp**, **arap**, **shell**, **tty-daemon**, **connection**, and **system**. This attribute must always be included.	yes	yes	yes	yes	yes	yes	yes
source-ip=x	Used as the source IP address of all VPDN packets generated as part of a VPDN tunnel. This is equivalent to the Cisco **vpdn outgoing** global configuration command.	no	no	yes	yes	yes	yes	yes
spi	Carries the authentication information needed by the home agent to authenticate a mobile node during registration. The information is in the same syntax as the **ip mobile secure host <addr>** configuration command. Basically it contains the rest of the configuration command that follows that string, verbatim. It provides the Security Parameter Index (SPI), key, authentication algorithm, authentication mode, and replay protection timestamp range. Used with the service=mobileip and protocol=ip.	no	no	no	no	no	yes	yes
timeout=x	The number of minutes before an EXEC or ARA session disconnects (for example, timeout=60). A value of zero indicates no timeout. Used with service=arap.	yes	yes	yes	yes	yes	yes	yes
tunnel-id	Specifies the username that will be used to authenticate the tunnel over which the individual user MID will be projected. This is analogous to the *remote name* in the **vpdn outgoing** command. Used with service=ppp and protocol=vpdn.	no	no	yes	yes	yes	yes	yes

Table 80 Supported TACACS+ Authentication and Authorization AV Pairs (continued)

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2
wins-servers=	Identifies a Windows NT server that can be requested by Microsoft PPP clients from the network access server during IPCP negotiation. To be used with service=ppp and protocol=ip. The IP address identifying each Windows NT server is entered in dotted decimal format.	no	no	no	yes	yes	yes	yes
zonelist=x	A numeric zonelist value. Used with service=arap. Specifies an AppleTalk zonelist for ARA (for example, zonelist=5).	yes	yes	yes	yes	yes	yes	yes

For more information about configuring TACACS+, refer to the chapter "Configuring TACACS+." For more information about configuring TACACS+ authentication and authorization, refer to the chapters "Configuring Authentication" and "Configuring Authorization."

TACACS+ Accounting AV Pairs

Table 81 lists and describes the supported TACACS+ accounting AV pairs and specifies the Cisco IOS release in which they are implemented.

Table 81 Supported TACACS+ Accounting AV Pairs

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2
Abort-Cause	If the fax session aborts, indicates the system component that signaled the abort. Examples of system components that could trigger an abort are FAP (Fax Application Process), TIFF (the TIFF reader or the TIFF writer), fax-mail client, fax-mail server, ESMTP client, or ESMTP server.	no	no	no	no	no	yes	yes
bytes_in	The number of input bytes transferred during this connection.	yes	yes	yes	yes	yes	yes	yes
bytes_out	The number of output bytes transferred during this connection.	yes	yes	yes	yes	yes	yes	yes
Call-Type	Describes the type of fax activity: fax receive or fax send.	no	no	no	no	no	yes	yes
cmd	The command the user executed.	yes	yes	yes	yes	yes	yes	yes
data-rate	This AV pair has been renamed. See nas-rx-speed.							
disc-cause	Specifies the reason a connection was taken off-line. The Disconnect-Cause attribute is sent in accounting-stop records. This attribute also causes stop records to be generated without first generating start records if disconnection occurs before authentication is performed. Refer to Table 82 for a list of Disconnect-Cause values and their meanings.	no	no	no	yes	yes	yes	yes

Table 81 Supported TACACS+ Accounting AV Pairs (continued)

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2
disc-cause-ext	Extends the disc-cause attribute to support vendor-specific reasons why a connection was taken off-line.	no	no	no	yes	yes	yes	yes
elapsed_time	The elapsed time in seconds for the action. Useful when the device does not keep real time.	yes	yes	yes	yes	yes	yes	yes
Email-Server-Address	Indicates the IP address of the e-mail server handling the on-ramp fax-mail message.	no	no	no	no	no	yes	yes
Email-Server-Ack-Flag	Indicates that the on-ramp gateway has received a positive acknowledgment from the e-mail server accepting the fax-mail message.	no	no	no	no	no	yes	yes
event	Information included in the accounting packet that describes a state change in the router. Events described are accounting starting and accounting stopping.	yes	yes	yes	yes	yes	yes	yes
Fax-Account-Id-Origin	Indicates the account ID origin as defined by system administrator for the **mmoip aaa receive-id** or the **mmoip aaa send-id** command.	no	no	no	no	no	yes	yes
Fax-Auth-Status	Indicates whether or not authentication for this fax session was successful. Possible values for this field are success, failed, bypassed, or unknown.	no	no	no	no	no	yes	yes
Fax-Connect-Speed	Indicates the modem speed at which this fax-mail was initially transmitted or received. Possible values are 1200, 4800, 9600, and 14400.	no	no	no	no	no	yes	yes
Fax-Coverpage-Flag	Indicates whether or not a cover page was generated by the off-ramp gateway for this fax session. True indicates that a cover page was generated; false means that a cover page was not generated.	no	no	no	no	no	yes	yes
Fax-Dsn-Address	Indicates the address to which DSNs will be sent.	no	no	no	no	no	yes	yes
Fax-Dsn-Flag	Indicates whether or not DSN has been enabled. True indicates that DSN has been enabled; false means that DSN has not been enabled.	no	no	no	no	no	yes	yes
Fax-Mdn-Address	Indicates the address to which MDNs will be sent.	no	no	no	no	no	yes	yes
Fax-Mdn-Flag	Indicates whether or not message delivery notification (MDN) has been enabled. True indicates that MDN had been enabled; false means that MDN had not been enabled.	no	no	no	no	no	yes	yes
Fax-Modem-Time	Indicates the amount of time in seconds the modem sent fax data (x) and the amount of time in seconds of the total fax session (y), which includes both fax-mail and PSTN time, in the form x/y. For example, 10/15 means that the transfer time took 10 seconds, and the total fax session took 15 seconds.	no	no	no	no	no	yes	yes

Table 81 Supported TACACS+ Accounting AV Pairs (continued)

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2
Fax-Msg-Id=	Indicates a unique fax message identification number assigned by Store and Forward Fax.	no	no	no	no	no	yes	yes
Fax-Pages	Indicates the number of pages transmitted or received during this fax session. This page count includes cover pages.	no	no	no	no	no	yes	yes
Fax-Process-Abort-Flag	Indicates that the fax session was aborted or successful. True means that the session was aborted; false means that the session was successful.	no	no	no	no	no	yes	yes
Fax-Recipient-Count	Indicates the number of recipients for this fax transmission. Until e-mail servers support Session mode, the number should be 1.	no	no	no	no	no	yes	yes
Gateway-Id	Indicates the name of the gateway that processed the fax session. The name appears in the following format: hostname.domain-name	no	no	no	no	no	yes	yes
mlp-links-max	Gives the count of links which are known to have been in a given multilink session at the time the accounting record is generated.	no	no	no	yes	yes	yes	yes
mlp-sess-id	Reports the identification number of the multilink bundle when the session closes. This attribute applies to sessions that are part of a multilink bundle. This attribute is sent in authentication-response packets.	no	no	no	yes	yes	yes	yes
nas-rx-speed	Specifies the average number of bits per second over the course of the connection's lifetime. This attribute is sent in accounting-stop records.	no	no	no	yes	yes	yes	yes
nas-tx-speed	Reports the transmit speed negotiated by the two modems.	no	no	no	yes	yes	yes	yes
paks_in	The number of input packets transferred during this connection.	yes	yes	yes	yes	yes	yes	yes
paks_out	The number of output packets transferred during this connection.	yes	yes	yes	yes	yes	yes	yes
port	The port the user was logged in to.	yes	yes	yes	yes	yes	yes	yes
Port-Used	Indicates the slot/port number of the Cisco AS5300 used to either transmit or receive this fax-mail.	no	no	no	no	no	yes	yes
pre-bytes-in	Records the number of input bytes before authentication. This attribute is sent in accounting-stop records.	no	no	no	yes	yes	yes	yes
pre-bytes-out	Records the number of output bytes before authentication. This attribute is sent in accounting-stop records.	no	no	no	yes	yes	yes	yes
pre-paks-in	Records the number of input packets before authentication. This attribute is sent in accounting-stop records.	no	no	no	yes	yes	yes	yes

Table 81 Supported TACACS+ Accounting AV Pairs (continued)

Attribute	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2
pre-paks-out	Records the number of output packets before authentication. The Pre-Output-Packets attribute is sent in accounting-stop records.	no	no	no	yes	yes	yes	yes
pre-session-time	Specifies the length of time, in seconds, from when a call first connects to when it completes authentication.	no	no	no	yes	yes	yes	yes
priv_level	The privilege level associated with the action.	yes	yes	yes	yes	yes	yes	yes
protocol	The protocol associated with the action.	yes	yes	yes	yes	yes	yes	yes
reason	Information included in the accounting packet that describes the event that caused a system change. Events described are system reload, system shutdown, or when accounting is reconfigured (turned on or off).	yes	yes	yes	yes	yes	yes	yes
service	The service the user used.	yes	yes	yes	yes	yes	yes	yes
start_time	The time the action started (in seconds since the epoch, 12:00 a.m. Jan 1 1970). The clock must be configured to receive this information.	yes	yes	yes	yes	yes	yes	yes
stop_time	The time the action stopped (in seconds since the epoch.) The clock must be configured to receive this information.	yes	yes	yes	yes	yes	yes	yes
task_id	Start and stop records for the same event must have matching (unique) task_id numbers.	yes	yes	yes	yes	yes	yes	yes
timezone	The time zone abbreviation for all timestamps included in this packet.	yes	yes	yes	yes	yes	yes	yes
xmit-rate	This AV pair has been renamed. See nas-tx-speed.							

Table 82 lists the cause codes and descriptions for the Disconnect Cause Extended (disc-cause-ext) attribute.

Table 82 Disconnect Cause Extensions

Cause Codes	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1000 – No Reason	No reason for the disconnect.	no	no	no	no	yes	yes	yes	yes
1001 – No Disconnect	The event was not a disconnect.	no	no	no	no	yes	yes	yes	yes
1002 – Unknown	The reason for the disconnect is unknown. This code can appear when the remote connection goes down.	no	no	no	no	yes	yes	yes	yes
1003 – Call Disconnect	The call has disconnected.	no	no	no	no	yes	yes	yes	yes
1004 – CLID Auth Fail	Calling line ID (CLID) authentication has failed.	no	no	no	no	yes	yes	yes	yes

Cause Codes	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1009 – No Modem Available	The modem is not available.	no	no	no	no	yes	yes	yes	yes
1010 – No Carrier	The modem never detected data carrier detect (DCD). This code can appear if a disconnect occurs during the initial modem connection.	no	no	no	no	yes	yes	yes	yes
1011 – Lost Carrier	The modem detected DCD but became inactive. This code can appear if a disconnect occurs during the initial modem connection.	no	no	no	no	yes	yes	yes	yes
1012 – No Modem Results	The result codes could not be parsed. This code can appear if a disconnect occurs during the initial modem connection.	no	no	no	no	yes	yes	yes	yes
1020 – TS User Exit	The user exited normally from the terminal server. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes
1021 – Idle Timeout	The user exited from the terminal server because the idle timer expired. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes
1022 – TS Exit Telnet	The user exited normally from a Telnet session. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes
1023 – TS No IP Addr	The user could not switch to Serial Line Internet Protocol (SLIP) or PPP because the remote host had no IP address or because the dynamic pool could not assign one. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes
1024 – TS TCP Raw Exit	The user exited normally from a raw TCP session. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes
1025 – TS Bad Password	The login process ended because the user failed to enter a correct password after three attempts. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes
1026 – TS No TCP Raw	The raw TCP option is not enabled. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes
1027 – TS CNTL-C	The login process ended because the user typed Ctrl-C. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes
1028 – TS Session End	The terminal server session has ended. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes

Cause Codes	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1029 – TS Close Vconn	The user closed the virtual connection. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes
1030 – TS End Vconn	The virtual connection has ended. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes
1031 – TS Rlogin Exit	The user exited normally from an Rlogin session. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes
1032 – TS Rlogin Opt Invalid	The user selected an invalid Rlogin option. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes
1033 – TS Insuff Resources	The access server has insufficient resources for the terminal server session. This code is related to immediate Telnet and raw TCP disconnects during a terminal server session.	no	no	no	no	yes	yes	yes	yes
1040 – PPP LCP Timeout	PPP link control protocol (LCP) negotiation timed out while waiting for a response from a peer. This code concerns PPP connections.	no	no	no	no	yes	yes	yes	yes
1041 – PPP LCP Fail	There was a failure to converge on PPP LCP negotiations. This code concerns PPP connections.	no	no	no	no	yes	yes	yes	yes
1042 – PPP Pap Fail	PPP Password Authentication Protocol (PAP) authentication failed. This code concerns PPP connections.	no	no	no	no	yes	yes	yes	yes
1043 – PPP CHAP Fail	PPP Challenge Handshake Authentication Protocol (CHAP) authentication failed. This code concerns PPP connections.	no	no	no	no	yes	yes	yes	yes
1044 – PPP Remote Fail	Authentication failed from the remote server. This code concerns PPP sessions.	no	no	no	no	yes	yes	yes	yes
1045 – PPP Receive Term	The peer sent a PPP termination request. This code concerns PPP connections.	no	no	no	no	yes	yes	yes	yes
PPP LCP Close (1046)	LCP got a close request from the upper layer while LCP was in an open state. This code concerns PPP connections.	no	no	no	no	yes	yes	yes	yes
1047 – PPP No NCP	LCP closed because no NCPs were open. This code concerns PPP connections.	no	no	no	no	yes	yes	yes	yes
1048 – PPP MP Error	LCP closed because it could not determine to which Multilink PPP bundle that it should add the user. This code concerns PPP connections.	no	no	no	no	yes	yes	yes	yes
1049 – PPP Max Channels	LCP closed because the access server could not add any more channels to an MP session. This code concerns PPP connections.	no	no	no	no	yes	yes	yes	yes

Cause Codes	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1050 – TS Tables Full	The raw TCP or Telnet internal session tables are full. This code relates to immediate Telnet and raw TCP disconnects and contains more specific information than the Telnet and TCP codes listed earlier in this table.	no	no	no	no	yes	yes	yes	yes
1051 – TS Resource Full	Internal resources are full. This code relates to immediate Telnet and raw TCP disconnects and contains more specific information than the Telnet and TCP codes listed earlier in this table.	no	no	no	no	yes	yes	yes	yes
1052 – TS Invalid IP Addr	The IP address for the Telnet host is invalid. This code relates to immediate Telnet and raw TCP disconnects and contains more specific information than the Telnet and TCP codes listed earlier in this table.	no	no	no	no	yes	yes	yes	yes
1053 – TS Bad Hostname	The access server could not resolve the host name. This code relates to immediate Telnet and raw TCP disconnects and contains more specific information than the Telnet and TCP codes listed earlier in this table.	no	no	no	no	yes	yes	yes	yes
1054 – TS Bad Port	The access server detected a bad or missing port number. This code relates to immediate Telnet and raw TCP disconnects and contains more specific information than the Telnet and TCP codes listed earlier in this table.	no	no	no	no	yes	yes	yes	yes
1060 – TCP Reset	The host reset the TCP connection. The TCP stack can return this disconnect code during an immediate Telnet or raw TCP session.	no	no	no	no	yes	yes	yes	yes
1061 – TCP Connection Refused	The host refused the TCP connection. The TCP stack can return this disconnect code during an immediate Telnet or raw TCP session.	no	no	no	no	yes	yes	yes	yes
1062 – TCP Timeout	The TCP connection timed out. The TCP stack can return this disconnect code during an immediate Telnet or raw TCP session.	no	no	no	no	yes	yes	yes	yes
1063 – TCP Foreign Host Close	A foreign host closed the TCP connection. The TCP stack can return this disconnect code during an immediate Telnet or raw TCP session.	no	no	no	no	yes	yes	yes	yes
1064 – TCP Net Unreachable	The TCP network was unreachable. The TCP stack can return this disconnect code during an immediate Telnet or raw TCP session.	no	no	no	no	yes	yes	yes	yes
1065 – TCP Host Unreachable	The TCP host was unreachable. The TCP stack can return this disconnect code during an immediate Telnet or raw TCP session.	no	no	no	no	yes	yes	yes	yes
1066 – TCP Net Admin Unreachable	The TCP network was administratively unreachable. The TCP stack can return this disconnect code during an immediate Telnet or raw TCP session.	no	no	no	no	yes	yes	yes	yes

Cause Codes	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1067 – TCP Host Admin Unreachable	The TCP host was administratively unreachable. The TCP stack can return this disconnect code during an immediate Telnet or raw TCP session.	no	no	no	no	yes	yes	yes	yes
1068 – TCP Port Unreachable	The TCP port was unreachable. The TCP stack can return this disconnect code during an immediate Telnet or raw TCP session.	no	no	no	no	yes	yes	yes	yes
1100 – Session Timeout	The session timed out because there was no activity on a PPP link. This code applies to all session types.	no	no	no	no	yes	yes	yes	yes
1101 – Security Fail	The session failed for security reasons. This code applies to all session types.	no	no	no	no	yes	yes	yes	yes
1102 – Callback	The session ended for callback. This code applies to all session types.	no	no	no	no	yes	yes	yes	yes
1120 – Unsupported	One end refused the call because the protocol was disabled or unsupported. This code applies to all session types.	no	no	no	no	yes	yes	yes	yes
1150 – Radius Disc	The RADIUS server requested the disconnect.	no	no	no	no	yes	yes	yes	yes
1151 – Local Admin Disc	The local administrator has disconnected.	no	no	no	no	yes	yes	yes	yes
1152 – SNMP Disc	Simple Network Management Protocol (SNMP) has disconnected.	no	no	no	no	yes	yes	yes	yes
1160 – V110 Retries	The allowed retries for V110 synchronization have been exceeded.	no	no	no	no	yes	yes	yes	yes
1170 – PPP Auth Timeout	Authentication timeout. This code applies to PPP sessions.	no	no	no	no	yes	yes	yes	yes
1180 – Local Hangup	The call disconnected as the result of a local hangup.	no	no	no	no	yes	yes	yes	yes
1185 – Remote Hangup	The call disconnected because the remote end hung up.	no	no	no	no	yes	yes	yes	yes
1190 – T1 Quiesced	The call disconnected because the T1 line that carried it was quiesced.	no	no	no	no	yes	yes	yes	yes
1195 – Call Duration	The call disconnected because the call duration exceeded the maximum amount of time allowed by the Max Call Mins or Max DS0 Mins parameter on the access server.	no	no	no	no	yes	yes	yes	yes
1600 – VPDN User Disconnect	The user disconnected. This value applies to virtual private dial-up network (VPDN) sessions.	no	no	no	no	no	no	yes	yes
1601 – VPDN Carrier Loss	Carrier loss has occurred. This code applies to VPDN sessions.	no	no	no	no	no	no	yes	yes
1602 – VPDN No Resources	There are no resources. This code applies to VPDN sessions.	no	no	no	no	no	no	yes	yes
1603 – VPDN Bad Control Packet	The control packet is invalid. This code applies to VPDN sessions.	no	no	no	no	no	no	yes	yes

Cause Codes	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1604 – VPDN Admin Disconnect	The administrator disconnected. This code applies to VPDN sessions.	no	no	no	no	no	no	yes	yes
1605 – VPDN Tunnel Down/Setup Fail	The tunnel is down or the setup failed. This code applies to VPDN sessions.	no	no	no	no	no	no	yes	yes
1606 – VPDN Local PPP Disconnect	There was a local PPP disconnect. This code applies to VPDN sessions.	no	no	no	no	no	no	yes	yes
1607 – VPDN Softshut/Session Limit	New sessions cannot be established on the VPN tunnel. This code applies to VPDN sessions.	no	no	no	no	no	no	yes	yes
1608 – VPDN Call Redirected	The call was redirected. This code applies to VPDN sessions.	no	no	no	no	no	no	yes	yes
1801 – Q850 Unassigned Number	The number has not been assigned. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1802 – Q850 No Route	The equipment that is sending this code has received a request to route the call through a particular transit network that it does not recognize. The equipment that is sending this code does not recognize the transit network because either the transit network does not exist or because that particular transit network, while it does exist, does not serve the equipment that is sending this code. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1803 – Q850 No Route To Destination	The called party cannot be reached because the network through which the call has been routed does not serve the destination that is desired. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1806 – Q850 Channel Unacceptable	The channel that has been most recently identified is not acceptable to the sending entity for use in this call. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1816 – Q850 Normal Clearing	The call is being cleared because one of the users who is involved in the call has requested that the call be cleared. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1817 – Q850 User Busy	The called party is unable to accept another call because the user-busy condition has been encountered. This code may be generated by the called user or by the network. In the case of the user, the user equipment is compatible with the call. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes

Cause Codes	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1818 – Q850 No User Responding	Used when a called party does not respond to a call-establishment message with either an alerting or connect indication within the prescribed period of time that was allocated. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1819 – Q850 No User Answer	The called party has been alerted but does not respond with a connect indication within a prescribed period of time. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1821 – Q850 Call Rejected	The equipment that is sending this code does not wish to accept this call although it could have accepted the call because the equipment that is sending this code is neither busy nor incompatible. This code may also be generated by the network, indicating that the call was cleared due to a supplementary service constraint. The diagnostic field may contain additional information about the supplementary service and reason for rejection. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1822 – Q850 Number Changed	The number that is indicated for the called party is no longer assigned. The new called party number may optionally be included in the diagnostic field. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1827 – Q850 Destination Out of Order	The destination that was indicated by the user cannot be reached because the interface to the destination is not functioning correctly. The term “not functioning correctly” indicates that a signaling message was unable to be delivered to the remote party. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1828 – Q850 Invalid Number Format	The called party cannot be reached because the called party number is not in a valid format or is not complete. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1829 – Q850 Facility Rejected	This code is returned when a supplementary service that was requested by the user cannot be provided by the network. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1830 – Q850 Responding to Status Enquiry	This code is included in the STATUS message when the reason for generating the STATUS message was the prior receipt of a STATUS ENQUIRY message. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1831 – Q850 Unspecified Cause	No other code applies. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes

Cause Codes	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1834 – Q850 No Circuit Available	No circuit or channel is available to handle the call. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1838 – Q850 Network Out of Order	The network is not functioning correctly and the condition is likely to last a relatively long period of time. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1841 – Q850 Temporary Failure	The network is not functioning correctly and the condition is not likely to last a long period of time. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1842 – Q850 Network Congestion	The network is congested. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1843 – Q850 Access Info Discarded	This code indicates that the network could not deliver access information to the remote user as requested. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1844 – Q850 Requested Channel Not Available	This code is returned when the circuit or channel that is indicated by the requesting entity cannot be provided by the other side of the interface. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1845 – Q850 Call Pre-empted	The call was preempted. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1847 – Q850 Resource Unavailable	This code is used to report a resource-unavailable event only when no other code in the resource-unavailable class applies. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1850 – Q850 Facility Not Subscribed	Not a subscribed facility. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
1852 – Q850 Outgoing Call Barred	Although the calling party is a member of the closed user group for the outgoing closed user group call, outgoing calls are not allowed for this member. This code applies to ISDN or modem calls that came in over ISDN.	no	no	no	no	no	no	no	yes
Q850 Incoming Call Barred (1854)	Although the called party is a member of the closed user group for the incoming closed user group call, incoming calls are not allowed to this member. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1858 – Q850 Bearer Capability Not Available	The user has requested a bearer capability that is implemented by the equipment that generated this code but that is not available at this time. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes

Cause Codes	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1863 – Q850 Service Not Available	The code is used to report a service- or option-not-available event only when no other code in the service- or option-not-available class applies. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1865 – Q850 Bearer Capability Not Implemented	The equipment that is sending this code does not support the bearer capability that was requested. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1866 – Q850 Channel Not Implemented	The equipment that is sending this code does not support the channel type that was requested. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1869 – Q850 Facility Not Implemented	The supplementary service requested by the user cannot be provided by the network. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1881 – Q850 Invalid Call Reference	The equipment that is sending this code has received a message having a call reference that is not currently in use on the user-network interface. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1882 – Q850 Channel Does Not Exist	The channel most recently identified is not acceptable to the sending entity for use in this call. This code applies to ISDN or modem calls that have come in over ISDN. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1888 – Q850 Incompatible Destination	The equipment that is sending this code has received a request to establish a call that has low-layer compatibility or other compatibility attributes that cannot be accommodated. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1896 – Q850 Mandatory Info Element Is Missing	The equipment that is sending this code has received a message that is missing an information element that must be present in the message before that message can be processed. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1897 – Q850 Non Existent Message Type	The equipment that is sending this code has received a message with a message type that it does not recognize either because this is a message that is not defined or that is defined but not implemented by the equipment that is sending this code. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes

Cause Codes	Description	11.0	11.1	11.2	11.3	12.0	12.1	12.2	12.3
1898 – Q850 Invalid Message	This code is used to report an invalid message when no other code in the invalid message class applies. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1899 – Q850 Bad Info Element	The information element not recognized. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1900 – Q850 Invalid Element Contents	The equipment that is sending this code has received an information element that it has implemented; however, one or more fields in the information element are coded in such a way that has not been implemented by the equipment that is sending this code. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1901 – Q850 Wrong Message for State	The message that was received is incompatible with the call state. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1902 – Q850 Recovery on Timer Expiration	A procedure has been initiated by the expiration of a timer in association with error-handling procedures. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1903 – Q850 Info Element Error	The equipment that is sending this code has received a message that includes information elements or parameters that are not recognized because the information element identifiers or paramenter names are not defined or are defined but not implemented by the equipment that is sending this code. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1911 – Q850 Protocol Error	This code is used to report a protocol error event only when no other code in the protocol error class applies. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes
1927 – Q850 Unspecified Internetworking Event	There has been an error when interworking with a network that does not provide codes for actions that it takes. This code applies to ISDN or modem calls that have come in over ISDN.	no	no	no	no	no	no	no	yes

For more information about configuring TACACS+ accounting, refer to the chapter "Configuring Accounting."