[redacted]

Best Practices in HTTP Security

Authors: Bruce Lowenthal

Project Team: [Bruce Lowenthal, Dan Damon]

Creation Date: January 19, 2001

Last Updated: June 5, 2001

Control Number: 2

Version: 1.0

Copy Number _____

Contents

Introduction

This document details a number of security best practices for the Oracle9*i*AS HTTP server. It will not address security issues pertaining to access of the Oracle database. It offers advice to two groups:

- Software architects, consultants, and developers who build applications on the Oracle HTTP Server
- IT managers and personnel who deploy and manage applications on the Oracle HTTP Server

Unlike most other system and application areas, security is a very open-ended concern, simply because there is no specification which states how one is "allowed" to break security. Oracle makes no claim, therefore, that this document is a comprehensive review of all Oracle9*i*AS HTTP security issues.

Security problems pertaining to intranets are widely understood, because intranets have been deployed for many years. IT managers and personnel have determined important intranet security risks and the appropriate levels of response to combat these risks. This is not the case for Internet issues, where risks are not generally understood, and appropriate responses to risks have not been generally agreed upon. We will focus, therefore, on issues which matter primarily in Internet environments. We will give only a glance here and there to issues which matter primarily in intranet-restricted applications and services.

We used several criteria when selecting topics for this document. To be included, the information should:

- Be unfamiliar to people moving from the intranet world to the Internet
- Reduce exposure for major security problems
- Avoid massive changes to deployment or application design

We expect Security Best Practices will be a living document, enhanced and modified in response to the needs of the Oracle community. Topics covered in this version include:

- Firewall Architecture
- Development and Deployment
- Global Server ID Certificates and 128 bit Encryption
- Performance Issues
- Client Certificates

Many of the topics first appeared as questions and issues on Oracle security-related discussion forums, which we feel gives them especially high priority.

Firewall Architecture

This section discusses firewall architecture from the perspective of organizations wishing to provide Internet-accessible services. We will ignore services available only on intranets, browsers attached to the Internet that access services on the Internet, and intranet-attached processes which need access to Internet services.

Background

There is no single best architecture for accommodating Internet requests requiring access to corporate intranets. Instead, trade-offs must be made between two competing goals:

- Security of the intranet against Internet attack
- Ease of access to services by both Internet and intranet clients

Neither goal can be totally met, because complete security means no access to services, while complete ease of access means that anyone is free to peruse, corrupt, or modify corporate sites. We can only try to reach a balance between these goals, whose relative priorities are often unclear.

Oracle recommends two approaches, which we believe should satisfy the requirements of most of our customers:

- DMZ with bastion hosts
- Switched connection DMZ hosts

For the purposes of this discussion of Oracle's recommendations, network architecture can be divided into three regions (see figures F1 and F2 below). On one side is the wide world of the Internet; on the other side is the corporate intranet. In the middle is the De-Militarized Zone or DMZ (from the military term for an area between two opponents where fighting is prevented), separated from both of them by devices called firewalls. These firewalls block or allow data transfers based on IP address, port, protocol type, or some combination of these. They can also employ stateful inspection technology (detailed later in this paper) to detect illegal protocol transitions.

Firewalls are sometimes defined as including the DMZ, the exterior firewall and the interior firewall parts of Figure F1. In this less common definition, what we call the interior firewall is typically called a router.

Bastion hosts (see F1 below) are well-fortified servers running initial point of contact protocols, such as HTTP or SMTP. They should be set up with the expectation that outsiders will attempt to break into them. Special care should be taken to ensure that break-ins are difficult. If a break-in does occur, there should be good fault containment.

The switched connection DMZ architecture (see F2 below) takes advantage of newer firewall technology, which allows inexpensive switched connection attachments of servers. With switched connections one server cannot see the traffic generated by another server, except for broadcasts. This provides a major fault containment benefit compared to bussed connections, where all devices attached to the bus can view traffic to and from other devices on the bus.

While we recommend these approaches, we do not mean to exclude all alternatives. Other reasonable network structures might give higher priority to security or ease of access. Consideration of the issues, risks, and costs of these alternative approaches, however, should precede any deployment decision.

It should be noted that "outgoing" requests should be handled differently and have different policies than incoming requests. Thus, a reasonable policy might be to allow "outgoing" FTP requests but no incoming FTP requests.

Best Practices

Best Practice SEC-1: Place servers providing Internet services behind an exterior firewall of the stateful inspection type.

It is very important that the servers used to provide Internet services be placed in the appropriate part of the service provider's network architecture. In almost no cases should servers be placed directly on the Internet, where they are vulnerable to too many forms of attack. They should instead be behind an exterior firewall. The best firewall type is the stateful inspection type, such as those available from Checkpoint (www.checkpoint.com) and other vendors. Stateful inspection means that the firewall keeps track of various sessions by protocol and ensures that illegal protocol transitions are disallowed through the firewall. This blocks the types of intrusion which exploit illegal protocol transitions.

Best Practice SEC-2: Set exterior firewall rules to allow Internet-initiated traffic only through specific IP and PORT addresses where smtp/pop3/imap4 or http services are running.

Generally, it is best to provide only smtp / pop3 / imap4 (e-mail) and http (Browser) services to the Internet, because other protocols are too vulnerable to attack. Where it is feasible, it is best to provide the http and smtp / pop3 / imap services on the DMZ, while running applications or accessing databases on the intranet part of the network.

PORT and IP combinations which are not assigned to running programs should not be permitted. Once messages are received by the http or smtp / pop3 / imap server, they can be forwarded from DMZ processes to intranet processes.

Handling of other protocols by processors attached to the DMZ, especially raw TCP / IP or UDP, is not recommended. FTP use results in major security vulnerabilities, because it essentially sends usercodes and passwords in the clear; while IIOP opens too many ports without waiting system processes attached. If handling these protocols on the DMZ is a requirement, then the processors doing the handling should have no incoming access to the intranet.

Best Practice SEC-3: Set interior firewall rules to allow messages through to the intranet only if they originate from servers residing on the DMZ.

Intranet-initiated requests to the DMZ are not a problem, but no direct access to the intranet from the Internet should be allowed. All incoming messages *must* first be processed on the DMZ.

Best Practice SEC-4: Send outgoing messages through proxies on the DMZ.

Messages originating from processes in the intranet can be passed directly to the Internet. For more security, they can be forwarded to the Internet by *proxies* on the DMZ. Many proxy types exist for the different protocols typically operating over TCP / IP. If even more security is desired, we recommend the switched connection DMZ host architecture.

DMZ with Bastion Hosts

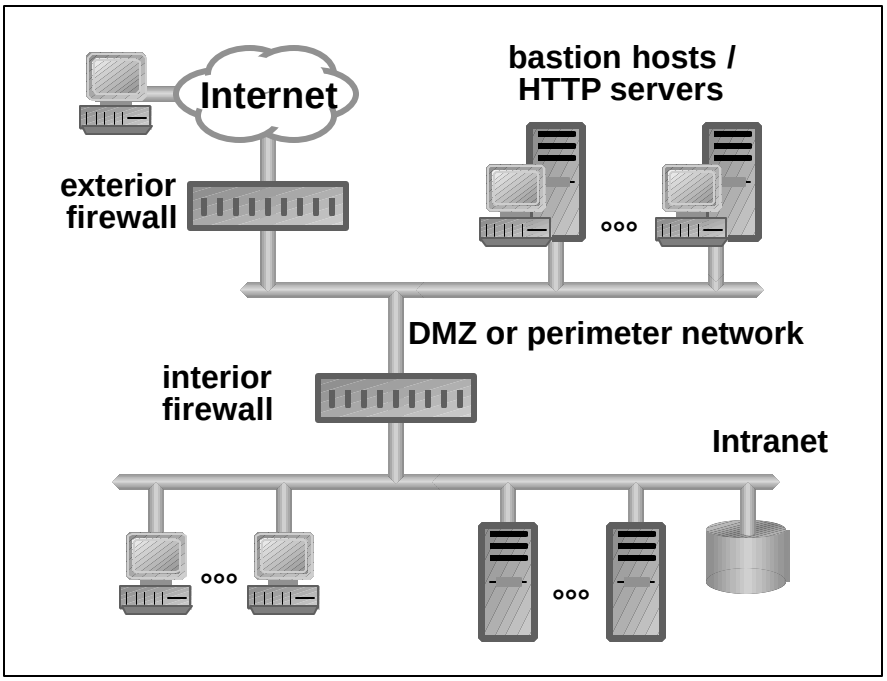

Figure F1 - Firewalls with Bastion Host

Best Practice SEC-5: Do not store the information of record on bastion hosts.

Information and processing should be segmented such that bastion hosts (fortified servers on the DMZ) provide initial protocol server processing and generally do not contain information of a sensitive nature. They should certainly not contain the information of record. That is to say, updates or corruption to information on the bastion host should not result in updates to the database of record. The database of record and all sensitive processing should reside on the intranet.

Best Practice SEC-6: Disallow all traffic types unless specifically allowed.

No one can predict what form the next attack on your network might take, and disallowing only the forms taken by past attacks will always leave you one step behind the attackers. We recommend instead that you disallow all types of traffic not required by your organization.

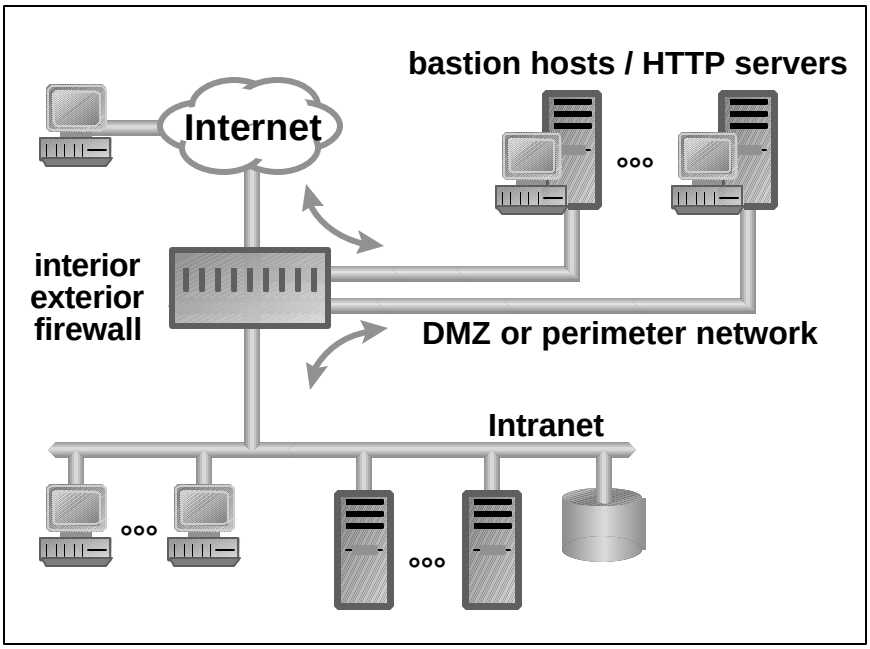

Figure F2 - Switched Connection DMZ hosts

With the bussed connection firewalls discussed in the last section, rules allowing or disallowing traffic between different server / hosts can become quite complex. While in the past there was often concern with building multiple levels of DMZs for better fault containment, most such questions are eliminated with the use of switched connection firewalls.

Note that some protocols are not encrypted. One example is AJP, the protocol between the Oracle9iAS HTTP server and JServ. When the HTTP server and the JServ server are on different computers, the use of switched rather than bussed connections provides significant additional fault containment.

Some of the Best Practices described in the last section, applicable to DMZs with bastion hosts, apply as well to switched connection DMZ architecture. Routing all incoming traffic to a DMZ server before forwarding it to the intranet is still a good idea. Also still a Best Practice is banning messages from the Internet to the DMZ, if the messages have IP addresses used in the DMZ. This preserves the DMZ concept, even though the DMZ is no longer a formal LAN or network.

Switched connection DMZ hosts architecture allows secure segregation of processing tasks. In Figure F3 for example, inexpensive servers assisted by cryptographic hardware are used to convert http to https, while separate hosts are used to segregate http servers from servlet servers on the DMZ.

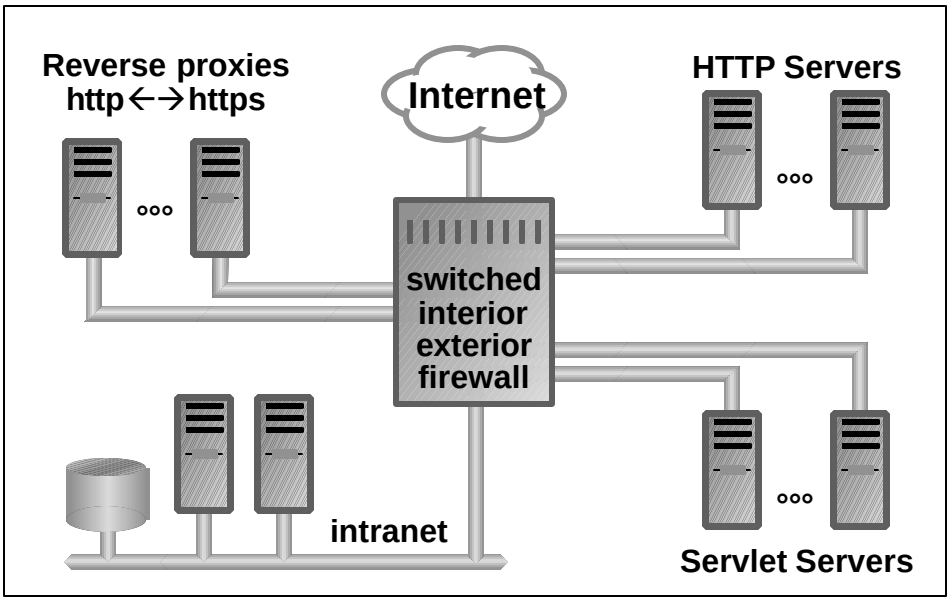

Figure F3 - Complex Switched Configurations

The switched interior exterior firewall in this example, which may be built from a number of distinct pieces of hardware and software acquired from different vendors, can provide quite complex routing rules. Incoming https traffic might get routed first to reverse proxies, which would convert the https protocol to http. Such traffic could then be routed to http servers, where requests for static content might be satisfied, while dynamic content requests might be routed to servlet servers. The servlet servers might then route requests to the intranet for further processing and database access. Another category of processors might provide WebCache capabilities.

Note that this architecture provides excellent fault containment. If one of the http servers is compromised, it cannot see the traffic from other http servers or servlet servers. Further, rules provided to the firewall could prevent compromised servers accessing other servers on the DMZ or intranet, thereby preventing more corruption and theft. Oracle plans to explore such architectural alternatives for many of Oracle's products soon in a Firewall and Load Balancing white paper.

References

- For information about stateful firewalls and Checkpoint products:
  http://www.checkpoint.com/

Process Development and Deployment

Background

Attacks on corporate networks often attempt to trick components with high levels of privilege into revealing information or modifying internal infrastructure, thereby allowing further incursion into protected resources. This section presents Best Practices pertaining to the setting of program privileges, including usercode / password management

Best Practices

Best Practice SEC-7: When assigning privileges to modules, use the lowest levels adequate to perform the module's functions.

This does not prevent trickery, but it limits the damage if a module is taken over or tricked. Faults are better contained.

Best Practice SEC-8: Ensure that programs are reviewed against buffer overflow for received data.

Buffers can overflow into data structures, resulting in a number of exploitation types—especially denial of service. Buffer overflows are considered by many to be the leading area of security vulnerability and as such deserve special consideration.

Best Practice SEC-9: Ensure that programs are reviewed against cross site scripting attacks.

These attacks typically trick HTML and XML processing via input from browsers (or processes which act like browsers) to invoke scripting engines inappropriately. In one of the attack's basic forms, the attacker enters various escape characters such as '>' or '<' when presented a form for regular input via a browser. With careful crafting the attacker can cause a script engine to process the attacker's script using the security level of the script processing engine (which may be quite high).

Consider a simple example. A form is presented to a browser requesting a description of a desired good or service. The attacker enters a bogus description along with '>' and '<' escape characters. The escape characters would later cause the output processor to process Javascript (entered as part of the bogus description) when the description was replayed. The script could read protected information from the server and then send it to the attacker via smtp (e-mail) message.

No simple and effective solutions to this problem exist. Each application writer needs to write code to scan input to ensure that such trickery is not being attempted.

Best Practice SEC-10: All U.S. built products that contain cryptographic software should be reviewed months in advance of production or Beta shipment outside the U.S. or Canada by the building company's export control unit.

A one-time review by the U.S. Department of Commerce is required for all U.S. built products using encryption technology, if the products are to be shipped outside the U.S. and Canada. Legal penalties exist for exporting such products without proper license. This review, which can take many weeks, applies to all new products and may apply to patched or new versions of existing products as well. In order to meet on-time delivery of products to the international market, it is essential that development consult with the corporate export control department months in advance of international shipment for both product Beta and production use. Corporate export control will advise development on the proper processes to ensure that appropriate licenses are obtained in time.

Where cryptographic functions are needed, use already reviewed facilities (that is, common code). This will keep the one-time review as short as possible. When shipping new versions of existing products, avoid changes to portions involving encryption if possible. This may eliminate delays for governmental review entirely.

Best Practice SEC-11: When deploying software, change all default passwords and close accounts used for samples and examples.

The risks here should be clear.

Best Practice SEC-12: Apply all relevant security patches.

Check Metalink and TechNet for current security alerts. Many of these patches address publicly announced security holes.

Best Practice SEC-13: Remove unused services from all hosts.

Examples of unused services are FTP, SNMP, NFS, BOOTP, and NEWS. It is almost always worthwhile finding ways to eliminate FTP, because it is especially noxious. HTTP or WebDAV may be a good alternatives.

Best Practice SEC-14: Limit the number of people with root privileges.

Best Practice SEC-15: Disable the “r” commands if you do not need them.

Examples of “r” commands are rlogon and rsh.

References:

- ISO Common Criteria (ISO 15408) for security evaluations
- FIPS 140-1 (Security Requirements for Cryptographic Modules)
- CERT (www.cert.org) This nonprofit organization helps other organizations defend against network attack. It is a good resource for reviewing deployment criteria, and the CERT site monitors security alerts.
- http://www.apache.org/info/css-security/ An excellent reference on the cross site scripting attack problem

Global Server ID Certificates and 128 Bit Encryption

Background

In this section, we explain why the Verisign 40-bit certificates are appropriate for use by nearly all organizations which desire bulk encryption protected by 112, 128 and / or 168 key sizes.

As a direct result of export rules in force until early 2000, web browsers and server software containing encryption technology were often built in DOMESTIC and EXPORT versions—both official designations of the U.S. Department of Commerce. The DOMESTIC versions used strong encryption and the EXPORT versions used weak encryption. Many users outside the United States and Canada (and some domestic users as well) therefore currently have weak encryption versions of popular browsers like Netscape Navigator and Microsoft Internet Explorer. In this context, weak encryption means key sizes of 64 bits or less (including 40-bit keys) for RC4 and DES bulk encryption. Strong encryption means key sizes greater than 64 bits, including the common 112, 128, and 168 bits key sizes.

Many organizations (especially in the financial, securities, medical, and insurance markets) believe that weak encryption SSL communication is unacceptable for their applications. They have successfully lobbied for export laws and international agreements to allow single use licenses for strong encryption server products in their markets. Unfortunately, use of strong encryption server products with weak encryption browsers results in weak cryptography SSL sessions. So even with their licensed strong encryption servers, these organizations would often be limited to unacceptably weak encryption SSL sessions when communicating with their customers.

Verisign, the largest supplier of server certificates, responded to this problem by developing a Global Server ID (GSID) Certificate and getting approval for it from the U.S. Department of Commerce. The GSID certificate contained a signed digital right, which will be called the *step-up* digital right in this discussion. In conjunction with development of the step-up digital right, browser and server logic supporting SSL was revised such that when a GSID certificate was used in a server, weak encryption (that is, EXPORT version) client browsers would automatically step up their encryption strength from weak encryption to strong encryption.

Global Server IDs are now of limited utility, however, because the strong encryption export ban was lifted early in 2000. Now anyone outside the handful of countries designated as “terrorist” by the U.S. State Department can legally download a strong encryption browser.

Note: There are different types of X.509 V3 certificates. In describing their X.509 v3 certificate that includes the *step-up* digital right, Verisign uses the terms 128-bit, Global Server ID, Secure Site Pro, and 128 bit global server IDs. Other companies (for example, Baltimore) offer X.509 v3 certificates with the *step-up* digital right and use other names for them. But because Verisign has about 90% of the market, its names are often applied to X.509 V3 certificates from other vendors.

This is unfortunate, because Verisign’s naming system is particularly misleading. Certificates without the step-up digital right are now called 40 bit certificates, while certificates with the step-up digital right are called 128 bit certificates. While it is true that the 128 bit certificate will allow 128 bit encryption sessions, it is also true that the 40 bit certificates will allow 128 bit encrypted sessions--if both the browser and the server support 128 bit sessions. Further, X.509 certificates are not actually 128 bit or 40 bit. They do contain a key; but it would likely be 512, 1024, or 2048 bits, and it is not for bulk encryption (which is what the 128 bit and 40 bit key sizes reference).

Best Practices

Best Practice SEC-16: Configure your web server to fail attempts to use weak encryption. Display an error page explaining the need to upgrade client browsers to 128 bit (strong encryption).

If you are considering the Verisign 128 bit certificate to ensure 128 bit encryption, use the 40 bit certificate (or an equivalent certificate from another vendor) instead, and eliminate weak encryption cipher suites from those allowed by the web server. Attempts to use weak encryption will then fail, with the consequential display of an error page explaining the need to upgrade the browser to 128 bit (strong encryption).

Some service providers may see customer inconvenience issues in requiring a move up to 128 bit browser versions. But the move is really a win-win situation for the customer, because:

- They will always get 128 bit encryption at any site that supports it
- Newer browser are more efficient because they use the latest versions of http and other protocols

References

- For general information about Verisign, see: http://www.verisign.com/
- For information on Verisign Secure Site Services, see http://www.verisign.com/products/site/secure/index.html
- For a discussion of differences between 128-bit and 40-bit Verisign certificates, see http://www.verisign.com/products/site/secure/Secure-Site.pdf

Performance Issues

Background

CPU resources required to accommodate varying loads, both with and without security, is an important Best Practices issue. When developing applications where cryptography is required, it is important to get an early estimation regarding the CPU and other resources required for volume production systems. Applications are often developed without adding SSL until late in the development cycle, and unpleasant performance surprises frequently occur. Using https with SSL can increase CPU requirements by 10 to 100 times, compared to http without SSL.

Reverse proxies and special http-to-https conversion hardware appliances offer the prospect of significant performance gains by shifting SSL processing away from web servers and WebCaches. Oracle is currently investigating these options and will report on them in a later version of Security Best Practices.

Below are some rules of thumb for SSL in general, relevant for conventional Wintel and Sun processors of around 200 MHz. Because SSL can run on many different platform types in many different implementations, our predicted results should be taken as neither definitive nor particularly accurate for any particular Oracle product. Your results may vary by one half to one order of magnitude.

- Measured in terms of http versus https pages per second, use of SSL can slow down sites by as much as two orders of magnitude. This assumes only one modest sized (15K or less) http or https message is sent every few minutes. When many pages are sent or received from a single browser in a small time period (two minutes or less), caching of bulk encryption keys can reduce the performance difference between http and https. But you should expect at least a 10-to-1 slowdown when using https for normal traffic, and you should **always** test reasonable load scenarios.
- Use of SSL adds a latency of several hundred milliseconds, in part because of additional CPU utilization and in part because significant extra network traffic occurs. However, this does not directly translate into throughput numbers.
- Bulk encryption key size makes little difference in performance when using RC4, because it always uses 128 bit operations. With a 40 bit key, we just know what the other 88 bits in the key are.
- RC4 bulk encryption is about 8 times faster than DES and about 25 times faster than 3DES. Even with a slow Pentium (150MHz), RC4 encrypts 8.5M bytes per second or about 2 milliseconds for a modest sized (15K) message.

Best Practices

Best Practice SEC-17: Performance test applications during development.

Performance testing **with SSL** early in the development cycle is prudent. Test during the prototype or feasibility stages if possible. Testing should emulate volume production.

Best Practice SEC-18: Ensure that sequential https transfers are requested of the same web server.

Expect several hundred milliseconds to be required to initiate SSL sessions on a 300 MHz machine. Most of this CPU time is spent in the key exchange logic, where the bulk encryption key is exchanged. Caching the bulk encryption key will significantly reduce CPU overhead on subsequent accesses, provided that the accesses are routed to the **same web server**.

Best Practice SEC-19: Keep secure pages and pages not requiring security on separate virtual servers.

While it may be easier to place all pages for an application on one https virtual server, the performance cost is very high. Reserve your https virtual server for pages needing SSL, and put the pages not needing SSL on a http virtual server.

If secure pages are composed of many gif, jpeg, or other files that would be displayed on the same screen, it is probably not worth the effort to segregate secure from non-secure static content. The SSL key exchange (the major consumer of CPU cycles) is likely to be called exactly once in any case, and the overhead of bulk encryption is not that high.

> **Note**: A single iAS server can accommodate many virtual servers. Some of these can be https virtual servers and others can be http virtual servers.

References

http://isglabs.rainbow.com/isglabs/shawn/SSL_Perf/otpssl8.html is a good reference for http versus https performance.

Client Certificates

Background

In the near future, the identity of browser users will be authenticated more and more frequently by client certificates which comply with ITU X.509 version 3 specification. Client certificates contain the following information:

- Expiration date
- Owner's public key
- Owner's distinguished name (including organizational information)
- Trust point (issuer's name) which guarantees that the information in the certificate is valid.

Certificates are issued by a certificate authority (CA). Certificates are often issued in-house by large companies, but CA services can also be outsourced. As of January 2001, Verisign is the largest outsourcing agency of certificates, with about 90% of the market. Other players in the market include Thawte, EnTrust, and GTE Cybertrust (part of Baltimore Technologies).

CA standard practice is first to issue certificates and then to keep revocation lists. This allows certificate validation to be handled in the same manner as credit card validation: a certificate is presumed valid if it is not expired, and if it is not on a revocation list. Revocation lists are usually maintained at a central server managed by the CA. They can also be replicated or partially replicated, depending on the security policies of the CA.

Certificates are relatively new instruments for providing user authentication. Firms may be tempted to build policies similar to those already in place for usercodes. The traditional usercode model works well for intranets, but it may work poorly in the evolving world of outsourced Internet services. If each service provider were to create its own certificate building rules, users would be forced to have many different certificates, one for each of the provided services.

It would be very difficult for users to manage multiple certificates within one organization with the infrastructure currently available, especially if multiple certificates were required for a single transaction (because it included services from a number of organizations). When creating and managing certificates, assume that users will have one certificate for personal use and at most one for each organization where they work.

The Oracle9iAS HTTP Server has facilities for certificate handling, including revocation lists. The Oracle9iAS HTTP Server allows use of certificates for authentication via SSL and for authorization using the Oracle9iAS HTTP Server's URL Wildcard scheme. URL Wildcards are specified with an authorization that can depend upon any of the fields of authenticated certificates. Certificates received via the SSL mechanism encounter a series of checkpoints, each of which must be passed before proceeding to the next:

1. The expiration date is checked, and its issuer is checked against the list of valid trust points configured for that Oracle9iAS HTTP server.
2. The revocation list is checked to ensure that the certificate is not on it.
3. The URL is examined for matches against various configured URL Wildcards.
4. Each URL Wildcard has a list of allowable valid certificate patterns. A certificate pattern can include any of the fields of a certificate. This could include the issuer, the organizational unit of the owner, the owner's name, or fragments of any of these fields. If a match occurs, the operation proceeds and the URL is processed.

Authentication using X.509 certificates (rather than usercode / password) offers several interesting possibilities, especially when viewed from the perspective of the relatively new market of outsourced CA services:

- Certificates could be handled like traditional usercodes. Each user (whose identity would include organizational unit and issuer fields) would be explicitly entered in access control lists. Membership in such lists would be required to allow access to protected services.
- Particular certificate fields (such as the issuer or organizational unit) could be used to authorize access to services without regard to the certificate holder's distinguished name. Thus, Gartner might allow anyone at

Oracle to access their market research information, even though they might not specifically know that the person with a particular distinguished name was an Oracle employee. Also, Oracle might allow anyone whose issuer was “Oracle” to have access to certain Oracle services.

- A revocation list can be used several ways. It could be used, for example, only to exclude certificates whose private keys were compromised or for people having left an organization. On the other hand, it might be used instead to remove both invalid users (compromised keys and those who leave) and otherwise valid users (employees on probation). There is nothing which prevents organizations from adding and deleting certificates from revocation lists.

Best Practices

Best Practice SEC-20: Ensure that certificate organization unit plus issuer fields uniquely identify the organization across the Internet.

One way to accomplish this would be to include the Dun and Bradstreet or IRS identification as identification for the issuer and the organizational unit within the certificate.

Best Practice SEC-21: Ensure that certificate issuer plus distinguished name uniquely identify the user.

If the combination of issuer and distinguished name is used as identification, there must be no chance of duplication. Note that authentication based on the public key may be a poor idea, because you would have to revoke it if the private key were compromised. Public key authentication is risky even if you expect to use the certificate only within an intranet, because you may later decide to outsource services.

Best Practice SEC-22: Include expiring certificates in tests of applications using certificates.

Expiration is an important consideration for a number of reasons. Unlike most usercode-based systems, certificates expire automatically--typically (but not necessarily) within one year. With longer duration certificates, fewer reissues are required but revocation lists become larger.

If expiring certificates are not included in tests of application systems using certificates, they can become time bombs of bugs. Consider the following examples:

- If processing is distributed over several time zones, certificates might be simultaneously authorized on one system and not on others, due to differences in system clock and assumed GMT.
- The replay of a transaction might fail if a certificate expires between the initial transaction and the replay.
- An expiring certificate might provide authorization during early stages of a transaction but fail later.

In systems where certificates replace traditional usercodes, these situations may result in unexpected bugs. Careful consideration of the effects of expiration is required and new policies will have to be developed, because most application and infrastructure developers have not worked in systems where authorization might change during transactions.

Best Practice SEC-23: Use certificate reissues to update certificate information.

Because certificates expire, infrastructure for updating expired certificates will be required. Take advantage of the reissue to update organizational unit or other fields. In cases of mergers, acquisitions, or status changes of individual certificate holders, consider reissue even when the certificate has not yet expired. But pay attention to key management. If the certificate for a particular person is updated before it expires, for example, should the old certificate automatically be put on the revocation list?

Best Practice SEC-24: Audit certificate revocations.

Revocation audit trails can help you reconstruct the past when necessary. An important example is replay of a transaction to ensure the same results on the replay as during the original processing. If the certificate of a transaction participant was revoked between the original and the replay, the audit trail enables certificate "unrevocation".

Note the quote marks. We are not recommending actual "unrevocation" operations, which can significantly complicate the management of systems employing certificates. It may be best to disallow "unrevocation" operations in production systems entirely, even though certificate infrastructure may allow them.

Appendix: Supporting Information

This section contains supporting information for some of the best practices described in other sections of this document.

New EXPORT versus DOMESTIC Encryption Issues

For years U.S. developers have had to cope with building DOMESTIC and EXPORT versions of products containing cryptographic technology. In many cases, getting products through the complex process of certification for export was so difficult that it limited other areas of functionality as well.

Changes to US export law in early 2000 have eliminated the requirement for EXPORT and DOMESTIC versions of products containing cryptographic functions. But each new version of software containing cryptographic functions must still have a one-time U.S. government review before shipment outside the U.S. and Canada.

As of June 2001, rules for products containing encryption are roughly:

- Products that are never to be distributed outside the U.S. and Canada can be released without government review.
- No shipments of products containing cryptographic functions are allowed to countries deemed “sponsors of international terrorism” by the U.S. State Department, without specific State Department approval . As of May 2001, these countries include: Cuba, Iran, Iraq, Libya, North Korea, Sudan, and Syria.
- New versions of products planned for delivery to other (that is, non-terrorist) countries outside the U.S. and Canada may require a one-time review by U.S. State and Commerce Departments (or their agents), depending on the cryptographic functions which they employ. While general guidelines are provided here, all products containing any cryptographic logic should be reviewed by corporate export control.
- Products containing no encryption or whose encryption is limited to nondata items like usercodes and passwords can be shipped without restrictions, licenses or reviews. Examples of cryptographic functions which can be used without restriction or license include digital signatures and authentication technology in general. These functions are exempt because they could not easily be adapted for keeping the *content* of messages or data confidential.
- Products containing encryption technology which could be used to send encrypted data or messages must be reviewed. This includes new versions of such code any time new or substantially changed versions of cryptography functions are issued.

These reviews generally take six weeks or less if new cryptographic techniques are not employed. Using encryption technology such as SSL (which has already been reviewed) will usually shorten the review. Developing or using a new encryption technology or a new usage of existing technology can lengthen the reviews to many months.

Note that shipments to military organizations and some other governmental organizations may require special licenses for every shipment. This should not be an issue for developers as long as the one-time review has been performed.

References

- http://cwis.kub.nl/~frw/people/koops/lawsurvy.htm is a very good reference for international cryptography laws.