# *A Step-by-Step Guide to Securing Windows 2000 for Use as an Internet Server*

Prepared as partial fulfillment of the requirements
for the GIAC Certification Course for Windows 2000 Security

David S. Courington
March 29, 2001

# Table of Contents.

# Introduction.

Windows 2000 Server is the latest iteration of Microsoft's premier Network Operating System (NOS). Many advances in technology are built into this new version, not the least of which is the integration of Internet Information Services into the core of the Operating System (OS). Earlier versions of Windows NT have a somewhat functional version of IIS loaded on the CD with the software, but the version included with Windows 2000 is tightly integrated with the OS in an effort to posture Windows 2000 as a competitor in the Internet Server market. As a result of this tight integration, IIS 5.0 allows the use of many operating system services to "ease" administration. One of the areas where this is most evident is in the security aspect of IIS 5.0. Windows 2000 has many new security features; Certificate Services, Digest Authentication, Fortezza Authentication, Kerberos v5, and NTLM. The complexity built in to the OS and all the new security features makes both Windows 2000 and IIS 5.0 more difficult to secure properly.

Microsoft has long been known for Wizards and automation tools used to configure their software, but security, especially for an Internet Server, should not be left totally to wizards. There are a few tools that are available, but many of them have limited functionality and are "Use at Your Own Risk." Two of these tools are available for IIS 5.0 Security and can be downloaded from Microsoft:

The Windows 2000 Internet Security Configuration Tool
(http://www.microsoft.com/Downloads/Release.asp?ReleaseID=19889)

The IIS Security "What If" Tool
(http://www.microsoft.com/technet/support/kb.asp?ID=229694)

There also are many articles, books, documents and white papers written on this topic. A reference of the ones used in preparation of this document will be listed at the end.

# Background.

This project is based upon using written documentation, white papers, and a general working knowledge of Windows 2000 security since many of the tools available to automate the security configuration of IIS 5.0 are still in Beta and are considered to be "Use at Your Own Risk," or are unsupported.

The following document will describe how to configure and implement Windows 2000 Server and IIS 5.0 with a reasonable amount of security. As we all know security is a never-ending task. As soon as one vulnerability is fixed, another is discovered. Also, making a system completely secure from attack while it exists on a public infrastructure is impossible. The best you can hope to do with security is to harden the system to the point that the attacker will look for an easier target. All reasonable attempts have been made to ensure that this document is correct and up to date. This document also assumes that the reader has a basic working knowledge of Windows 2000, IIS 5.0, and general network security practices. This document is not designed to be a "catch all" for establishing a secure IIS 5.0 Server for all applications. However, it does attempt to provide a good basis for establishing an acceptable level of security when preparing a Windows 2000 Server and IIS 5.0 as a web server to be deployed on the Internet.

This document is structured as a "Step-By-Step" guide. However, as stated above, it is assumed that the reader has a basic working knowledge of Windows 2000 and general security concepts. With this assumption made, this document uses some terminology and references that require this knowledge.

## General Security Considerations.

Before beginning the installation of the OS there are many tasks to be performed. Following is a checklist of items that should, at minimum, be addressed prior to preparing a server for use on the Internet.

*Planning*

The process of planning an installation is a topic worthy of its own paper and, as such, will be discussed only peripherally here. If you need a good basic guideline for planning your installation, refer to the *Site Security Handbook* (RFC 2196). If one thing is for certain in an installation, planning is everything. The process of securing an Internet server needs to be planned out in detail. This is not the type of installation that you can perform by making selections off dialogue boxes as they pop up during the install process. Decisions concerning the function of the system and the goals for this system need to be worked out prior to installation. This planning guide needs to be as comprehensive as possible because it will serve as your roadmap to installation, your troubleshooting guide, the basis of documentation for your server installation, and possibly your perimeter network.

*Policies*

Policies and planning go hand in hand when it comes to deciding not only what functionality a server is going to perform, but who has access to the server, what data will be stored on the server, and what actions are to be taken in various situations that might arise. Policies allow you to define the interaction that takes place between the organization the server is functioning for and the Internet public it is delivering services and data to. For a truly secure site, it is necessary to have the proper policies in place. Again, the *Site Security Handbook* (RFC 2196) is a great resource to help in structuring your corporate policies for this type of installation.

*Access Control*

Plans need to be made to address access to the server prior to its installation. To this end, there are three different categories of access control that need to be considered when designing a secured site.

*Physical Access Control*

In the world of network security, Physical access can be defined as the ability to actually touch and interact with the console of the server. This type of access is most often overlooked or it is simply unavailable. Smaller organizations may not have the physical resources (space) to provide a secure location within their organization to "lock down" or prohibit unauthorized physical access to the server. Many of the security measures that are in place can be circumvented if an attacker obtains physical access. Access to the console of a server negates the share permissions implemented on the server and other network security measures that you might have in place. It is generally acknowledged that if an attacker gains physical access to the console of your server, you have big problems with your security plan.

*System Access Control*

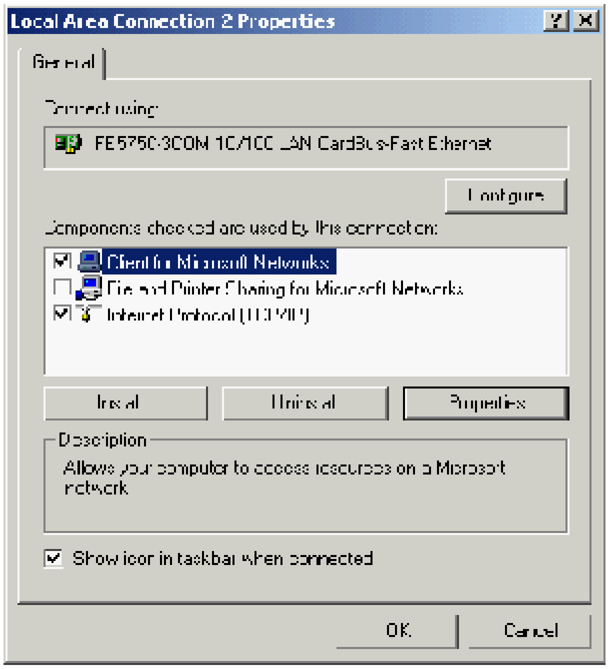
Part of the policies implemented for this installation should outline the personnel that have system level access to the server. This system level access should include a listing of groups and/or individual user accounts that are to be granted access to the server to perform specific functions such as backing up and restoring data, publishing documents to the web server folders for distribution, and those accounts and groups that are able to administer accounts and/or the server itself. The level of access granted to these various persons and groups should be the minimum amount of permissions needed to perform their functions. Not everyone needs to be allowed access as an administrator.

*Network Access Control*

Network access control must be very carefully planned and implemented. This is where the Internet and the server interact, and this is what dictates the level of access that individuals, both inside the company and those from the Internet, have to data and the resources of your server. A great deal of attention needs to be paid to this aspect of access as this is the area where ports are accessed, data is read and written, services are used, etc. Failure to protect and/or block access to certain things on your system can result in a security vulnerability waiting to be exploited. Further, not providing the proper level of access to these same items can result in your site being vulnerable to attack due to security levels being too low, or unusable to its target audience because security levels are too high. There is a balance to be found at this part of your implementation, and your documentation of policies and procedures should cover many of these items. Remember, attackers are not always from the outside; hackers may be the very users that you work with on a day-to-day basis. Threats to business data come from both sides of the network, inside and outside. To protect against these threats, it is important to locate the server in a secured network area, called a DMZ or perimeter network, which has restricted access to the server from both sides. Auditing, which will be covered later, also allows for the tracking of attempts to access the server from both sides.

# Installing and Configuring Windows 2000 Server.

When starting the installation of Windows 2000, performing a normal installation up to the point of configuring the network protocols is identical to any other Windows 2000 installation. When configuring the protocols for use on an Internet server it is important to close a large security hole: NetBIOS. This is done by selecting "Custom" in the Network Settings dialogue box. When the Custom dialogue appears, you should deselect every item except "Internet Protocol" and "Client for Microsoft Networks." (Figure 1.) The reasoning behind keeping the Client for Microsoft Networks option selected is that the NTLM (NT/LAN Manager) Security Support Provider component is embedded in this part of the OS. Without this component, Internet Information Server (IIS) will not run. Ideally, this server should be configured with two network interface cards (NICs), which will allow you to set up one for the connection to the Internet, and the second to be configured for accessing the server from your Local Area Network (LAN). With this configuration, you can disable and unbind services and functions not needed on the Internet Connection, and still have these items available for interaction with the server from your LAN.



Connect using:

FE575C-3COM 10/100 LAN CardBus-Fast Ethernet

Components checked are used by this connection:

☑ Client for Microsoft Networks
□ File and Printer Sharing for Microsoft Networks
☑ Internet Protocol (TCP/IP)

Description

Allows your computer to access resources on a Microsoft network

☑ Show icon in taskbar when connected

*Figure 1. The Windows 2000 Network Interface Configuration Dialogue Box.*

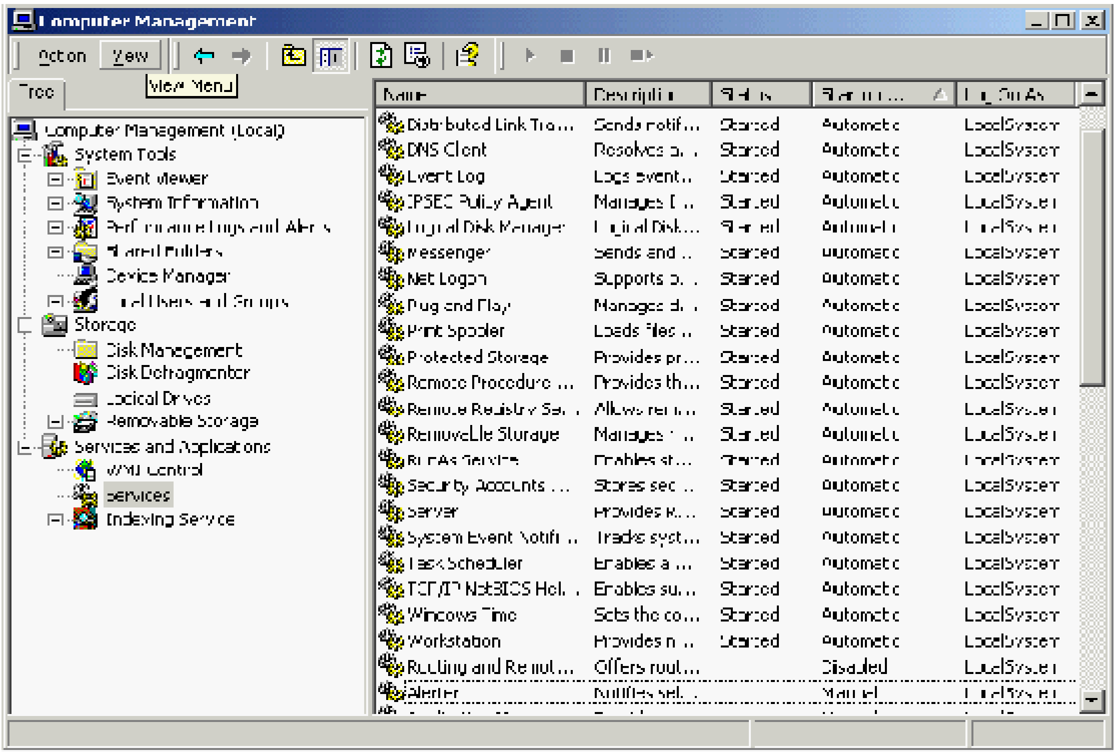
This is not the only configuration to be done on the Internet Protocol; you also must select "Internet Protocol" and go to the properties of this component. In the properties dialogue, you should set up the IP address for the Network Card and the Default Gateway as well. Additionally, you should go to the Advanced Section of TCP/IP Configuration and select the WINs Tab in the Advanced Dialogue. On this tab you should select "Disable NetBIOS over TCP/IP." This will stop the NIC from sending and receiving NetBIOS names to the Internet. NetBIOS names are the simple, friendly, machine names that we use on our network to allow easy UNC mapping of resources. A UNC Map is characterized as: \\servername\resource. If this is left open, utilities such as port scanners will detect the machine answering on Ports 137 and/or 139, known to be heavily used by the Microsoft OS, and they can then begin attempting to gain access to resources via NetBIOS Naming conventions on your network. It is generally known that servers listening on Port 137 and/or 139 are likely to be a Microsoft based OS, and therefore, a target for hackers who want to attempt to gain fame through exploiting security holes published about this software.

After completing the installation of the OS, there are several other items to address that are necessary to create a secure server for the Internet. First, install the Microsoft High Encryption Pack. This will upgrade your server to 128-bit encryption. It is not known why this is not the default for software distributed in the US and Canada, but the server software installs with a lower level of encryption by default and you must manually configure it for 128-bit encryption. This should be done prior to creating accounts, groups, etc. so that you can be certain that all items you are setting up on the server are at the 128-bit level. After this is done, installing the latest Service Pack from Microsoft should be the next step. These Service Packs contain fixes for known security holes, bugs, etc. that pertain to all aspects of the software. Microsoft also publishes "Hot Fixes" for items that are found to be what they consider serious problems that arise between the releases of a major Service Pack. Each Service Pack release is supposed to contain all previous Service Release fixes, plus any Hot Fixes, etc. that were released between Service Packs. These Service Packs can be found on the Microsoft Web Site at:

(http://www.microsoft.com/windows2000/downloads/default.asp)

This same location is home to a vast amount of Windows 2000 information, including Hot Fix announcements, etc. This is definitely a site to bookmark as a reference for the day-to-day management of Microsoft Windows 2000 Servers. One thing to remember about Hot Fixes, it is wise to only install the Hot Fix if your system needs it. Not all Hot Fixes are necessary on every server. Many of them are to fix specific vulnerabilities that exist only in certain configurations, so it may not be applicable to your configuration.

After applying the Service Pack and any needed Hot Fixes, it is time to perform more configuration on the server. These changes will be made at the Services Level. By default, many services are loaded and are set to automatically start with the system. Also, many of these services are set to start with a System Account, this is very dangerous as the system account is a very high level and powerful account with many rights and permissions that might not be necessary to do the job that service is designed for. You can look at a listing of all services that are loaded on the system, their current status, and their logon type by going to the Computer Management MMC Plug-in (Microsoft Management Console). (Figure 2.) There are certain services that you should have configured to start automatically. It may depend on your configuration and what your server is designated to do, but following is a list of the basic services that should start automatically.



| Name | Description | Status | Startup ... | Log On As |
|---|---|---|---|---|
| Distributed Link Tra... | Sends notif... | Started | Automatic | LocalSystem |
| DNS Client | Resolves a... | Started | Automatic | LocalSystem |
| Event Log | Logs event... | Started | Automatic | LocalSystem |
| IPSEC Policy Agent | Manages I... | Started | Automatic | LocalSystem |
| Logical Disk Manager | Logical Disk... | Started | Automatic | LocalSystem |
| Messenger | Sends and ... | Started | Automatic | LocalSystem |
| Net Logon | Supports p... | Started | Automatic | LocalSystem |
| Plug and Play | Manages d... | Started | Automatic | LocalSystem |
| Print Spooler | Loads files ... | Started | Automatic | LocalSystem |
| Protected Storage | Provides pr... | Started | Automatic | LocalSystem |
| Remote Procedure ... | Provides th... | Started | Automatic | LocalSystem |
| Remote Registry Se... | Allows rem... | Started | Automatic | LocalSystem |
| Removable Storage | Manages ... | Started | Automatic | LocalSystem |
| RunAs Service | Enables st... | Started | Automatic | LocalSystem |
| Security Accounts ... | Stores sec ... | Started | Automatic | LocalSystem |
| Server | Provides R... | Started | Automatic | LocalSystem |
| System Event Notifi... | Tracks syst... | Started | Automatic | LocalSystem |
| Task Scheduler | Enables a ... | Started | Automatic | LocalSystem |
| TCP/IP NetBIOS Hel... | Enables su... | Started | Automatic | LocalSystem |
| Windows Time | Sets the co... | Started | Automatic | LocalSystem |
| Workstation | Provides n... | Started | Automatic | LocalSystem |
| Routing and Remot... | Offers rout... |  | Disabled | LocalSystem |
| Alerter | Notifies sel... |  | Manual | LocalSystem |

*Figure 2. The Windows 2000 MMC showing the Services selection.*

- DNS Client. This is needed if the server you are running needs to be able to initiate communication with other servers. Many web servers only answer requests and do not make them. If your web server does not make requests, DNS Client services are not needed.
- Event Log. This service allows for the collection and logging of data from the server. This includes any auditing that has been configured, any system events, etc.
- Logical Disk Manager. This service allows the management of local disk drives and removable devices.
- Network Connections. This service allows the management of the Network Interface Cards and their properties, etc.
- Protected Storage. This service provides protected storage for sensitive data such as private keys, etc.
- Remote Procedure Call (RPC). This allows programs on one system to execute instructions or programs on a remote system.
- Security Accounts Manager (SAM). This service manages the local user account database.
- Windows Management Instrumentation (WMI). This service is required by the MMC. Without it, there is no management console to access to perform system management.
- Windows Management Instrumentation Driver Extensions. This service also is required by the MMC.

Other services that should be started, but set to start manually are:

- Logical Disk Manager Administrative Service. This allows for the administration of the local disks and removable devices.
- IIS Admin Service. Allows or the management of Internet Information Services.
- World Wide Web Publishing Service. This service actually publishes web content to the port specified in the web site setup (usually Port 80)

The key to really managing many of these services is to apply the least privileges rule. This rule is considered to be a basic tenant of maintaining a secure setting on a server connected to the Internet. The basic premise of this rule is that a service or application should be designed to run on only the privilege level it needs to execute properly. In theory this sounds like a great idea, but Microsoft designed IIS to be tightly integrated with the OS. This integration dictates that IIS be allowed to function at the system level on the system it is installed on. The repercussions of this are evident when we examine what a local system account is capable of; rebooting the system, deleting partitions and file systems, executing code and applications, etc.

The main reason that Microsoft implemented this system was to allow IIS to take full advantage of the functionality of Windows 2000. All the access control mechanisms, etc. are handled by Windows 2000 directly, so that there is no code dedicated in IIS for this functionality. This makes the setup of IIS and the administration more user-friendly, but when it is examined beneath this surface, it becomes something akin to a nightmare. Dealing with all the possibilities for intrusion, security breaches, etc. can make for a difficult time in really securing this application/OS combination.

Even after all this, we still are not through with our initial hardening of the OS. The preceding steps allowed us to secure portions of the system, but there are several other areas to deal with. Securing the System Accounts Database (SAM database) against the variety of tools that are used to crack this database and gain access to the user account information and passwords contained in it is a major issue to deal with. There are many tools on the Internet that will allow a hacker to crack, or un-encrypt, the user database on a Windows 2000 system. By default the SAM is encrypted using a locally stored startup key. This key contains a hash code that is processed during startup to allow the accounts database to be unencrypted and stored in memory where it can be accessed by the system. This default storage place for the encryption key can be changed with a Windows 2000 command line utility named ***syskey.exe***. This tool allows the administrator to change the location of the hash code to a floppy disk, or other location on the system. A floppy disk is the preferred location for a secure server. This disk needs to be backed up in several locations and stored in a very secure location. If the disk is lost or goes bad, the server cannot be restarted as it has no way to un-encrypt the database of user accounts and passwords. One other task to be performed on the SAM database is to implement password complexity requirements. This is done through the Local Security Policy Tool on the Start → Administrative Tools menu. On this menu there is an option to set the account policy and the password policy is defined within this. All that is required is that you select "Passwords must meet Complexity Requirements" from the menu. This option has several different libraries that can be used to force users to establish various levels of password complexity. Microsoft has one named "*passfilt.dll*" that provides for basic complexity requirements. Another is the Quakenbush Password Utility. This utility implements fairly strong password complexity requirements and is considered to extremely secure. Password policies are a necessity in today's interconnected world. At this point in the document it seems right to establish the guidelines for secure password policies.

There are rules that dictate the level of complexity for a password policy. These rules are based on the type of data, the importance of the data, and the risk of exposure the data has to the outside world. Generally, a strong password (7+ characters) made up of random alphanumeric and special characters that is changed on a regular schedule is a good basis for password policy on any server exposed to the Internet. There are many, varied, thoughts on proper password policy for web site access. Based on the circumstances that the server is created for and the type of information that it is required to distribute (i.e. Is the server used only as a public information site; does it provide client access to secured data; does the site require users to submit data to it; the list can go on and on for every scenario, but you get the idea). There are different levels of security that can be implemented for each of the possible site types, it is only through practice, study, and implementation that we can better decide on what works for our servers. All Internet servers can be set up using the same basic security techniques, but, when the decision as to what type of services are to be provided is made, there are certain security decisions that will have to be made. If a server is to simply provide a web presence for your company, it might be best to allow only anonymous access to the web site. This will disallow passwords being passed from the server and client. Without allowing any other logon than anonymous, the server will not accept any attempted logon, reducing the chances that an individual will be able to pass account and password combinations to your server in an attempt to gain higher than guest level access. One last item on passwords; it is good practice to rename the administrator account and set a password on it that includes extended ASCII characters.

In Windows 2000 there exists the possibility for an attacker to log on to your system with what is called a null user account and establish a null session. This session can be used to obtain a listing of the user and group accounts on the server, as well as get a listing of all shares on the server. There are ways to prevent this from happening. This prevention is accomplished through registry modifications. Following are the keys and values that must be changed to prevent attackers from being able to use this vulnerability to obtain information from your system.

To block null sessions from obtaining a username listing and the share name listing from the server make the following modifications:

| | |
|---|---|
| Hive: | HKEY_LOCAL_MACHINE |
| Key: | \System\CurrentControlSet\Control\LSA |
| Value Name: | RestrictAnonymous |
| Value Type: | REG_DWORD |
| Value: | 1 |

Be aware that setting this registry value may cause certain network services to malfunction. Some services use these null sessions to perform tasks on the network and to contact other servers, systems, etc. If these types of connections are disallowed, the services cannot perform their specified task. This type of setting is best tested in a non-production environment prior to putting on a production server.

You may also want to consider modifying null session access to shares on the system. There are two ways to do this: RestrictNullSessAccess, a registry value that toggles null session shares on or off. This is accessed at the following location:

| | |
|---|---|
| Hive: | HKEY_LOCAL_MACHINE |
| Key: | \System\CurrentControlSet\Services\LanManServer\Parameters |
| Value Name: | RestrictNullSessAccess |
| Value Type: | REG_DWORD |
| Value: | 0 or 1 |

When set to 1, null session users cannot access any shares. If it is set to 0, the null session user can attach to any share or printer that is shared to the Everyone Group. This registry edit also affects null session access to Named Pipes. Named Pipes provide a way for processes on one system to communicate with another process on a different system. There are many named pipes that are set up in Windows 2000; one of them is called Winreg. This functions as an IPC mechanism to allow Regedit to be run on a client machine and access the registry on a remote server. Netlogon uses an RPC Connection through a named pipe to do logon authentication. Server Message Blocks (SMB) use named pipes for communication between servers on the network. There is a listing in the registry that contains all the named pipes that are accessible to null session users. The location of this registry table is:

| | |
|---|---|
| Hive: | HKEY_LOCAL_MACHINE |
| Key: | \System\CurrentControlSet\Services\LanManServer\Parameters |
| Value Name: | NullSessionPipes |
| Value Type: | REG_MULTI_SZ |
| Value: | <List of Named Pipes Available to Anonymous Users> |

Any named pipe listed in this registry list is accessible by null session users. It is possible to remove certain named pipes from the list, but you should always research the pipe and what applications and services are actually using it. Removing them from the list may cause these applications and services to stop performing their functions. Many system accounts use these named pipes to communicate with other systems on the network.

Removing administrative shares on server drives also is something to be considered. These shares are hidden shares set for administrative use only. They are named C$, D$, etc. When accessing one of these shares, the results displayed are the entire listing of directories and files stored on the root of that hard drive. These shares are accessible only by the administrator and the local service account. These shares can be disabled or removed from the system, but the removal of them may have adverse effects on things like backup software, etc.

Even after all this, there is still more to be done. Microsoft built Windows 2000 to be flexible in its handling of other OS applications; namely OS/2 and POSIX. These are referred to as subsystems in Windows 2000 and they allow certain OS/2 command line applications and any POSIX 1.x applications to make requests of the server to execute their code. This is a large security hole for a web server. The simplest way to close this hole is to remove the subsystems, making them unavailable for use. This should not present any problem for the Windows 2000 server or for IIS because almost everything on a Windows 2000 server runs in the Win32 subsystem. To disable these other subsystems it is necessary to perform some rudimentary registry edits. Following is a listing of the edits and the order in which they should be performed as excerpted from Jason Fossen's text *Securing Windows NT, Step-by-Step* (Pg. 45).

To remove the OS/2 and POSIX subsystems, perform the following steps:

1. Delete the following folder and all of its contents: \%systemroot%\system32\os2
2. Delete al the subkeys underneath \HKLM\Software\Microsoft\OS/2 Subsystem for NT
3. Delete value Os2LibPath in \HKLM\System\CurrentControlSet\Control\Session Manager\Environment
4. Clear the contents of Optional in \HKLM\System\CurrentControlSet\Control\Session Manager\Subsystems (but leave the value named Optional itself in place)
5. Delete the Os/2 and Posix subkeys in \HKLM\System\CurrentControlSet\Control\Session Manager\SubSystems
6. Reboot.

Another item that should be dealt with is the location of page files on the system. Page files are used by Windows 2000 as a temporary holding area for application code, etc. when an application or system process needs to access physical RAM. Page files must have space on the hard drive to expand. If the page file runs out of space, it will cause the system to crash. There are several ways to avoid this situation:

- Install as much RAM on the system as is feasibly possible. The more physical RAM available, the more efficiently the system can perform.
- Put all Operating System files on their own partition. This partition should contain only the OS files and a page file at least the size of physical RAM. This page file will allow the system to create a crashdump file when the system encounters a STOP error.
- Create a page file on at least one other partition that is the size of physical RAM + 11 MB. If possible, place this page file on a partition that is on a separate physical drive. This will allow the system to perform I/O operations much more efficiently as it can write to that partition while it is performing another read/write operation on the system drive.
- Configure services and applications that generate log files and any other expandable data files to write their files to folders on a drive other than the one that the OS exists on. Also, make sure that the location you choose has enough space to allow for as much expansion as these files may need.
- Configure the maximum size of your audit logs to be as large as necessary so that they do not fill up and start to either overwrite information, or stop the system from functioning since the log files are filled. A more in depth look at Auditing and a listing of tools available to aid in log file analysis is provided later in this document.

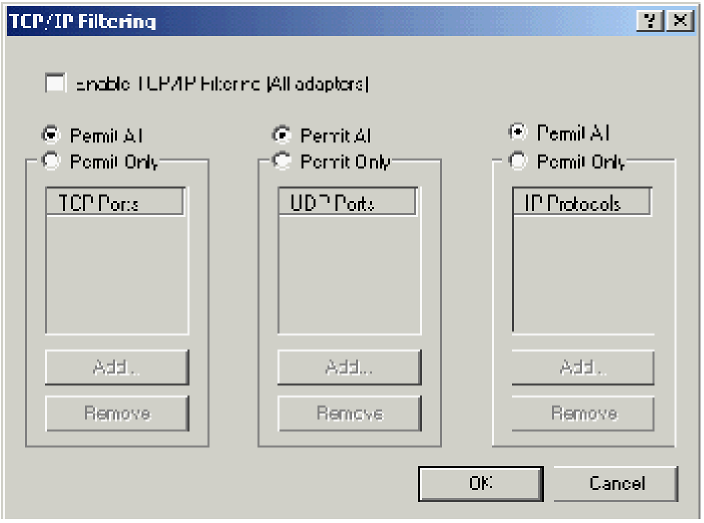
# Configuring TCP/IP Security Settings.

Windows 2000 offers three ways to perform access control on incoming network connections:

- TCP/IP Security
- Access Control Lists for incoming connections provided by the Routing and Remote Access Service (RRAS)
- IPSec Policy Filters which are enforced by the IPSec Policy Agent

TCP/IP Security is identical to the Windows NT 4.0 implementation. It is very rudimentary in its configuration, allowing access on an “All or Nothing” basis. Configuration of TCP/IP Filtering is performed by accessing the filtering dialogue from the Control Panel → Network and Dialup Connections → Local Area Connection → Properties → Internet Protocol → Advanced Options → TCP/IP Filtering as shown in Figure 3. This allows you to select Permit All (all traffic from all ports), or permit only (deny all port traffic unless specifically listed). This type of TCP/IP Security is generally the first type of port filtering that most administrators attempt to configure. For many web server applications this type of filtering will provide very basic protection. If you feel that you need more control over the settings, read up on the configuration of IPSec on Windows 2000. If your server is going to perform multiple functions on the web (i.e. FTP server, WWW Server, Telnet Server, Streaming Media Server, Chat Server, etc.)

you will need to configure multiple ports for your filtering as these services all use different ports in the TCP/IP stack to transfer information. TCP/IP Filtering is better than no filtering, but it is fairly inflexible and can cause problems with administration and authoring (publishing data to the web server).


TCP/IP Filtering

□ Enable TCP/IP Filtering (All adapters)

| TCP Ports | UDP Ports | IP Protocols |
|---|---|---|
| ● Permit All | ● Permit All | ● Permit All |
| ○ Permit Only | ○ Permit Only | ○ Permit Only |

*Figure 3. The Windows 2000 TCP/IP Filtering dialogue in Windows 2000.*

For a more fine-tuned and granular approach to TCP/IP Security, you may choose to implement IPSec on your Windows 2000 Server. This document does not go into detail on IPSec, as this topic is complex enough to warrant its own documented procedure for implementation. Suffice it to say that IPSec is a policy based traffic analysis service that compares traffic to a rule set and allows or disallows the traffic based on if it fits a predetermined set of guidelines that have been established for that type of traffic. These rule sets are called filter lists and can be designed around several different secure authentication protocols. These protocols are:

- The Internet Key Exchange Protocol (IKE)
- Authentication Header (AH)
- The Encapsulating Security Payload (ESP)

The third option for filtering is Microsoft's Routing and Remote Access Service (RRAS). This tool provides the ability to deploy much more flexible packet filtering (Figure 4.). Even thought this packet filtering is static, it allows you to establish filters based on packet direction, IP address, ports, and various protocol types (Figure 5.).

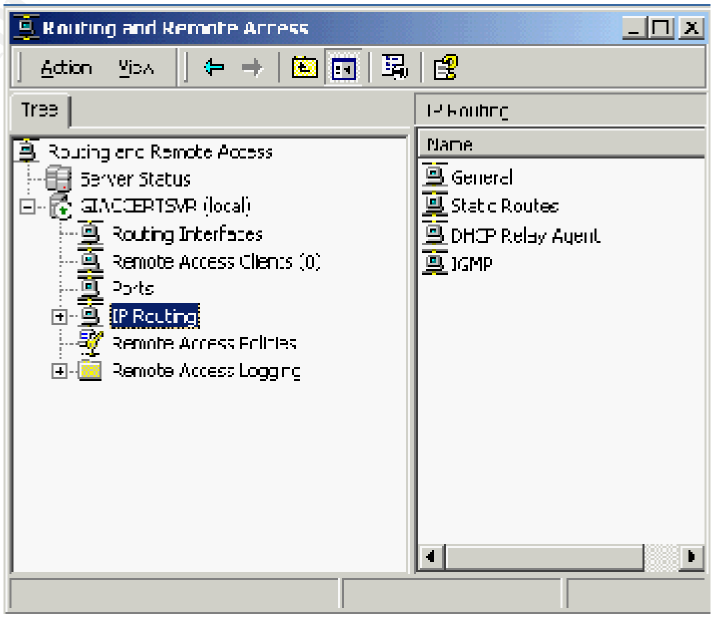
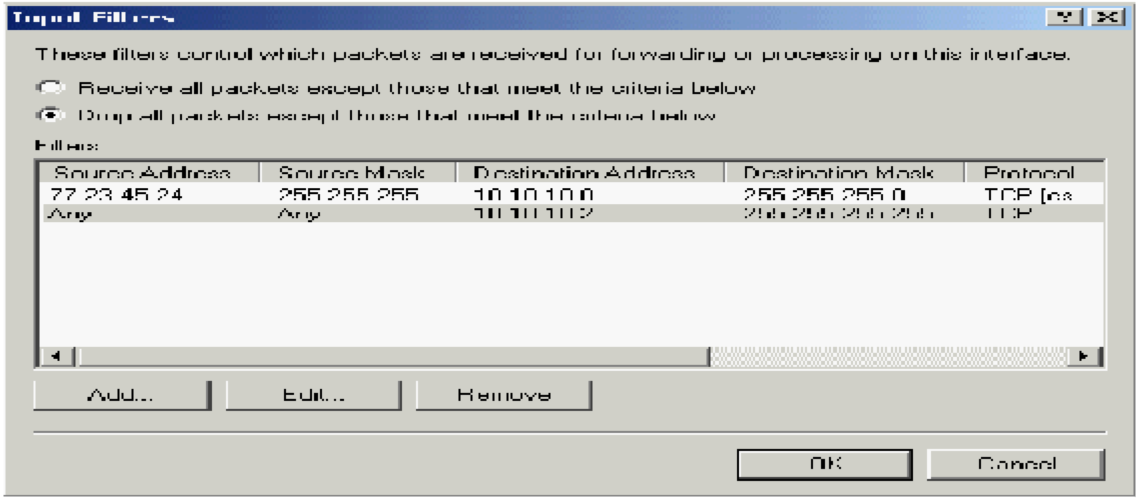


*Figure 4. The Windows 2000 MMC Routing and Remote Access Display.*



Input Filters

These filters control which packets are received for forwarding or processing on this interface.

○ Receive all packets except those that meet the criteria below

● Drop all packets except those that meet the criteria below

Filters:

| Source Address | Source Mask | Destination Address | Destination Mask | Protocol |
|---|---|---|---|---|
| 77.23.45.24 | 255.255.255 | 10.10.10.0 | 255.255.255.0 | TCP [es |
| Any | Any | 10.10.10.2 | 255.255.255.255 | TCP |

*Figure 5. The Input Filters Dialogue Box from the Routing and Remote Access Configuration IP Configuration Dialogue.*

This concludes the section on configuring the operating system. In the next section, we will go through the configuration of IIS 5.0.

## Configuring and Securing IIS 5.0.

In an ideal installation, IIS will be installed on a stand-alone server. That is, a server that is not a member of any domain. If the sever is not connected to any domain, it does not have to establish a Netlogon channel to a domain controller. This fact will aid in lower security risks associated with null user sessions being established through this link between the servers. Also, no authentication traffic has to be passed between the systems, lowering the possibility of passwords and logons being intercepted.

As discussed earlier, there are certain services that can be disabled on a system that is functioning solely as a web server. Following is a general list of services generally not needed on a web server. Adapted from Fossen's text *Securing Internet Information Server 5.0*.

- Alerter
- Clipbook Server
- Computer Browser
- DHCP Client
- Messenger
- Netlogon (This is not required if the server is a stand-alone system)
- Network DDE and Network DDE DSDM
- Network Monitor Agent
- Simple TCP/IP Services
- Spooler
- NetBIOS Interface
- TCP/IP NetBIOS Helper
- NWLink NetBIOS

With the services taken care of, it is time to look at the location of the web site root folder (wwwroot). This folder contains all the information for the default web site that is created when IIS is installed on the system. Earlier in this text, it was stated that there should be more than one physical drive in the system for performance reasons. This need for multiple drives is reiterated as we start to look for a location on the system for our web site. It is best if the web site is stored on a different partition than the OS. It is preferred to have the web site folders on a different physical drive altogether for performance and security reasons. Also, when setting up virtual, or redirected, folders on the web site, make sure that none of these virtual directories are redirected to the boot partition. There are several attacks that can be performed against a web server that allow the attacker to navigate the other folders stored on the partition that the folder they accessed is housed on. These same attacks allow the attacker to execute commands and/or scripts that they can upload to these folders. It is possible to use IIS to redirect web requests for information to a virtual folder that is actually a URL to a different folder on the IIS server, a share on a remote system on the network or on a different subnet altogether. One of the benefits of this is that the share name that is accessed from the web site can be a different name than the folder actually has.

It also is possible to set the root folder up on a different server and have the entire web server root folder stored in a secure location. At this point, the IIS server becomes a system responsible only for caching requests and the answers to those requests. This means that the server is basically generic and has no content stored on it. If this is the case, it becomes a simple matter to restore the server in its entirety from tape or other backup device. If the site is attacked and crashed, it is simple to restore it and be up and running with the same site in a short period of time (Fossen, 2000).

In addition to there being dangers associated with the location of the web root folder, there are many files and utilities in the OS that need to be deleted, renamed, or have NTFS Permissions set on them to protect against an attacker having a nice set of tools waiting on them when they get into the server. Following is a list adapted from Fossen's text *Securing Internet Information Server 5.0*.

- AT.EXE
- CACLS.EXE
- CMD.EXE
- CSCRIPT.EXE
- DEBUG.EXE
- EDLIN.EXE
- FINGER.EXE
- FTP.EXE
- ISSYNC.EXE
- NBTSTAT.EXE
- NET.EXE
- NETSH.EXE
- POLEDIT.EXE
- RCP.EXE
- REGEDIT.EXE
- REGEDT32.EXE
- REGINI.EXE
- REGSRV32.EXE
- REXEC.EXE
- RSH.EXE
- RUNAS.EXE
- RUNONCE.EXE
- TELNET.EXE
- TFTP.EXE
- TRACERT.EXE
- TSKILL.EXE
- WSCRIPT.EXE
- XCOPY.EXE

After these tools have been secured, you should check to see if some components of IIS were installed during the Windows 2000 installation process. These components pose a security threat to the server you are setting up and should be removed from the system unless it has been decided that they must be installed for functionality. The list that follows is short, but it has some components on it that can be used against your server and your network.

- Internet Service Manager (HTML). This is the web based administration page for the IIS server. It should not be loaded unless it is absolutely necessary to provide management via the Internet.
- Sample Pages and Scripts. These are installed with the server and should be deleted from the system after installation is complete. There are scripts in these samples that are designed to show the power and functionality of IIS, but these same scripts can be used to execute applications and to browse the server from the Internet, which is not a good thing.
- Windows 2000 Resource Kit or the IIS Resource Kit. These items are some of the best hacker tools around. They were written by the experts, and have many useful items for an attacker to use to extract information and wreak general havoc on your server.
- SMTP and NNTP. If you are not planning on having mail forwarded from this server, and are not planning on serving News Groups, remove these items from the server. If they are not needed, it is wise to remove them and close any security holes that might be provided by the existence of these services.
- Internet Printing. Windows 2000 provides users the ability to access printers via the Internet. This seems like a neat idea, but it opens up your system to a variety of possibilities to be exploited. The printers on your network are accessed via a web page; the administration of these printers also is accessed from a web page.

Simplicity is a rule to live by when planning and implementing a server to function on the Internet. The simpler your setup and configuration are, the easier the server is to secure and manage.

There are a few registry edits that need to be made to aid in keeping the Internet server from being crippled easily in basic attack schemes. One of these is SYN Flooding, or a Distributed Denial of Service Attack (DDOS). This type of attack uses incomplete TCP session request packets to siphon off as much of the servers resources as it can. By doing this, the attack keeps your server from responding to legitimate requests (denying service). This cannot be totally stopped by a registry modification, but it can be used to reduce the damage potential of a DDOS. The registry value to modify is:

```
HKLM\System\CurrentControlSet\Services\Tcpip\Parameters
        The value name to modify is SynAttackProtect
        The value type is REG_DWORD
        The value data is 0 by default; the value should be changed to 2
```

# Authentication Methods in IIS 5.0.

At this point the Operating System and the network are basically secure, but IIS has more security options that need to be addressed. One of the main things that IIS provides for is authentication. This is not to say that IIS is responsible for providing all the tools for authentication; we have already discussed how IIS and Windows 2000 are tightly integrated. One of the main items they share is authentication and the various methods that this can be accomplished. We have several methods of authentication available to us when implementing IIS: Anonymous, Basic, Digest, NTLM, Integrated (Kerberos + NTLM), Certificate, and Fortezza. For a basic web server, like the one we are implementing we can narrow the list of authentication methods that we need to choose from. We can set the server to use multiple types of authentication, allowing for different levels of access to different areas of the server. For example, on a single site we might have our public area, and an area that requires a username and password before access is granted. This area that requires a password can be set to allow only certain types of authentication on it as well.

*Anonymous Authentication*

Anonymous authentication allows the system to provide the account and logon credentials automatically; the user is not asked for an account name and password to gain access to the system. The server uses an automatically created account, the IUSR_Computername account, to log on all users of the web site. This may seem very unsecure, but it actually is more secure as there is no transmission of usernames and passwords. This reduces the possibility of this type of data being captured and used by an attacker. There is a down side to Anonymous logons; everyone logs on with the same credentials, there is no control over different levels of user permissions. This level though is perfectly acceptable for a site that provides data to the general pubic. There are a couple of things that you should do to further secure the anonymous logon account. First, disable the IUSR_Computername account; this will be left to sidetrack any would be attacker trying to use this account as a logon mechanism for the server. Second, create a new anonymous account and set up IIS to use it as the account for logging users in to the site. This account should have a very strong password (14+ characters, including numbers, letters and symbols constructed in as random a fashion as possible).

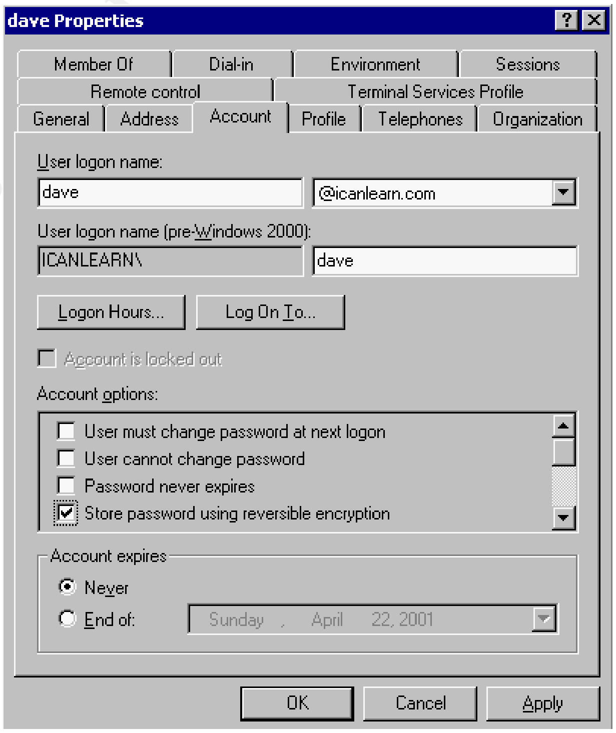
When creating this account, make sure it is a local account and not a domain account. This will negate the use of the IUSR account as a stepping-stone to move from the web server to other systems that it might be connected to. Also, if this account is local only, it is possible to disallow it being able to log on locally to the machine it is set up on. You can modify the properties of the IUSR account and remove the right to "Access this Computer from the Network" right. Without this right, a hacker will be unable to use this account to log on to the web server, even if they do obtain the password.

*Basic Authentication*

This type of authentication is derived from a Base64 encryption algorithm, but the implementation of this type of authentication sends the username and password to the server in plain text. This exposes the username and password to the Internet where it is possible to grab this information during transit and use it to initiate further intrusion to the target server and network. This type of authentication might also require that the user account have the right "Log on Locally" granted to it. If this account information falls into the hands of an attacker, they would have the right to log directly on to that server and use its resources to further their attack on the network. If this type of authentication is to be used on your system, you also should implement SLL Encryption. SSL will establish an encrypted channel between the client machine and the server prior to the transmission of any logon authentication routine.

*Digest Authentication*

This is a fairly new authentication method. Currently only Microsoft IIS 5.0 supports it on the server side when it is a member of a Windows 2000 domain, and on the client side, only by Microsoft Internet Explorer 5.0 and later. This method is being positioned to take the place of Basic Authentication. Its security features are far better than Basic. The encryption method (MD5) is far superior to that of Basic, but it has several drawbacks to it. The encryption method requires that Reversible Encryption be enabled on the account password (Figure 6), the logon is not transparent to the user, and, even though many vendors have pledged support to this new method, there has been very slow response to providing products that actually do use this method.



User logon name: dave @icanlearn.com

User logon name (pre-Windows 2000): ICANLEARN\ dave

□ Account is locked out

Account options:

□ User must change password at next logon
□ User cannot change password
□ Password never expires
☑ Store password using reversible encryption

Account expires

● Never
○ End of: Sunday , April 22, 2001

*Figure 6. The Windows 2000 Account Properties Dialogue Box.*

*Integrated Windows Authentication (NTLM + Kerberos v5)*

This authentication method is tied directly to Microsoft Browsers (version 2.0 and later) and IIS 5.0. It actually is a combination of the Windows NT Challenge/Response Authentication and Kerberos v5. These two methods work together, in parallel, to initiate secure, encrypted communication between the client and the server. This parallel action is limited in its functionality, as it requires that both the client and server be in the same domain or in mutually trusting domains (hard to do on the Internet). Additionally, the server must be a member of a Windows 2000 domain, the client must be running Windows 2000, and the browser must be IE 5.0 +. With all these requirements, it is easy to see that this is not really applicable to a web server on the Internet. It is a great method of authentication for an Intranet site, provided that your implementation meets the requirements.

*Certificate Authentication*

This type of authentication depends upon verification from a third party certificate authority. It uses a trust mechanism to verify that the party attempting to log on is who they say they are and they have the proper permissions to do so. This trust is not a domain trust, but, rather recognition that the company giving out the certificates is a recognized and Above-board Company that is capable of actually verifying that the person using one of their certificates is who they say they are. This type of authentication is accessible to IE 4.0 and Netscape v4. It does require that a certificate be installed on the client, and that the server being logged on to recognize that certificate authority as valid. This does provide for a transparent logon from the client side because, once the certificate is validated, the client is allowed access; no passwords have to be exchanged.

*Fortezza Authentication*

This type of authentication is not widely used as yet. It employs a combination of hardware and software (a smart card with digital signatures stored on it). This is a suite of security measures that currently is being expanded and brought into use on a more daily basis. A good example of this type of technology is the American Express BLUE® Credit card that has an embedded chip to store information about shipping, purchases, etc. on the card itself. This type of authentication is not at a level where it can be widely implemented for use on the Internet. The cost and complexity involved rule it out as an option for the majority of web servers in use today.

For the majority of web sites, Anonymous and/or Basic is what will be implemented as an encryption method. Most sites will simply be set up as anonymous as it is more secure than Basic.

## Permissions and Auditing in IIS 5.0.

There are file and share permissions that have to be set on folders and files from the Operating System, and there are file and folder permissions that have to be set in IIS itself. These two sets of permissions work in concert to secure your data and provide only the level of access that you desire your visitors/users to have. As a general rule, a folder should never have both Write and Execute Permissions set on it. This will allow an attacker to upload and execute malicious code on your site. Other permissions such as Directory browsing should be disabled. This will allow an attacker to look through the folders on your site and see what they can find to make mischief with. IIS has a set of Default Permissions, Script Only and Read. This will allow a user to execute a script that has been loaded to a directory to which they have been given Read and Script Only access.

It is a good practice to make directories for the various types of files that you will be using on your web site and assigning the proper level of permissions to them. For example, you might want to create a listing like the following:

> Scripts – Contains all script files (CGI, VBS, etc.) for your site, this folder has Script Only Permission set on it.
> BIN – Contains any files that must be directly accessed in the execution of something on the site. This folder should be set up with Script and Execute Permissions only.
> Static – This is the location for storing any static files like HTM or HTML.
> Active – This is the location for storing any Active Server Pages or DHTML Pages.

One last thing to do is to secure the IIS Metabase and the location that the metabase is backed up to. This is the item that controls almost every aspect of the IIS configuration. This includes passwords, etc., which are all stored in clear text within the metabase. There are a couple of fairly simple things to do that will allow you to secure the metabase. Following is a step-by-step procedure for this.

- Move all HTTP/FTP root folders off the %systemroot% volume (these should not be here anyway)
- Consider renaming and r moving the Metabase (difficult to do, think about this)
- Secure the registry key that determines the Metabase location
- Set up Auditing to log all failed access attempts to edit the Metabase
- Delete the file Iissync.exe from the %systemroot%\system32\inetserv folder
- Set new NTFS Permissions on the Metabase file to the following:
    - Administrators – Full Control
    - System – Full Control

To secure the backup location for the Metabase perform the following steps:

- Make a backup of the Metabase after you finish configuring IIS. This will create a folder called %systemroot%\system32\inetserv\MetaBack and store the file inside it.
- Audit all failed attempts to access the \MetaBack folder
- Set NTFS Permissions on the \MetaBack Folder to the following:
  - Administrators – Full Control
  - System – Full Control

One last detail to secure the Metabase is to secure utilities that can be used to edit the Metabase. Follow this procedure to complete this task:

- Move the \Inetpub\Adminscripts folder, which contain all the administrative scripts for IIS.
- Move Metaedit.exe and Metautil.dll from the \Program Files folder to the %systemroot%\system32\Inetserv folder and adjust the start menu shortcut.
- Audit all failed attempts to access the \Adminscripts folder.
- Set NTFS Permissions on the %systemroot%\system32\csript.exe. This is the executable for .VBS files. Set the permissions to: Administrators – Full Control.
- Set NTFS Permissions on the \Adminscripts folder to: Administrators – Full Control.

  With these tasks complete, the server is almost ready to bring online as a true Internet server. Before allowing access to the site, we must configure an Auditing routine for this server. Auditing is a necessity for Internet servers, this is the main method of determining if the site /server is under attack, or if either has been compromised. If you establish auditing on the system, it will serve no purpose if you do not set aside time each day to go through the logs and check for anomalous activity. Audit logs are only useful if you actually check them. This is a very difficult thing to set time aside for as there are many, more pressing, issues each day. You have to consider this: what if an attack has been mounted against your system and you do not know about it. Clues to this intrusion might be in the Audit log but, unless they are analyzed, you might not know until it is too late. There are several commercially available tools that can aid in this task. These tools perform functions like gathering all the logs from your servers at a predetermined time and merging, sorting, and categorizing the events so you can check them in a more efficient manner. If you are unsure what types of events to audit, consider using this list as a baseline for your site. This listing has been adapted from Fossen's text *Securing Internet Information Server 5.0*.

  - Failed Logons
  - Failed File and Object Access
  - Failed Use of User Rights
  - Failed Change of Security Policies
  - Failed User/Group Policy Changes

These items are considered the essentials, but there are a few other items that should be audited. These include:

- All access to the Scripts and Bin Folders
- Consider auditing all access to the directories that contain files that are published as part of your web site.

## Data Protection.

Protection of the data being stored on a server exposed to the Internet also is a high priority. In addition to NTFS settings, regular tape backups are necessary for keeping a server available when it is exposed to possible attack. In the event a server is breached, it is possible to use a current backup to get the system restored and back online in a short amount of time. As part of your server documentation, a formal backup policy should be developed. This policy should contain all pertinent information concerning the backup and restoration policies for the server. These policies should include information that identifies the following:

- Who is responsible for making backups of the data and server configuration?
- How often are backups performed?
- What is the default location for backup media that has been stored?
- Who is authorized to restore data to the system?
- Are there copies of the data stored off-site?
- Who is responsible for maintaining off-site copies of the data?

Once this has been done and a backup procedure is designed, the method, or location of backup devices should be determined. In most instances a locally configured backup device is preferred over a network backup location. For security reasons a local backup device is better since no network connection has to be made to initiate the backup. The first thing to do after completing the installation of the system is to perform a complete backup of the server. This will serve as your recovery image in the event of a server crash or intrusion incident. You also will have to decide the frequency and type of backups to be performed. There are several options available to you, should you do daily backups? Should the daily backup be a full backup, incremental backup or differential backup? These decisions must be made prior to putting the server on line. The backup policy must be determined and in place before exposing the system to risk. If a server is attacked before a good backup plan is in place, it is too late.

# References.

Bragg, Roberta *Windows 2000 Security* Indianapolis, IN. New Riders, 2001.

Brenton, Chris *Mastering Network Security* San Francisco, CA. Network Press/Sybex, 1999.

Cox, Philip and Tom Sheldon *Windows 2000 Security Handbook* St. Louis, MO. Osboume/McGraw-Hill, 2001.

Curry, Benjamin, et al. *The Art and Science of Web Server Tuning with Internet Information Services 5.0* URL: http://www.microsoft.com/technet/iis/iis5tune.asp

Fossen, Jason. *Securing Internet Information Server 5.0* (Jan. 28- Feb. 1, 2001). The SANS Institute GIAC Training, 2001.

Fossen, Jason. *Securing Windows NT Step-by-Step* (Jan. 28- Feb. 1, 2001). The SANS Institute GIAC Training, 2001.

Fraser, B. *Site Security Handbook* (RFC 2196) 1997. URL: http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2196.html

Howard, Michael *Secure Internet Information Services 5 Checklist* URL: http://www.microsoft.com/technet/security/iis5chk.asp

McClure, Stewart, Joel Scambray and Goerge Kurtz *Hacking Exposed*. St. Louis, MO. Osboume/McGraw-Hill, 1999.

Norberg, Stefan *Securing Windows NT/2000 Server for the Internet* Cambridge, MA. O'Reilly, 2001.

Papatla, Ram *Setting Up a Reliable Web Server Using Windows 2000* URL: http://www.microsoft.com/technet/iis/iis5feat.asp

*Performance Monitoring and Reliability* URL: http://www.microsoft.com/technet/iis/prfrelmn.asp

Russell, Ryan and Stace Cunningham *Hack Proofing Your Network* Rockland, MA. Syngress, 2000.

Schmidt, Jeff *Microsoft Windows 2000 Security Handbook* Indianapolis, IN. Que, 2000.

Tulloch, Mitch and Patrick Santry *Administering IIS 5.0* Washington D.C. McGraw-Hill, 2000.