Global Information Assurance Certification Paper

Securing an IIS 4.0 web server, machine and all
Marshall S. Heilman
September 20, 2002
GSEC 1.4 Option 2

Introduction

The objective of this paper is to show how I secured my organization's web server, which fatally crashed earlier this year. I will describe the steps taken in securing the server from OS (Operating System) to IIS (Internet Information Server) and the vulnerabilities corrected by the configuration. I will outline the state of security the web server was in before the crash and the final state of security the server was in after all the changes were made.

The web server acts as an information server for those who wish to know about my organization. It provides them with technical information, location, maps, many links and many other things they might want to know about my organization. It also serves those of us in the organization when we are away from our home site. The web server provides the necessary links to our OWA (Outlook Web Access) accounts for those employees that need to travel throughout the world. In addition, the web site also provide our users access to our helpdesk trouble ticket system so they can easily enter a trouble ticket without having to burden the already over-burdened helpdesk technicians with phone calls.

Section I – The Crisis (Before)

Having never dealt with a web server before, I had my attention turned elsewhere when it was inevitably brought to the web server. One of my associates asked me to look at the security aspects of the server to determine its state of security, especially since security on the rest of our servers seemed lax at best. The web server existed on the service net, also known as a DMZ (De-Militarized Zone). After reviewing some very basic security measures, I determined that the web server was not very secure.

From an OS point of view there were no permissions set on files other than the default WinNT load. If an attacker were able to execute commands on the web server, he would easily have been able to run dangerous commands such as cmd, telnet, ftp, regedt32, etc. Basically, all of the system32 files were not protected. Looking in the User Manager I noticed that the guest account had not been renamed (though it was disabled) and a password had more than likely not been set (this is part of Defense in Depth, even

though the account is disabled, no account should have a blank password). An attacker should be hindered as much as possible, if the attacker is able to easily guess usernames (such as the existence of a default user named Guest), we are making his job that much easier. Instead the Guest account should be renamed to Administrator and the Administrator account renamed to something not easily guessable (this will perhaps slow down the attacker some if he concentrates on attacking the Administrator account thinking he will have system access). In Network Neighborhood I noticed that TCP (Transmission Control Protocol) filtering had not been enabled (See figure 1).

> Right click *Network Neighborhood -> Properties -> Protocols -> TCP/IP -> Properties -> Advanced ->* Check *Enable Security* then click *Configure*

Figure 1

For my web server out on the DMZ only two TCP protocols needed to be allowed: 443 (Secure Socket Layer) and 2998 (ISS RealSecure – an IDS, or Intrusion Detection System). Instead, TCP was allowing all. I checked IE (Internet Explorer) and noticed that it was version 5.5, which is less secure than IE6.0, even without the latest cumulative hotfix installed. Finally, I noticed that Norton Antivirus v5.0 was loaded, even though the rest of my organization was using v7.61. I never checked the date of the latest anti-virus signature.

The next area I checked was in IIS (version 4.0). I checked to see if any of the insecure script application mappings, .htr (Helper) and server side includes, had been disabled (See figure 2).

> Start *Internet Service Manager -> Default Web Site -> Properties -> Home Directory -> Configuration -> AppMappings*

Figure 2

None of them had been disabled. These scripting applications were designed with functionality and not security in mind. Even Microsoft recommends disabling them (such as in security vulnerability MS01-004 or in any Microsoft published web server security guides).

In fact, the only security that seemed to be in place was that we were running SSL (Secure Socket Layer), which runs on port 443, though we were also allowing port 80 HTTP (Hypertext Transfer Protocol) connections to the web server. NAV at least being installed should also get a security checkmark, though a small one from the outdated version.

One factor that I (admittedly) forgot to check (and I have definitely learned this lesson) was the status of the web server's backup tapes (our site runs a Scalar backup unit, but our security policy will not allow it to connect to the service net). Therefore backups had to be manually completed, but were unfortunately given a backseat to everything else that was going on. Due to the high volume of work my organization receives, basic system administration practices had tended to get left by the wayside. A backup is something that is so simple to do, as well as important, and yet had gotten

neglected.

Backups can be used for any number of reasons, though the main reason they are used at my organization is to ensure data availability. If a server were to crash or through some other catastrophe we were to lose all our data, my organization should not come to a standstill. Instead, we should be able to rebuild a server and restore the information from backup with minimal, if any, data loss. In addition, backups should be kept in an off site location, such as a secure warehouse or storage center, which prevents complete data loss during the event of a fire or other physical disaster. Another common use for backups is to move large amounts of data from one machine to another, an example of this could be adding the entire webroot directory from a web server on a DMZ to a test server on another network, backups are the fastest way to accomplish this.

The next day the web server crashed due to an incorrect installation of an IOMEGA zip device by an associate of mine. In all fairness, the web server should not have crashed as hard as it did due to an incorrect driver installation, but since that was the case, I must go forward from there. Checking the status of the backup tapes now became a priority, only for me to discover that the previous backup was almost six months old. I got the web machine back on its feet by re-installing the OS on the "C" partition and preserving the web server information (which was located on a separate partition).

Obviously getting the web server back up was my top priority as I had my management breathing down my neck to get it back up. However, as any up and coming security administrator will tell you (notice how I hesitantly deem fit to call myself that after the backups mishap), you never want to take a web server and throw it back online as soon as possible, you want to fix the security vulnerabilities first. Since my site had not been hacked, but had suffered from an internal "stupidity attack," I felt somewhat assured that if my web server went back up prematurely, it would not get instantly taken down (that does not mean that I planned to leave it online for long until full security was in place).

Section II – Securing the Server (During)

The Operating System

The very first thing I did was to take the LAN (Local Area Network) cable out of the server to prevent any unnecessarily easy hacks on the server while it was coming back up. After the initial OS [WinNT Server] install, I installed the latest Service Pack, SP6a (Service Pack 6a) w/128k bit encryption. One of my projects that I had almost finished was writing an information (.inf) file for WinNT server. Basing my registry key and file permissions off of Microsoft's Securing WinNT Server 4.0 Baseline Checklist, I created the baseline permissions for a WinNT server. Using the above document, as well as some of my companies' security policies, I configured the account policies, local policies, and event log settings in the information file. Being that this server was out on the DMZ I turned on logging for everything and set the logs to overwrite events older than one month. I spent about the next hour finishing the information file and tested it briefly on a

test server (note: while I was doing this I performed a backup of the web server data to be loaded on the test machine).

To test the validity of the information file I ran to my trusty test network. My test network is very small but incorporates almost everything I need to adequately test new patches. Included in this network are four machines; a Red Hat Linux 7.2 machine, a W2K client machine and of course, and two WinNT Server machines. One of the WinNT servers acts as the PDC (Primary Domain Controller) and runs DNS (Domain Naming Service) for the test domain. I formatted and re-installed the other WinNT server with WinNT Server 4.0, SP6a w/128 bit encryption, IE6.0 with the latest hotfix: MS02-047, IIS 4.0 (the same install as would be used on the actual web server which is detailed later), FrontPage 2002, MMC (Microsoft Management Console) and SCM (Security Configuration Manager). This is an identical configuration to the actual web server. I then loaded the web server data that I had backed up earlier, configured IIS 4.0, and pointed it to the restored information. I then loaded the information file into the SCM and the test server was configured appropriately.

To determine if there was an improper file permission setting defined in the information file, I turned off all auditing except for File and Object Access Failures. The information file that was loaded enabled auditing for all objects so I manually disabled the logging (See figure 3).

Run *User Manager, select Policies, Audit*

Figure 3

This would allow me to research which file permission was not allowing the user to correctly browse through the web pages (my major concern was that I might have made the permissions too tight and users would not be able to browse the web site, thus making all the security worthless). It is a good security practice to "tighten the security screws" until you break functionality, and then loosen the screws some until the functionality is restored. This way you are sure that your permissions are secure (note: ensure that you are testing on a test server, this is not something you want to do to a production server, unless you do not value your current job). After loading the initial web page as a privileged user I logged off and logged on as my test user, which had basic user rights. I was able to effectively browse around the web pages with no difficulties. Just for safety I logged back in as a privileged user and ran through the object access failures, there were a few entries but nothing that affected the web server functionality. I was now confident that the information file was ready to be implemented on our production server.

Back on the actual web server I installed the EnPassFlt.dll file by copying it into the WINNT\SYSTEM32 folder. This file requires passwords be 12 characters in length, at least one upper and lower case letter, one number, and a special character, such as: !@#$%^&*.

In order to be able to install the information file in a WinNT environment, a few other programs need to be installed first. As a general rule, I install IE6 next because many programs require IE in order for the installation to work. After IE6 is installed, the correct cumulative hotfix must be applied. In this case, MS02-047 was installed. MS02-047 was the latest cumulative patch for IE6 at the time of this paper. Since this is a

cumulative hotfix I will not list all vulnerabilities it patches, but I will give the reference page here. As a general rule, you always want to install the hotfix for a program right after its installation so you do not lose track of what you have installed. After the installation of IE6 I needed to install the SCM program. To get the SCM, click here. Before the SCM can be installed, MMC must be installed. MMC is automatically installed from the SCM .exe (Executable) file downloaded from the link displayed above. After all three of these programs had been installed, I was ready to install the information file. In this case, I had already created the information file. Information files can be used to accomplish any number of things; in this case I am using one to accomplish the task of configuring a machines baseline security. Figure 4 is a screen capture of what an information file looks like while being created.

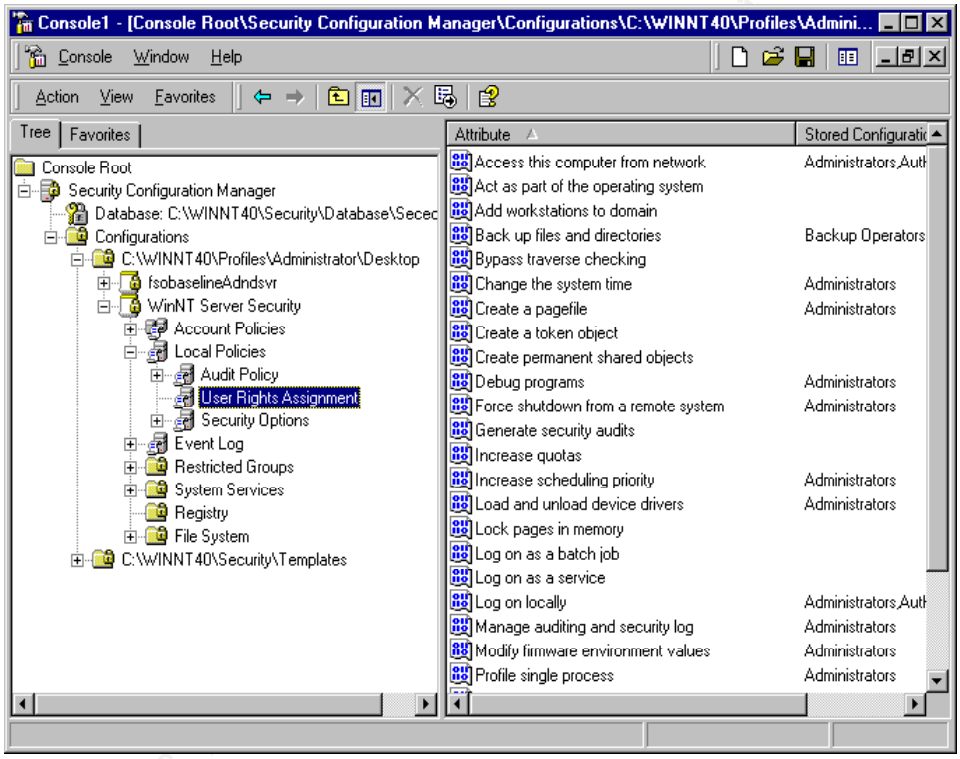

Figure 4

To install an information file on a WinNT server (after SCM and MMC are installed) you must first add the SCM snap-in to your main view in MMC. See figure 5 for instructions to load an information file into a machine.

> Bring up the run line, type *MMC*, Click *Console -> Add/Remove Snap-in -> Add -> Security Configuration Manager -> Add -> Close -> OK*. Next open *Security Configuration Manager -> Database ->* Right click on *Database ->* select *Import Configuration* and then select the appropriate .inf file. Next, right click on *Database ->* and select *Configure computer.*

Figure 5

Unlike the W2K Security Configuration add-in for MMC, SCM does not include the ability to add/remove registry keys via an information file. Instead, all you are allowed to do is set permissions on certain registry keys (note: you can add registry keys to the baseline). To overcome this I created a small registry (.reg) file that changes the key value to the security conscience value; information as to why one would want to make those changes can be found in the document that I used to know which keys to set - Microsoft Internet Information Server 4.0 Security Checklist. Figure 6 is a screen shot of that file. After the registry file was installed, I ran a simple batch (.bat) file, which removes all OS/2 and POSIX files (see figure 7).

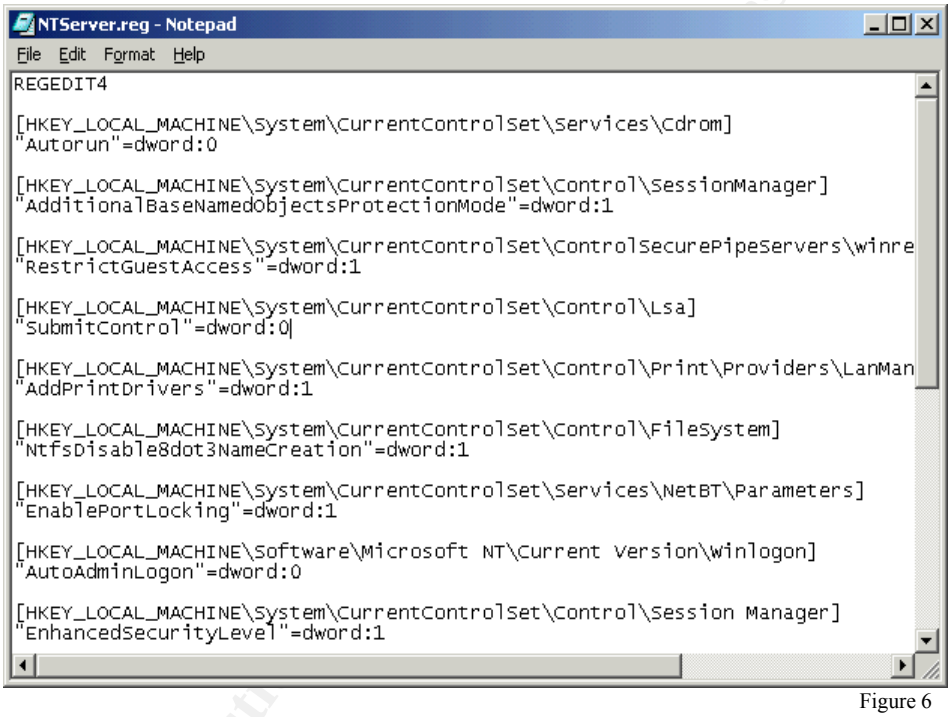

Figure 6

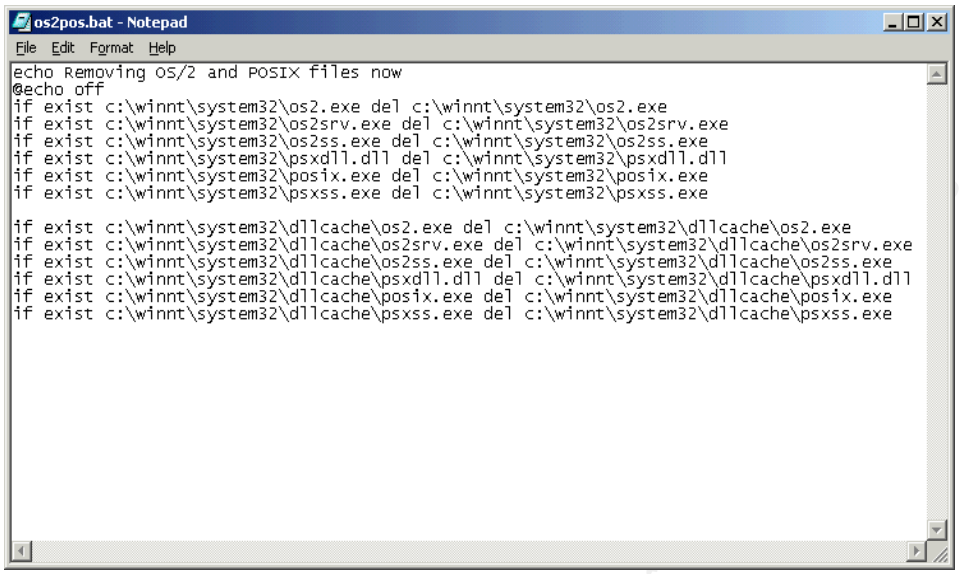

Figure 7

The reason the batch file deletes files out of both the winnt\system32 and the winnt\system32\dllcache is because in W2K (Windows 2000) the dllcache directory is used to replenish files that have been deleted through a process called WFP, or Windows File Protection – for a brief overview of this and how to disable it, and why you might want to, click here. Obviously this is not needed since I ran it on a WinNT system, but the original batch file was created for W2K machines and the extra lines would not actually hurt anything because the dllcache directory does not exist.

The main security reason for disabling, or in this case deleting, the POSIX and OS/2 files (and registry keys, see figure 8), is to prevent Denial of Service attacks. Another reason to disable these OS subsystems can best be said by Keith McClellan, "POSIX is a standard for operating system interoperability that is required of all operating systems purchased by the government, and OS2 support is, well, support for programs originally written for the OS2 platform. Since almost no one has a use for these subsystems, we might as well disable them and save ourselves the wasted memory, right?"[1] Obviously, if one has a OS/2 machine on their network, than these files may be necessary, but in my organizations case, a mix of W2K clients and WinNT servers, they were unnecessary. The more services, or in this case subsystems, that are offered by a server, the more potential holes an attacker can find. Why give them unessential services with which to play?

> Run *Regedt32 -> HKEY_LOCAL_MACHINE -> SYSTEM -> CurrentControlSet -> Control -> Session Manager -> SubSystems –>* remove *OS2* and *POSIX* keys.

Figure 8

If you are wondering whether I changed the administrator account name and

[1] McClellan, Keith (2002) "Windows 2000 Memory Subsystem Tweaking" http://www.arstechnica.com/tweak/win2k/others/memory-2.html

renamed the guest account, the information file did it for me. As for the newly renamed administrator account and the renamed guest account passwords, I set them manually. All other account configurations (such as lockout time, who may log on, who can act as a service, etc.) are defined in the information file (pretty useful file!)

Once all the above had been accomplished, I went to the Microsoft's Security Patch Center and got the updated list of all relevant OS security patches that had been released since May of 2001's SRP (Security Rollup Package) which encompassed all of the previous OS hotfixes up until that point. Armed with new OS security patches I installed the following hotfixes:

SRP – This is rollup of all current security hotfixes for the WinNT OS after SP6a
MS02-013 – VM (Virtual Machine) Cumulative Hotfix
MS02-032 – Media Player Cumulative Hotfix

These hotfixes were applied because they contain fixes for vulnerabilities that affected my server. The SRP contains fixes for many different vulnerabilities, all listed here. The VM hotfix fixes two vulnerabilities: (1) "Java requests for Proxy resources"[2] are mishandled and potentially allow a denial of service attack or a session transfer into the control of an attacker and (2) VM's security check on casting operations (when types are converted by casting operations[2]) is flawed allowing an attacker to execute code outside of the program (at user level permissions).

Other hotfixes were either determined to be unnecessary (due to my network configuration or because of unsupported software). An example of such a hotfix is MS02-006, which is designed for SNMP. As I will state later in this paper, SNMP is not a service that is needed on the web server to allow it to properly perform its job. Therefore, I turned it off (See figure 9).

> *Start menu -> Settings -> Control Panel -> Services ->* Highlight *SNMP ->* Click *Startup ->* Set *StartupType* to *Disabled*. Repeat procedure for *SNMP Trap Service*.
>
> Figure 9

The next OS security procedure that I put in place dealt with the harmful commands located in the Winnt\system32 directory. I moved executable files such as telnet, ftp, cmd, regedt32, netstat, etc… onto a different drive in a different folder (such as e:\cmdtools). This will at least slightly slow down a hacker. For a complete listing of all the files that were moved, see the Microsoft IIS 4.0 Security Checklist. After moving these files I set appropriate permissions on the directory and on each individual file, so only the sysadmin had any type of access.

The final security measure I put into place on the OS was the installation of NAV (Norton Anti-virus) 7.61 client. Even though we have a NAV server, which manages all our clients on the internal domain, the same is not true for the DMZ, so the client had to

[2] Microsoft Corporation (2002) "Microsoft Security Bulletin MS02-013) http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS02-013.asp

be installed in unmanaged mode. After installing NAV (accepting all defaults unless you wish to change the drive where the installation will be located), configure the System Realtime Protection to scan all drives, clean all viruses as the primary action taken and to log only viruses that could not be cleaned. I did not set the secondary action to delete the file because that can be a dangerous option if important files start getting deleted. Instead I set the secondary action to only log the files because I perform, at a minimum, weekly virus checks.

The next thing to do since I had the OS stable and "secure," was to create an Emergency Repair Disk (ERD). An updated ERD is very useful if your OS should fail to boot correctly, however, one must make sure that the ERDs are updated anytime a system configuration is changed, otherwise they are next to useless. The ERD contains copies of necessary system files in order to bring a crippled OS back to its feet. To create an ERD, I inserted a new diskette and ran the command: *rdisk /s* from the command line. This command will format your diskette as well as install the necessary files and update your winnt\repair directory. For an excellent article on how to keep your Emergency Repair Data current NOT using actual diskettes (as long as the machines are networked), read this article. Once that had completed, I proceeded to backup the server because I had learned my lesson!

Internet Information Server 4.0

With the OS security in place, it was time to move on to the real problem with web servers, IIS. In my case, since I was running WinNT server, I had no choice but to implement IIS 4.0 (vice IIS 5.0 which would have been a more secure option.) Even though my "D" had been saved, I still needed to re-install IIS in order to publish the web page.

When installing IIS 4.0, many unnecessary options are installed by default. To overcome this, I performed a custom install to get rid of the unnecessary and/or unsafe options. For IIS I only installed the Internet Service Manager and the WWW (World Wide Web) server. The NNTP (Network News Time Protocol), FTP (File Transfer Protocol) and SNMP options were more risky than they were worth to my organization.

One security stance that I believe all security professionals advocate is the policy of least permissions. That means to give people the bare minimum permissions they need to do their job. I am not saying to restrict users from doing their job because security so tight that people cannot properly do their jobs is no good to anyone. With that being said, look at the policy from another angle, instead of using permissions, substitute programs, or services. Install only the bare minimum of services needed for the server to accomplish its job. In this case, the web server's job included allowing access to necessary information about my organization as well as providing a link to OWA. NNTP, FTP, and SNMP provide absolutely no job essential service for the web server. Why should I risk a current or an as-of-yet-undiscovered vulnerability with any of these services when they are not needed for the web server to perform its job? The simple answer is that they are a risk that is not worth taking.

Next I ensured that all options, except for RDS (Remote Data Service) v1.1, and

MDAC (Microsoft Data Access Component) v1.5, were checked for. RDS was not installed because it was not a necessary component for my web server to perform its job, therefore it is an unnecessary security risk. MDAC is user for data access programs, of which my server had none. I then ensured that Index server with all of its options and Transaction server, without the server development piece, would install.

One thing I like to do, especially when installing a new box, is to run the security patches first, and then make most security changes. This will ensure that the changes you personally make do not get overwritten or changed by other patches. This brings me to my next point, in which I installed a hotfix and a lockdown tool for IIS. The hotfix, MS01-044 is a SRP for IIS. For a listing of all vulnerabilities it fixes, click here.

One stance that I take on security, mostly because my predecessor here ingrained this philosophy into me and I agree with it, is that if a patch isn't necessary, do not install it. The reason for this policy is that sometimes patches break more than they fix. Sure, they fix the immediate problem, but they can also have unforeseen affects; such as opening up new holes or by taking away functionality of a program that already exists. There are two main flaws with this stance: 1) If the current configuration changes the necessary patches may not be installed and 2) An attacker could potentially enable the program that the patch was meant for. For the first flaw, if the current configuration changes, I will know about it and I have documented the patches that have been installed on the machine. Therefore I will check the current list of patches and see that there is a patch that has not been installed because it was not necessary, and I will install it. For the second flaw, I am not very concerned about it because if the attacker has permissions to change such features, they have already taken control of the machine and the application of that certain patch would not have made a difference. To illustrate my point, hotfix MS02-028 did not need to be installed because my web server is protected against this vulnerability by other means. What that means is that since MS02-028 deals with a chunk overflow in malformed .htr requests, and we do not allow .htr scripting (something that I would disable later on), the patch was an unnecessary install. Patches should always be fully tested, thought-out and necessary before implemented on a production server.

The next tool I installed on top of IIS was the IIS Lockdown Tool. This tool gives the ability to lockdown unnecessary features and permissions of IIS. Figures 10-12 are few screen shots from the IIS Lockdown Tool installation.

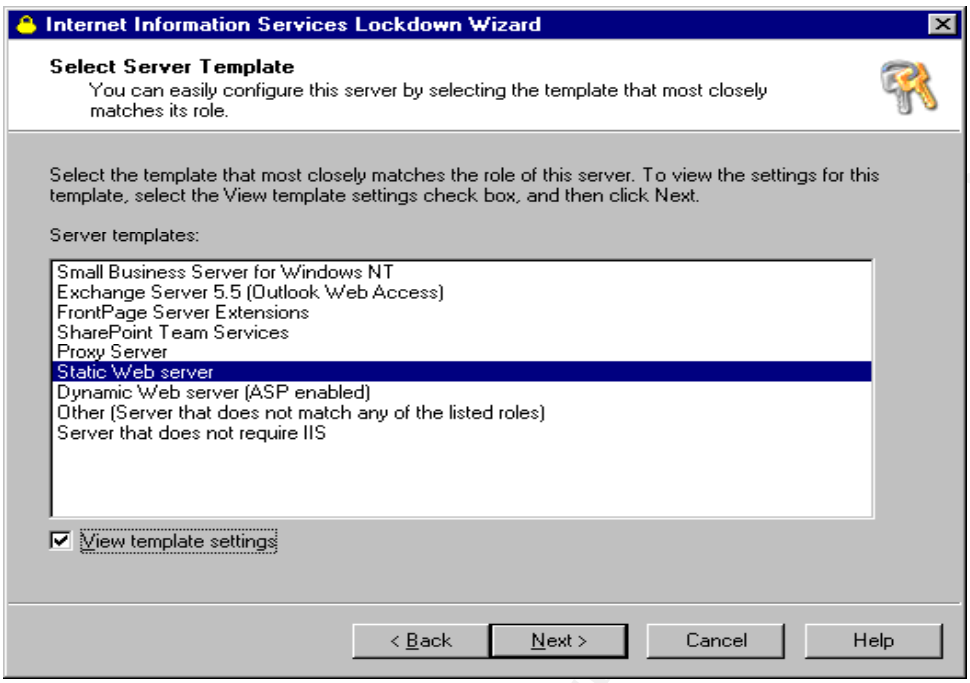

Figure 10

Since the web server has a static IP address, I chose the corresponding template. Always ensure that *View template settings* is checked, this is enable you to see what changes the tool is planning for your system. You never want to be caught unaware of changes being made to a system you are responsible for.

Figure 11 shows that I chose not to disable Active Server Pages or Index Server Web interfaces. The reason is that our webmaster uses Index Server (in order to support searches on our web site) and Active Server Pages (.asp). I also removed .HTR scripting, Internet Data Connector, and Server side include script mapping because there is no business need for these on the web server. Once again, there is no need to run unneeded software and open new security risks that are not absolutely necessary.

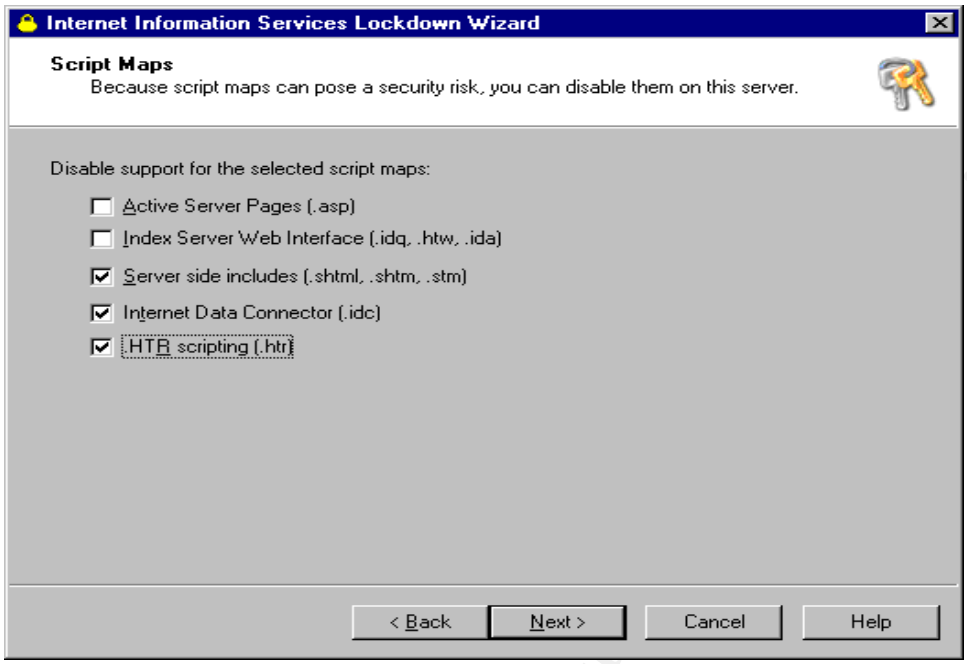

Figure 11

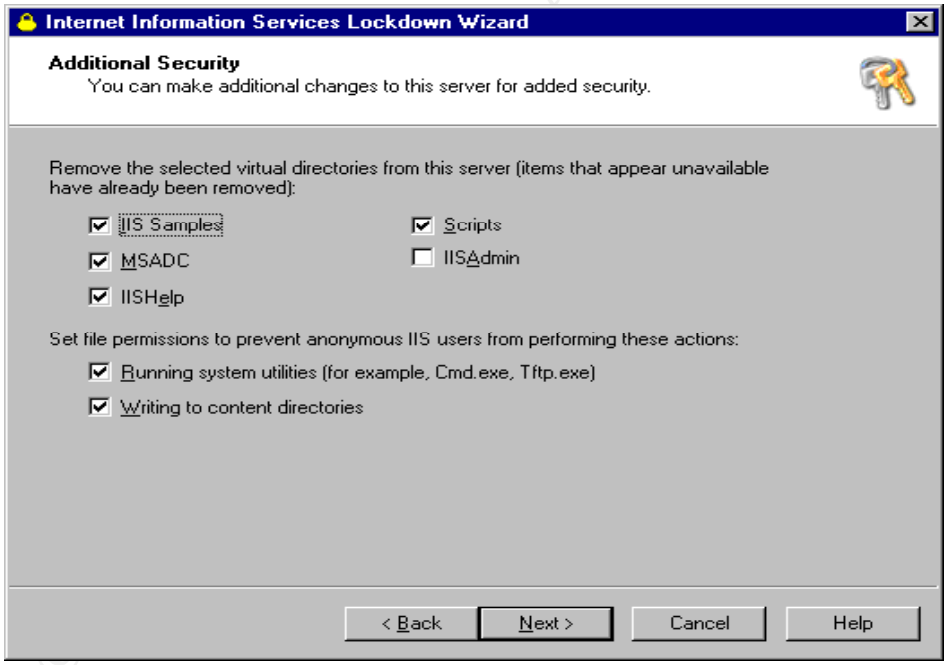

Figure 12

Figure 12 shows that I selected all options except for the IISAdmin. The only virtual directory needed for the web server to function correctly (that have the option of being removed), is the IISAdmin directory. This will allow us the run the IIS Admin program and administer IIS without having to go through the web manager. As for setting the file permissions, anonymous users should not be allowed to perform any action on the web server, except for what is expressly needed for them to get the necessary

information, which most certainly does not entail them saving information or running system utilities.

The last option screen (not shown) defaults to install the URLScan filter. I left this option checked because the URLScan filter protects against many known attacks on web servers. By default, the Lockdown Tool installs URLScan 1.0, but I would update this later (to v2.5). The final screen will show a summary of changes to be made to the web server. Run through these changes one final time to ensure that the changes are correct, then finish the program. When the installation is done, a reboot is not necessary.

After installing the IIS Lockdown Tool, I upgraded to URLScan 2.5, SRP version. All that need to be done was to run the executable file downloaded above. I chose the SRP version over the baseline version because it protects against all vulnerabilities listed in MS02-018. For a description of some of the features of URLScan 2.5 (baseline and SRP), click here.

Following Microsoft's IIS 4.0 Server Security Checklist I performed such security acts as locking down the enabled ports, only allowing ports 443, 80 and 2998, though port 80 was only temporary until the server was fully enabled again, enabling remote administrator account lockout, ran SYSKEY, and enabled logging. The exact procedures I used are outlined in the above web site. As mentioned earlier in the OS portion of the document, port 443 allows SSL traffic, port 80 allows HTTP traffic, and port 2998 is necessary for ISS RealSecure's Server Sensor, an IDS program.

I then made a few registry changes to prevent against "The most common attack against an IIS server,"[4] which is an RDS attack. In this case, even though I did not install RDS, the fix for this vulnerability does not include loading any new software, so the Defense in Depth method is appropriate here. The following registry keys were deleted.

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch

The three keys (mentioned in the quoted paper) were:
RDSServer.DataFactory
AdvancedDataFactory
VbBusObj.VbBusObjCls
(Note: Our web server only had the first two keys present)

With IIS installed and the home directory set, I tested to ensure that my web page was accessible via port 80 because my certificate to enable SSL no longer existed, so I was forced to test with straight HTTP. I was able to hit our homepage no problem, so the web server was at least [functionally] back in action. The next step was to get the web site back on SSL.

In order to enable SSL on a web server, a PKI (Public Key Infrastructure) key must be installed. For my organization, the CA (Certificate Authority) is at another location. Our LRA (Local Registration Authority) created a new 1024 bit asymmetric key

---

[4] Cooper, Russ (2001) "10 Steps To Better IIS Security," September, Information Security Magazine. www.infosecuritymag.com/articles/september01/features_IIS_security.shtml pg1

pair using the IIS Key Manager. After plugging the LAN line back in, he sent our administrative data as well as the public key via a TLS (Transport Layer Security) connection to the CA. RFC2818 contains information governing TLS. We had to install Netscape 4.76 in order to send and receive the certificate because it was the only browser that supported the necessary options with the CA's website. After a few hours we were able to download the new cert (created using our public key) from the CA's certificate issuing web site. I would uninstall Netscape shortly after the successful posting of the certificate. The certificate was then installed into the IIS Key Manager. The root CA certificates were also installed into the web server using Internet Explorers Certificate Manager.

You might be wondering why all the fuss over SSL? Why not just use HTTP and make life so much simpler? The main reason for using SSL is that it ensures that the server and the client are actually talking to one another and not to a "man in the middle," which is possible unless a secure communications channel has been created between server and client. Through an unencrypted transfer of information it is possible for a "man in the middle" to pose as the target server or client and intercept (and/or respond) to all communications; this could also result in a denial of service attack if the attacker, the man in the middle, chooses not to respond and only to intercept. In brief, SSL works by the client sending random information to the server, the server responding with its certificate (containing information about the server and the issuing CA's name). The client then asks the server to prove his identity, which the server does by sending back a response that has been hashed and encloses his private key. The client then authenticates that and sends a premaster secret along with the server's public key back to the server for authentication, who then authenticates it and responds with a digested message including the MAC (Message Authentication Code) and the secret-key. For a complete article on SSL, click here.

I then went back into IISADMIN and made the following changes shown in figures 13 and 14.

Open IISADMIN, *Default Web Site -> Properties -> Directory Security -> Anonymous access and authentication control -> Edit*: check *Allow Anonymous Access.*

Figure 13

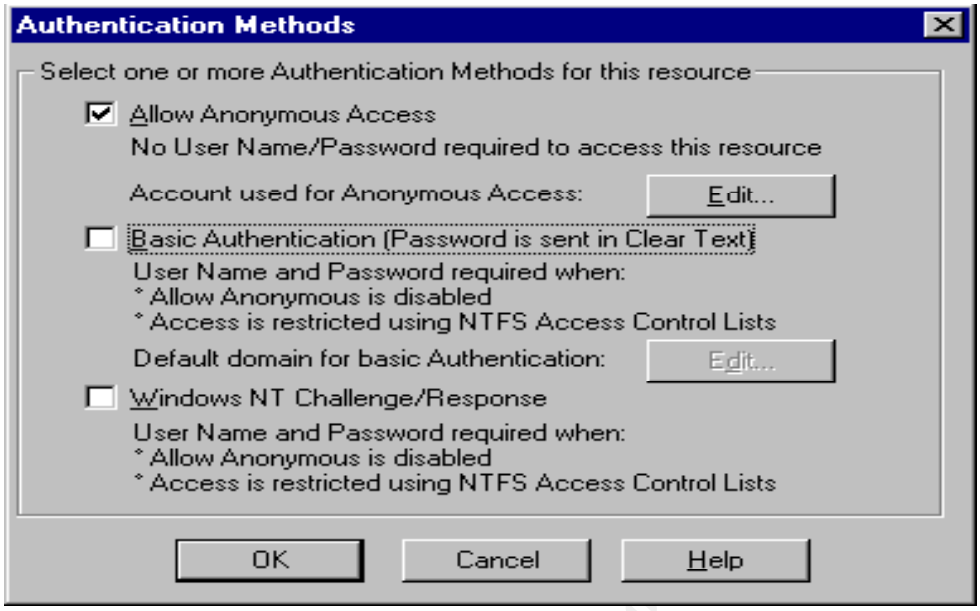

Figure 14

Allow anonymous access was necessary because I did not want users to have to log into the web server, this would defeat the purpose of having people that do not belong to my organization go to our web site to glean information.

Under *Default Web Site -> Properties -> Advanced*, I added my server's identity and SSL port to re-enable SSL (see figure 15). I removed the pictures of my sites IP address for security reasons.

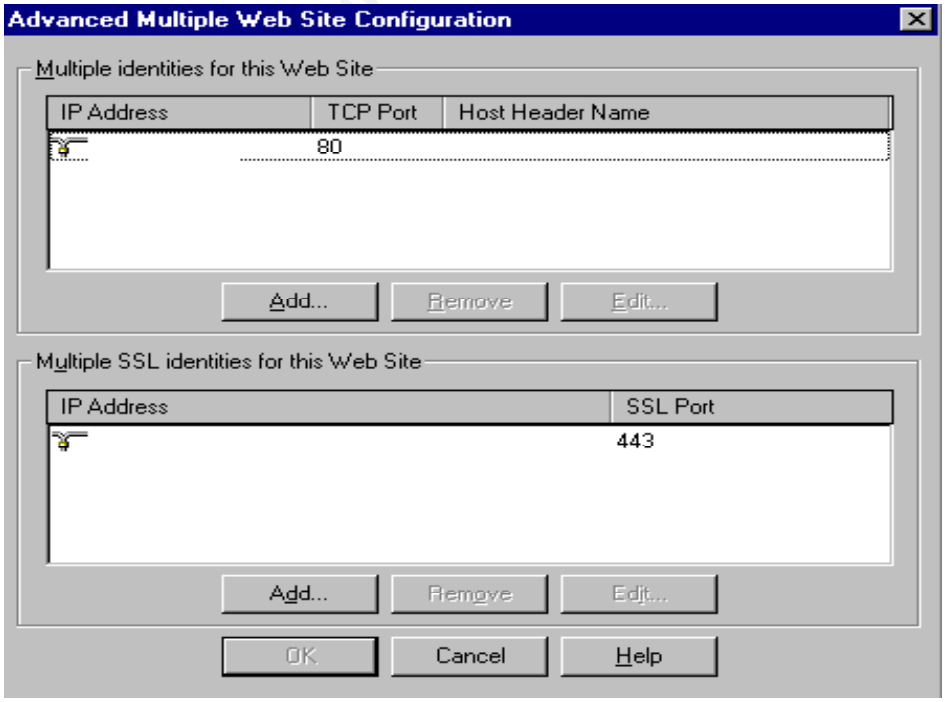

Figure 15

After re-enabling SSL, there was one more change I needed to make (see figures 16 and 17).

Right click *Network Neighborhood -> Protocols -> TCP/IP -> Properties -> Advanced,* Check *Enable Security*. Next click on *Configure* and remove TCP port 80 and add TCP ports 443 and 2998.

Figure 16

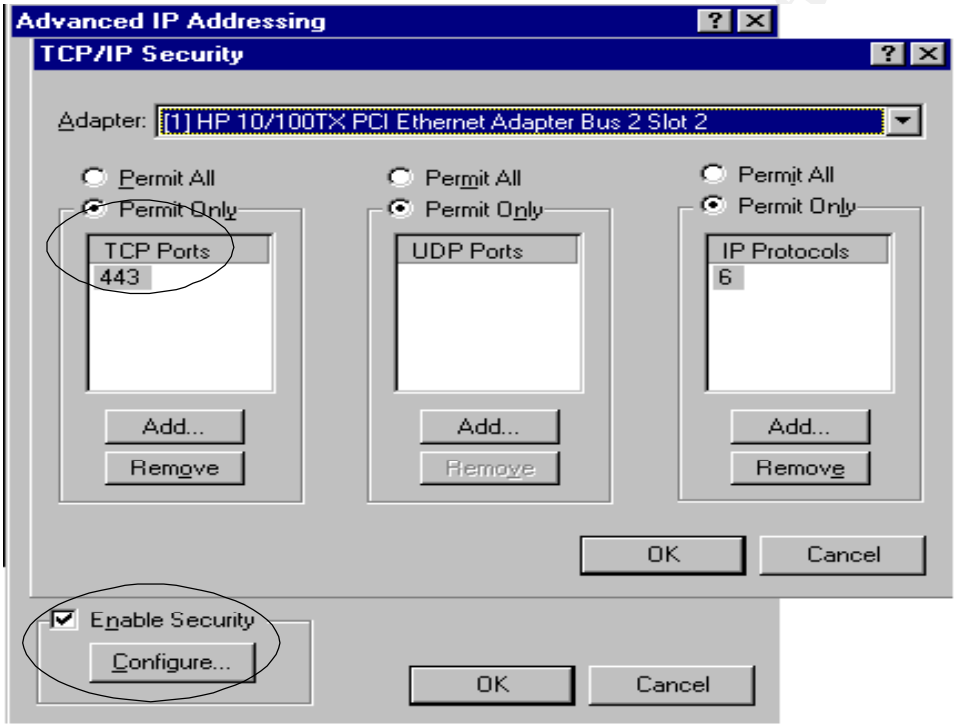

Figure 17

Two final measures of security were put into place. The first was to monitor which ports the server was listening on. To accomplish this I could have used the netstat command (which I have found to be very slow if numerous connection have been made), or even fport, which is an excellent tool, made by Foundstone Inc. Instead, I chose to go with the graphical version [of Fport] called Vision, also available from Foundstone Inc. Figure 18 provides a screen shot of the Vision program.

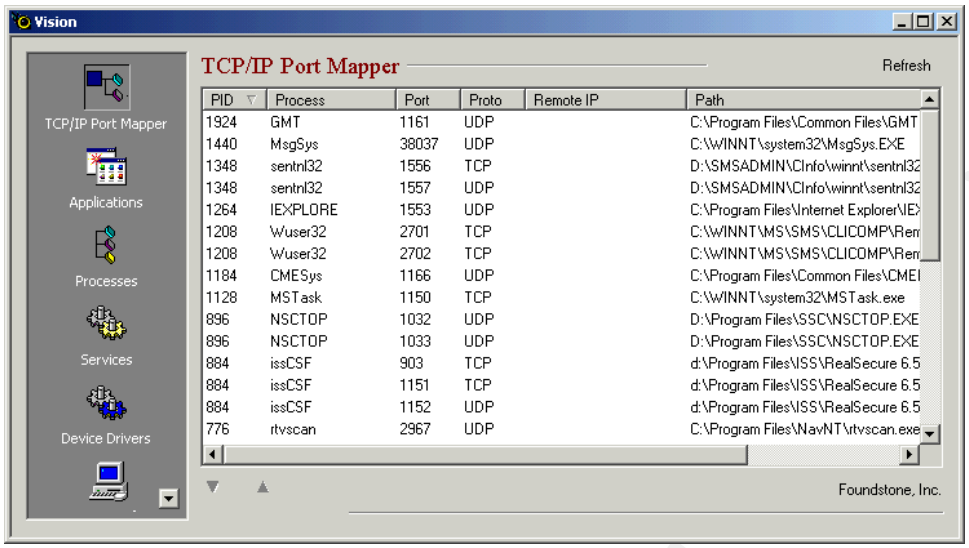

Figure 18

The final measure of security that I put into place (and every machine in a DMZ should have this), was to install the ISS RealSecure Server Sensor on the server. This allows the web server to be monitored from our ISS RealSecure 6.5 workstation, which is also located on the DMZ. We did not have to purchase any software or licenses to accomplish this as we already using the software, but had not configured it for use yet on the web server. This extra measure of security allows for some known attacks to be blocked as well as giving early detection of an attack or what is perceived, by the IDS using signatures, to be an attack. This can also be used to strengthen ones security posture by sifting through the "false alarms," and tightening the screws where an actual security event gets reported, be it by an attacker or just an accidental event that shouldn't have been allowed.

At this point the server was about as secure as it was going to get before being "officially" put back online. I performed another backup (appending it after the initial backup just in case), and re-ran the repair disk utility on the original diskette (the original ERD was overwritten with this new one that contained any updated information due to the IIS installation and changes). The emergency data is also contained in the WINNT\repair directory.

One unexpected information leak was discovered on the web server a few days after I was "finished" securing the box. This leak had existed in the previous setup of the web server, it just had not been noticed. The link to the OWA page, which was located on the internal network, displayed the actual IP address of the exchange machine. This just goes to show that even with defense in depth it is possible for attackers to find holes or gather information carelessly thrown out there for them, many times through human error, such as this case.

To fix the problem, I set up a CNAME (Canonical Name) in DNS for the exchange server and had the web link use that instead. In this case, I set it up to be another WWW address. This also served an alternate purpose of giving our users another web address they could use to access their OWA accounts should the web server ever go down again (OWA access is the main actual business aspect the web server was used for).

Alternate solutions to this could have been to (1) use the hostname of the exchange server – this solution is not any better than an IP address since the hostname will lead you straight to the machine as well, or (2) to have set up a trust relationship between the web server and the exchange server (which resides on the internal domain), share out the drive containing the startup page, and point the link to the specific file – this is the worst solution of all and could have severe security consequences. A security consequence would be an attacker that takes over the web server then has instantaneous access to the exchange server through a trust relationship, defeating the purpose of having the server on the DMZ. With the current configuration, transferring the page to another WWW page, the web server is transferring the link to the control of the exchange server. If an attacker were to take over the web server, he could not access the exchange server any more than a regular user could, through the use of OWA.

Section III – Looking Towards the Future (After)

Now that the web server is at a point where it can be considered secure, it is time to look ahead to the future of the server. Saying that the web server is secure is not a claim that vulnerabilities cannot still be found, but that to the best of my knowledge and resources, the server is protected from known vulnerabilities. As new vulnerabilities are discovered, I will continue the never-ending quest of securing the web server against attackers. The immediate question one would ask is, "Was the problem solved?" There are three answers to this question:
(1) The immediate problem (the web server was offline due to a crash) was fixed.
(2) The problem of the web server having almost nothing in the way of security has been fixed.
(3) The problem of securing a server against all possible vulnerabilities was not solved, and the solution has not yet been discovered.

In looking toward the future of my organization's web server, I have made two major proposals. My first proposal is to upgrade from WinNT server to W2K server and to IIS 5.0. In the words of my SANS instructor (who was quoting another SANS instructor), "I want to write a one page document on securing WinNT. Upgrade to W2K!" My second proposal was to install a host based firewall.

Even though I am recommending upgrading our web server to W2K server, I did not recommend using Microsoft's ISA (Internet Security and Acceleration Server 2000) firewall for a number of reasons. When purchasing new security software, cost must be weighed against the return. The web server is important to my organization, but it does not contain our "corporate jewels," or our most valued information. Though important, my organization will not lose out in millions of dollars or trade secrets if the web server gets compromised. The cost of a single license (non-Enterprise edition) is $1509.00 and the full-fledged Enterprise edition costs $5999.00. While the enterprise edition would be overkill for my organization, $1509.00 is still a lot to spend to have the web server protected by a second firewall. In contrast with those figures, BlackICE Server Protection costs $299.95 and an additional $149.95 will get one year of service and support; a grand

total of $449.90. ZoneAlarm costs $49.95 and comes with a year worth of updates and online support. Comparing necessity and, of course, the price, I believe that the best solution for my organization is ZoneAlarm Pro 3.1, though I added BlackICE to give my managers another option.

Hopefully in the near future I will be able to continue using the defense in depth, or multi-layered, strategy on the web server and at the bare minimum get one of my two requests granted. Even baby steps are steps toward progress in the security community.

Section IV - Web pages

Microsoft's WinNT Server 4.0 Baseline Security Checklist
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/chklist/nt4svrcl.asp

Microsoft Internet Information Server 4.0 Security Checklist
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/chklist/iischk.asp

WinNT SRP
http://www.microsoft.com/NTServer/sp6asrp.asp

WinNT OS SRP Information
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/news/nt4srp.asp

Security Configuration Manager download
ftp://ftp.microsoft.com/bussys/winnt/winnt-public/tools/scm/

Microsoft's Security Tool page
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/default.asp

MS01-004 Malformed .htr Request Allows Reading of File Fragments
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-004.asp

MS01-044 IIS 4.0 SRP and Information
http://support.microsoft.com/default.aspx?scid=kb;en-us;Q297860

MS02-006 SNMP Hotfix
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS02-006.asp

MS02-013 Virtual Machine Cumulative Hotfix
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS

02-013.asp

MS02-018 Contains list of vulnerabilities fixed by URLScan 2.5
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS02-018.asp

MS02-028 Chunk overflow in malformed .htr request (IIS)
http://support.microsoft.com/default.aspx?scid=KB;EN-US;Q321599&

MS02-032 Media Player Cumulative Hotfix
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS02-032.asp

MS02-047 IE 6.0 Cumulative Hotfix
http://www.microsoft.com/windows/ie/downloads/critical/q323759ie/default.asp

ARS Technica: Article on disabling WFP
http://www.arstechnica.com/tweak/win2k/others/disable_sfp-1.html

IIS Lockdown Tool
http://www.microsoft.com/Downloads/Release.asp?ReleaseID=33961

URLScan 2.5, SRP
http://www.microsoft.com/Downloads/Release.asp?ReleaseID=37756

URLScan 2.5 Information Page
http://support.microsoft.com/default.aspx?scid=KB;EN-US;Q307608&

SSL Information Page
http://developer.netscape.com/tech/security/ssl/howitworks.html

TLS Information Page
http://www.ietf.org/html.charters/tls-charter.html

RFC2818 HTTP over TLS
http://www.ietf.org/rfc/rfc2818.txt

Fport
http://www.foundstone.com/knowledge/free_tools.html

Vision
http://www.foundstone.com/knowledge/free_tools.html

ISS

© SANS Institute 2000 - 2005 Author retains full rights.

http://www.iss.net/

Microsoft Internet Security and Acceleration Server 2000
Pricing
http://shop.microsoft.com/Referral/Productinfo.asp?siteID=10538

BlackICE
http://www.iss.net/netice/
Pricing
http://www.iss.net/products_services/hsoffice_protection/buy.php

Zone Alarm Labs
http://www.zonelabs.com/store/content/home.jsp
Pricing:
http://www.zonelabs.com/store/application?namespace=zls_main&origin=global.jsp&event=link.skuList&&zl_catalog_view_id=201

Section V – Acronyms

.asp – Active Server Pages
Used to combine HTML, scripting and web components to create web pages

.bat – Batch file
Used to run small DOS scripts

.exe – Executable file
Runs programs compiled into machine code

.htr – Helper application
This was originally designed and implemented to allow users to change their passwords from the Web.

.inf – Information file.
Used in the Security plugins of MMC to configure baseline security on WinNT or W2K machines

.reg – Registry file
Used to make changes to the registry. Runs like a script.

CA – Certificate Authority
People with the ability to cut top-level PKI certificates

CNAME – Canonical Name
Used in DNS to establish another name by which a machine can be identified.

DNS – Domain Naming System
Used to match IP addresses with hostnames

DOS – Disk Operating System
Microsoft's command line interface, also known as DOS (earlier versions)

ERD – Emergency Repair Disk
Used to help a crippled OS get back on its feet

FTP – File Transfer Protocol
This allows people to connect via FTP to your server and send or get files.

HTTP – Hyper Text Transfer Protocol
Protocol used to browse the WWW

IE6 – Internet Explorer 6
Microsoft's program used to browse the WWW

IDS – Intrusion Detection System
Intrusion detection Systems can be server, host, or network based. They are used to warn of a [detected] impending attack or suspicious activity.

IIS – Internet Information Server
WinNT Server uses version 4.0, this is the program that publishes web pages as well as FTP

ISA – Internet Security and Acceleration Server 2000
Microsoft's W2K full-featured firewall

LAN – Local Area Network
Term given for a bunch of machines connected together, usually within the same room, office space, or even building

LRA – Local Registration Authority
Used in large companies where only one CA is present, they are able to cut local certificates

MAC – Message Authentication Code
Used with the SSL protocol, helpful in preventing "man in the middle" attacks

MDAC – Microsoft Data Access Components
This technology is enables the ADO (Universal Data Access) for Microsoft, which allows different types of data formats to be interchangeable

MMC – Microsoft Management Console
Program used to control many aspects of a WinNT or W2K machine

NETSTAT – Command designed to show which TCP and UDP ports your machine is listening on. For a listing of all netstat options, enter the cmd: netstat ? from your command prompt

NNTP – Network News Transport Protocol
Protocol used to post, distribute, and retrieve USENET messages[5]

ODBC – Open DataBase Connection
Allows for various database program interconnectivity within a Microsoft environment.

OS – Operating System
The Operating System is the interface between the user and the computer itself (such as UNIX, Microsoft, etc..)

---

[5] Exact definition taken from http://www.webopedia.com/TERM/N/NNTP.html

OWA – Outlook Web Access
This is a service of Exchange that allows a user not directly connected to the network access to their exchange account. IIS must be installed for this to work.

PDC – Primary Domain Controller
The "first" server in a domain, acts as a logon server and replicates domain information

PKI – Public Key Infrastructure
Asymmetric Cryptography

RDS – Remote Data Service
Allows queries to be conducted directly against an ODBC data source

SCM – Security Configuration Manager
Allows addition of MMC snap-in

SMTP – Simple Mail Transport Protocol
Protocol that allows email to be sent/receive from clients to servers, also via the Internet.

SNMP – Simple Network Management Protocol
Allows remote administration of machine through use of various tools

SP6a – Service Packet 6 A
Microsoft's release of various OS patches (not necessarily security patches) that are needed in order for the OS to function properly with many of the newer programs.

SRP – Security Rollup Package
Microsoft's hotfix release that encompasses all security hotfixes before its release

SSL – Secure Socket Layer (https://)
Uses port 443 to establish a secure connection with a remote web server

TCP – Transmission Control Protocol
Most widely used Transport layer (Layer 4 of the OSI model) protocol

TLS – Transport Layer Security
Successor to SSL

VM – Virtual Machine
VM acts as a separate computer. Java applications run in their own Java VM that has no access to the OS

W2K – Windows 2000
Microsoft's OS

WFP – Windows File Protection
W2K utility that replaces deleted or changed system files (Also known as SFP, System File Protection and SFC, System File Checker)

WWW – World Wide Web
Vast collection of web pages on the internet

Section VI – Research Sites and papers

Microsoft Corporation – Information on WinNT Server 4.0 Security
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/chklist/nt4svrcl.asp

Microsoft Corporation – Information on IIS 4.0 Security
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/chklist/iischk.asp

Microsoft Corporation – TechNet
http://www.microsoft.com/technet

Microsoft Corporation – Library
http://msdn.microsoft.com/library

Netscape Communications Corporation – SSL
http://developer.netscape.com/tech/security/ssl/howitworks.html
http://developer.netscape.com/docs/manuals/security/sslin/contents.htm#1

Napernikov, Boris (2001) "What Does It Take to Harden an IIS Web Server?"
http://rr.sans.org/

Cooper, Russ (2001) "10 Steps To Better IIS Security," September, Information Security Magazine. www.infosecuritymag.com/articles/september01/features_IIS_security.shtml

Caesar (2000) "Managing (& Disabling) Windows File Protection (WFP)
http://www.arstechnica.com/tweak/win2k/others/disable_sfp-1.html

McClellan, Keith (2002) "Windows 2000 Memory Subsystem Tweaking"
http://www.arstechnica.com/tweak/win2k/others/memory-2.html

Rescorla, E (2000) RFC2818 "HTTP over TLS"
http://www.ietf.org/rfc/rfc2818.txt

Wayne's Windows NT Resources for Administrators and Users
Homepage - http://is-it-true.org/nt/index.shtml
ERD page - http://is-it-true.org/nt/atips/atips23.shtml

Last Updated: July 23rd, 2017

Upcoming Training

SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event
Community SANS Omaha SEC401*	Omaha, NE	Aug 14, 2017 - Aug 19, 2017	Community SANS
Community SANS Trenton SEC401	Trenton, NJ	Aug 21, 2017 - Aug 26, 2017	Community SANS
Virginia Beach 2017 - SEC401: Security Essentials Bootcamp Style	Virginia Beach, VA	Aug 21, 2017 - Aug 26, 2017	vLive
SANS Chicago 2017	Chicago, IL	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
Community SANS Pasadena SEC401 @ NASA	Pasadena, CA	Aug 23, 2017 - Aug 30, 2017	Community SANS
Mentor Session - SEC401	Minneapolis, MN	Aug 29, 2017 - Oct 10, 2017	Mentor
SANS Tampa - Clearwater 2017	Clearwater, FL	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS San Francisco Fall 2017	San Francisco, CA	Sep 05, 2017 - Sep 10, 2017	Live Event
Mentor Session - SEC401	Edmonton, AB	Sep 06, 2017 - Oct 18, 2017	Mentor
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
Mentor Session - SEC401	Ventura, CA	Sep 11, 2017 - Oct 12, 2017	Mentor
Community SANS Albany SEC401	Albany, NY	Sep 11, 2017 - Sep 16, 2017	Community SANS
Community SANS Columbia SEC401	Columbia, MD	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Dallas SEC401	Dallas, TX	Sep 18, 2017 - Sep 23, 2017	Community SANS
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Copenhagen 2017	Copenhagen, Denmark	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Boise SEC401	Boise, ID	Sep 25, 2017 - Sep 30, 2017	Community SANS
Baltimore Fall 2017 - SEC401: Security Essentials Bootcamp Style	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
Community SANS New York SEC401	New York, NY	Sep 25, 2017 - Sep 30, 2017	Community SANS
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Sacramento SEC401	Sacramento, CA	Oct 02, 2017 - Oct 07, 2017	Community SANS
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Oct 02, 2017 - Oct 07, 2017	Community SANS
Mentor Session - SEC401	Arlington, VA	Oct 04, 2017 - Nov 15, 2017	Mentor