Global Information Assurance Certification Paper

CASE STUDY: Securing a Network of Windows 2000 and 2003 Servers using HFNETCHKPRO4

GIAC Security Essentials
Certification (GSEC)
Practical Assignment
Version 1.4c

Option 2 - Case Study in
Information Security

Submitted by: Debbi Atkins
Location: Lone Star 2004
Mar. 12th, 2005

Table of Contents

Abstract/Summary

There is a very simple way to maintain complete and total security of any Windows 2000 or Windows 2003 server. All one needs to do is to make it a stand-alone machine with no network connectivity and the job is done. That would be a fail-proof solution, but it is not a realistic one. In the real world, servers are connected to a network. At a minimum, the servers are exposed to risks from an internal network. In most locations there is also Internet connectivity, thus exposing the servers to any number of continually evolving external threats.

Security is an essential part of maintaining a network. There are multiple layers of security required at both the hardware and the software levels, as well as at both the server and the network hardware level. This case study describes the process by which we recognized the need for a patch management solution on the Windows servers at my company, the decision to use HFNETCHKPRO by Shavlik Technologies to provide that functionality, and the implementation of the solution.

Before

Current Security Posture

The network that is discussed in this case study is owned by a municipal government. There are approximately 40 physical sites where computers are located. All of these sites connect back to a central location. At the time this project began, the connectivity was either by T-1 or fiber to our central location. Servers were located at many locations, though all mission-critical servers resided in data centers that were at the more centralized locations. The primary data centers were connected to each other by a fault-tolerant single-mode fiber run. Other sites were connected to our central site by T-1 lines that terminated at our primary data center. Users at remote sites accessed servers at the primary data centers in addition to using dedicated file and print servers at some of the remote locations. Network traffic from all locations accessed the Internet after passing through our primary data center. There were (and still are) multiple levels of network security such as firewalls, transaction zones, DMZ's, and gateway products that are outside the range of this case study.

When this project began, we had approximately 130 servers running Windows 2000 Server, Windows 2000 Advanced Server, Windows Server 2003, and a legacy Windows NT 4 system. We had 2 Windows 2000 Active Directory forests, each with a single domain. These forests were separated by firewalls and had no trust relationship to each other. For security reasons, we had a few

servers that were not a member of either domain, though they did reside on the same physical network as one of the Active Directory forests.

We bought our hardware from Tier-1 hardware vendors and installed the operating systems ourselves. We used a document compiled in-house to ensure that our servers were built uniformly and that the final build was consistent with our required security. The configuration of the server's hardware and operating system was determined by the application that would run on the server. We have a standard build for all web servers, for all SQL servers, for all application servers, etc., and that build is driven by requirements for proper disaster recovery of the system as well as by application requirements. Vendors can request more fault tolerance than what is in our standard build, but not less. At the time a server was built, all current patches available on the Windows Update web page were applied.

We used a major software vendor's antivirus solution on all desktops and servers on our network. The antivirus solution updated both the software engines and the definition files multiple times a day. Though this generated some additional network traffic, the benefit of deploying definition files quickly after they were released by the vendor was important to us. Email passed through an additional layer of antivirus filtering before it was passed out of our email servers to the user's PC to be read with their Outlook client.

We placed servers into production in a secure state. We had confidence in the antivirus system we had implemented. Yet, we knew that we still had a problem. The security of the systems was of paramount importance; however, we did not have a method by which to maintain a server in a secure state after it went into production.

Problem Description

Simply put, we had no consistency in our patch management. More accurately, we had no patch management at all. After the initial server deployment, patches were applied using the Microsoft Windows Update website at (http://windowsupdate.microsoft.com). When a member of our team knew of a specific risk to a server based upon the purpose of the server, (for example, if a new SQL exploit were known to be in the works), we would attempt to update all of our SQL servers with the required updates. This was a very time-intensive way to implement newly released patches. It was not consistent and was certainly not managed in any practical way. Even worse, there was no documentation maintained on patches that had been implemented.

Each department within the city has a unique function that often requires unique software. Many of the applications running on our servers are intended for one specific purpose and cannot be carried across to other areas. Small software companies thrive in this type of environment. There are companies that develop

software for libraries, for animal services, for Parks and Recreation, for environmental waste, for time and labor issues specific to public safety (police and fire departments), etc. Many of these small companies do not test Microsoft patches quickly, if at all. Carry that across to the many departments needed to operate a municipal government, and you have a patch management nightmare.

We also have servers that run software from major vendors such as PeopleSoft, Microsoft, and Oracle. Each vendor has its own timeframe for testing new patches released by Microsoft and approving those patches to be applied to systems running their software. Applying a patch that has not been approved by the vendor may keep your server free of exploits, but it can void a software maintenance contract.

Current Risks

Patches Not Applied:

Servers were frequently not patched after being deployed into the production environment. This left us vulnerable to any number of exploits. Though we felt confident that our antivirus solution would protect us to a high degree, there are too many exploits for which the most up-to-date antivirus solution is of no use. This is because some exploits take advantage of operating system weaknesses, many of which would not be a risk if all Microsoft patches were applied. As an example, the SQL Slammer exploit of January 2003 took advantage of a SQL buffer overrun for which Microsoft had release a fix in July of 2002. (http://www.microsoft.com/technet/security/bulletin/MS02-039.mspx) Even though administrators had six months to patch servers running Microsoft SQL 2000, it was estimated that 150,000-200,000 servers worldwide were infected in the first day of the Slammer attack (http://www.pcworld.com/news/article/0,aid,108988,00.asp)

Patches Are Applied:

This one seems quite at odds with the previous section. There is, however, a risk involved with applying a patch just because it shows up on the list when you run the software on the Windows Update page. Some patches that show up as being required or recommended using Windows Update may change the version of software that is installed on the server. As an example, the software application installed on your server may not be approved for MDAC 2.8, yet running Windows update will show the need to update your MDAC component if you are at a back level. We have one product running on our network that required Internet Explorer 5. Not IE 5.1, not IE 5.5., but IE 5.0. Since this application ran in a Citrix environment, it was critical that any patch that raised the version level of Internet Explorer on those servers not be applied.

We have a number of vendors that will not approve a patch to be applied for significant periods of time after the patch is released by Microsoft. We have some vendors who will not test patches at all. They only test service packs, but no patches released between service packs. That issue is addressed later in this paper.

A problem occurring because of either of the above situations could result in significant downtime for our servers at best and of complete system failure at worst.

Lack of Documentation:

We did not have an accurate and/or current list of our network servers. Different people on our network knew what was on different servers, but nowhere was there a list of all servers. The city had consolidated technology services into a central department rather than having each department hire its own IT personnel. They had also migrated from Novell to Microsoft and from Groupwise to Exchange. With these major changes came a need to learn new systems and a need to fully document what was contained on those systems. A solution was needed to maintain those systems as securely as reasonably possible.

During

Proposed Solution

Many times administrators have prior knowledge that a vulnerability is going to be exploited, and we spend hours, if not days, attempting to get our systems and networks protected before the exploit reaches our network. On one such occasion, our team of network engineers worked a 36-hour shift applying updates to our servers. Though we were successful in protecting our servers, it was an extreme burden on our users, our families, and our bodies! Once I recovered from that ordeal, I began to research products that were available to ease the burden of patch management. This case study is not intended to weigh the pros and cons of the solutions I investigated, but rather to discuss the implementation of the solution I ultimately selected, HFNetChkPro by Shavlik Technologies. (http://www.shavlik.com/hfn_windows.aspx)

A few of the reasons that I selected this product are:

- It can run with a backend SQL server database or a local Jet database.
- It applies only those patches that are for security purposes and does not

apply enhancements to installed products. If you have IE 5.5, it will report patches to that version of IE, but will not attempt to upgrade you to IE 6.x. This was important to us because we have some vendors who are slow to approve updates to patches and version upgrades for software. (Refer to the earlier section "Patches are Applied")

- It integrates with Active Directory.
- It has an interface to allow one to enter alternate credentials, which is important for our non-domain servers and for our second Active Directory forest.
- Multiple administrators can run Shavlik consoles and apply patches to servers at the same time (licensing for additional consoles is required).
- Shavlik reports on and applies patches to all the Microsoft operating systems we have on our network, both server and client. It also reports on and patches Microsoft applications (Exchange, SQL, Biztalk, Office products, etc.). This allowed us to fully protect our servers from within a single interface and to expand our patch management to our client PC's at a later date without having to learn a new product.
- Shavlik does not require an agent to be installed on our servers. We can enter a server name, an IP address, or a range of addresses. Shavlik finds the machines indicated and scans them for required patches.
- Shavlik has several built-in reports. Of most interest to me was one that can give either a synopsis or a detailed report of missing patches. (Remember my earlier comment about vendors that don't permit updates to servers hosting their applications)

Solution Implementation

Step one in implementing our Shavlik solution was to compile a complete and accurate list of all Microsoft Windows servers. For this I created a spreadsheet using Microsoft Excel. The workbook had several pages, each of which listed servers that had a common purpose. All domain controllers were on one sheet, all servers that had a network administration function were on one sheet, all file and print servers were on one sheet, etc. Each sheet had the same columns, the purpose of which was to assist us each time that we held a "patch management weekend". The columns I chose to include and their purposes are listed below:

- Server Name
- Operating System: To include the O/S version such as Advanced Server, Enterprise Server, etc.
- Server Location: Indicates the data center or remote site where the server is physically located.
- Application/Service: This column includes all applications or services that may affect which patches are applied. Examples of network services are DNS, DHCP, or Domain Controller. Applications may be Microsoft applications (Exchange or SQL and the version of each) or vendor/3rd

party applications.

- Contact in Technology Services: The name of the person in our department or the section within our department that can best help to resolve any problems with the server. This person/section is also contacted before any maintenance is done to the server. For example, we have many servers that are used to operate our 911 call center and other servers specific to fire department and police department purposes. There is a section of Technology Services that works specifically with the applications on the public safety servers. They are separate from the team of network engineers that maintains the integrity of the operating system.
- Contact Before Updating: This column gives the name of the city employee outside of our department that needs to be contacted before updates are applied to the server. This column also lists the vendor whose approval is required before any new patches are applied.
- History: This column title is given to all subsequent columns. Each time we hold a patch management party, we add a new column titled "History". If all available patches are applied, the date of the patch management party is put into the column ("Updated xx/xx/xxxx"). If no patches are applied, the column background is changed to red and "NO PATCHES PERMITTED" along with the date is typed into the column. If some, but not all patches, are installed, the column background is changed to red and all patches left off are listed, as is the date.
-

Step two of the implementation was to download a fully functional trial version of the software from the Shavlik website. This is available to anyone, though one must register his or her information with the company before being able to download the product. The product can be downloaded at http://www.shavlik.com/pdownloadform4.aspx . I initially installed the software on my desktop and ran it using a Jet database. This works as well as the SQL product for actual deployment in small networks but isn't as versatile. For our full implementation we required a network installation of HFNETCHKPRO and a SQL database. We wanted multiple engineers to be able to participate when we were having a patch management weekend, and we wanted a single database to show the state of all servers.

I used the test product to push patches to a few desktops. This also offered an opportunity to read the documentation on the product and to become familiar with the steps necessary to apply patches.

Step three was to install the trial version of the software on a server. For this deployment we chose to share a server that ran the management component of the product that provided our antivirus solution. After the installation, we had to configure the product. This is not referring to the configuration needed on the server when installing HFNETCHKPRO, but rather it means configuring Shavlik to be able to scan our network of servers. Shavlik offers the ability to create

machine groups. The computers within the groups can all be scanned at the same time. We created machine groups to coincide with how servers were categorized on the Excel sheet I discussed earlier. All Domain Controllers are in one group, all file and print servers are in one group, all Exchange servers are in one group, etc. We created a separate group for a particular section of servers that we are generally not permitted to update quickly.

Step four was to test the product on a small number of servers. We have several servers that are used for network management. Some examples of these are domain controllers, DNS servers, DHCP servers, and servers that run our management components for antivirus software. These servers are backed up regularly as a part of our disaster recovery plan. They are also servers whose outage has a low impact on our end users because multiple servers provide these services. I chose to run a small, private patch management party on these servers before trying it network-wide.

During the initial attempt at using HFNETCHKPRO, patches were deployed to each server individually. The Shavlik product is versatile and one can scan computers individually or as a group. Even when scanned as a group, patches can be applied individually, to a machine group, or to a subset of the scanned group. The product can be used to deploy patches immediately, and patches can be deployed with or without a reboot. One can also schedule the deployment of patches for a later time, again with or without a reboot. For our purposes on servers, we do not schedule deployment of patches. We always are onsite in our office for a server reboot so that we can confirm the status of the server and can test the applications after the installation of a patch.
The limited test was successful. All servers came back online, no patches failed to install, and all services functioned properly on the servers.

Step five was to schedule a full-blown patch management party. I schedule our patch management sessions for Saturdays. Though we have departments that run seven days a week, 24 hours a day, we impact the fewest users on the weekends. This also allows for disaster recovery time in the event that we have a server failure. Though we have had an occasional hardware failure on a server reboot such as a hard drive in a RAID array, we have not had a failure that could be attributed to the deployment of patches. Even so, we continue to schedule patch management so as to allow the maximum amount of time for the recovery of a server to its previous state. Shavlik allows for the rollback of a patch if the vendor permits rollback and that happens as quickly as a rollback and reboot. We choose to also allow time for a system recovery if a new patch causes an application to fail so completely that we must go into a full disaster recovery of a server.

A date was chosen and, using the contact information listed in my Excel spreadsheet, the necessary personnel and vendors were contacted. They were advised that we needed an affirmative or negative reply to the request that we be

permitted to apply all current Microsoft patches to servers running their applications. The vendors were given a deadline by which to answer our inquiry. We have some applications that are critical to their users, others that are a bit finicky in how they work. For those systems, we schedule for the member of our department with primary responsibility for the application to attend the patch party with the members of our team.

Our initial patch party was more difficult than subsequent sessions have been. This was because of the previously mentioned problem with our servers often not having been patched since deployment, and almost certainly not being patched completely to current standards. Though Shavlik uses the q-chain technology that allows patches to be deployed without constant reboots, we still had to reboot several times. Shavlik can be used to deploy both service packs and patches. Service packs get installed individually, each followed by a reboot. We then rescanned and applied the next service pack until all service packs were installed. Installing service packs before patches makes patch deployment much easier, particularly on systems that haven't been patched frequently, since older patches will often be included in a service pack. We always finish with a "clean" scan, one that shows there are no more service packs or patches that need to be installed.

Knowledge of the systems and applications was critical at this point, as was having the detailed and accurate spreadsheet of all servers. It permitted me to see which servers had not been approved for specific patches or, in some cases, for any patches. Applying all allowed patches and service packs took us about fourteen hours, significantly less than the pre-Shavlik time of thirty-six hours. Subsequent sessions have been completed in as little as seven hours for a network that now includes close to one hundred sixty servers. The times listed include all testing required after patches are installed to confirm that applications are working properly.

Step six was a decision making process. We had fully evaluated the product and were pleased with the performance. The cost of the product was easily offset by the benefits gained by increasing the level of security on our servers. For our city it was also important that the process of patching all servers to their highest allowed level was decreased from 36 hours to an average of 7-10 hours.

Step seven in our implementation was to purchase a fully licensed copy of Shavlik HFNETCHKPRO and a dedicated server on which to install the application. Though the Shavlik product can share space on a server, our network at the time did not have a server that we were comfortable placing this application on. The decision was made not because of Shavlik's overhead on the system, but because of the overhead of the applications already running on the servers used for network management.

Step eight was the installation of the product on the dedicated server. This

server ran Microsoft SQL 2000 and held the Shavlik database as well as the application itself. We installed additional consoles on the desktops of the people who would be called in to work when we scheduled patch management sessions. Experience showed us that a feature of Shavlik that greatly speeds up the deployment of patches is the ability to download all patches to a local machine by creating a download center on that machine. When Shavlik deploys, it then pushed those files to the server being updated rather than having every server go to the Internet to pull down each and every patch or service pack being applied. We have each engineer create a download center on his or her PC. It is possible to share a common download center on a server; however, when multiple users are pushing patches, the bandwidth from that server can be utilized to the point that deployment time is negatively impacted.

Step nine required the installation of HFNETCHKPRO on a laptop. As mentioned previously, we have two Active Directory forests. One of these forests sits within a DMZ, and we do not permit access into that location from outside of the firewalls except for very limited circumstances. The laptop installation uses a locally installed Jet database. It takes advantage of the feature of the Shavlik product that allows one to download the patches to the laptop, thereby creating a local download center. I can carry my laptop and plug into the switch inside of the secure DMZ. Prior to plugging into this secure network, I confirm that my antivirus is up-to-date and I run a full system scan to insure that my laptop is clear of viruses. I verify that my laptop is fully patched, and I confirm that I pass a scan using software that finds and eliminates spyware. I also download the most current Shavlik XML file and confirm that I have updated my local patch download center to include all current patches.

By default Shavlik will attempt to authenticate to a server using the logged on credentials. Authentication is necessary before a successful scan is performed. Since I do not join my laptop to the AD domain that exists in the DMZ, I use the option provided by Shavlik to designate specific credentials. I enter my logon for that AD forest within the DMZ, and it successfully and securely scans and deploys updates to all servers without risking them to the exposure required to open them to the Internet, even for the brief time required to apply updates. The unique logon credentials can be applied on a machine-by-machine basis or to a group of machines.

It took approximately eight weeks to complete this project through step eight. That included the time from the beginning of the research to the completion of the configuration of this product. Step nine is separate from the implementation itself, but along with step five gets repeated with each patch management party we schedule. (You may have noticed that I call them patch management parties rather than patch management sessions. Patch management is what you make it, and with the ease of patch management since deploying HFNETCHKPRO, we choose to make it a party!)

After

Solution Testing and Validation

The success of a patch management system is a difficult thing to prove definitively. The most that can accurately be said is that we have quietly passed by all exploits that have hit networks worldwide since the deployment of HFNETCHKPRO on our network.

Each member of our team has now been trained on the use of the Shavlik product. We receive emails from Shavlik and from Microsoft advising us of new patches that have been released. As members of Microsoft's Premier Support program, I receive a phone call when a patch is received that is of such critical nature that an immediate deployment of the patch is recommended. We also ask each member of our team to register to receive emails when a new Microsoft security bulletin is released.

Approximately one year ago we had a non-employee plug his personal laptop into a port at one of our public libraries and attempt to infect our network with a variant of NIMDA. We have since enhanced the port security at the libraries to protect the network from future incidents of that type, but this incident reinforced my belief in the success of our implementation of Shavlik HFNETCHKPRO. Though we had some desktops become infected, not one server on our network was impacted by the virus. This was possible because Shavlik had allowed us to maintain our servers in a secure and patched state.

Within the past few months I received a call from our Microsoft Technical Account Manager advising that a patch was being released outside of the normal patch release schedule. The patch was of a highly critical nature, and it was suspected that an exploit was being release within a matter of days to take advantage of the vulnerability. Since we had current documentation of our network servers, we were able to quickly contact everyone whose approval was needed before applying patches. We scheduled a patch management party for that same weekend, secured all the servers we could, and documented which vendors denied us approval to implement the patch. We had no impact from the exploit.

You may notice that I don't list specific dates, specific exploits, specific application, or specific vendors. This is done to protect the servers and applications on my network.

Risk Assessment

The network of servers for which my team and I have responsibility is now in a much more secure state than it was prior to the deployment of Shavlik HFNETCHKPRO on our network. At the time of the initial deployment, we did not have a single server that had all applicable patches and service packs deployed. We had no standards to show to our application vendors when they balked at having patches deployed on the systems that housed their applications. We did not have a system in place by which we could hold our software vendors or ourselves accountable for the security of our servers. We always knew that our servers might potentially be damaged by any exploit written to take advantage of the latest and greatest exposed vulnerability. This meant that there was a chance that we might fail to meet our obligation to our users to provide them with a stable and secure network of servers, and therefore provide them access to the applications required to perform their duties.

We now have a single location that documents our history of patch management. More specifically, we have an Excel spreadsheet that lists the dates that patches were applied. We know that if only the date is listed in the history, then that server was fully compliant with patches and service packs on that date. When we see a red cell in the history column for a server, we know that something was not applied and that the missing item (or items) will be listed in the cell on the worksheet.

We now have the means to run reports that will give a complete list of patches and service packs missing from a server, whether to the operating system or to a Microsoft product installed on the server. We have used the reporting functionality to create reports to send to vendors. If necessary, we will create a detailed report that lists not only the patch, but also a complete report detailing the vulnerability and the risk it exploits. When Shavlik finds a patch, whether it is already installed or it needs to be installed, the Shavlik product provides links to relevant articles and websites in the patch description. There will be a link to the relevant Microsoft knowledge base article, a link to a Microsoft security bulletin if one exists, a link to a page on the CVE (Common Vulnerabilities and Exposures) website (http://www.cve.mitre.org/) if applicable, and a link to a BugTraq ID webpage (http://ntbugtraq.ntadvice.com/) if one relates to the patch. All of these resources are valuable not only for becoming knowledgeable on the vulnerability, but also for using as ammunition when a vendor disputes your claim that a patch needs to be applied to a system. For the most difficult of vendors, I have been known to run fully detailed reports that document all the above listed information (which can run 35-50 pages per server when all exploits and vulnerabilities are completely detailed) and request that the report be shipped to the vendor, along with notification that we will hold them responsible if our network is compromised because of their refusal to permit us to deploy service packs and patches.

Recommendations:

Everyone who is responsible for managing a network, whether small or large, whether Windows-based or not, should have a written procedure for ensuring that his or her servers are properly and securely patched and maintained. A critical component in maintaining a secure network is learning the vulnerabilities to which a network is exposed. There are many mailing lists available to security professionals, some of which are specific to an application or operating system, others that cover a broader range of topics. Some of the security sites that I have found worthwhile are listed below. It is best to read the list rules before subscribing to a mailing list since some are more restrictive than others.

http://ntbugtraq.ntadvice.com/ : This list is for the discussion of security exploits and security bugs in Windows NT, Windows 2000, and Windows XP plus related applications. There is a link on the front page for subscribing to this listserv.

http://www.microsoft.com/technet/security/bulletin/advance.mspx : A link off of this page will take you to a subscription page for receiving advance notification of security bulletins. This subscription requires a Passport login. For those who prefer not to take that step, you can read all MS security bulletins at http://www.microsoft.com/security/default.mspx

http://www.cert.org/ : CERT covers all areas of security and offers papers on best practices as well as security bulletins. This website is operated by Carnegie Mellon University.

http://www.sans.org : SANS offers training in several areas of security and certifications that are respected industry-wide. One can follow links on this website to subscribe to newsletters, to read articles pertaining to security, or to learn about the training and certifications available.

http://www.shavlik.com : The website for Shavlik Technologies provides a link to whitepapers for patch management and a link to a security bulletin search where one can search and view bulletins by product. This is the website for the company that developed the HFNETCHKPRO product. HFNETCHKPRO users will want to subscribe to the email list to receive a notification when Shavlik releases a new XML file to coincide with the release of Microsoft patches.

Conclusion

The deployment of Shavlik HFNETCHKPRO on our network has greatly enhanced our ability to provide our end users with a secure and reliable network

of servers. At the time of my writing this case study, we have increased the number of Windows Server 2003 on our network and the remaining Windows NT 4 Server has been removed from service. We continue to use the Shavlik product and can quickly plan and successfully carry out a patch management party. We learned the importance of maintaining a current list that includes the required contacts both inside the city/company and at the software vendors. Our list continues to be maintained and updated as servers are deployed or removed from service. Our end users appreciate the reliability of the product that we provide and have commented on the high availability of the network. Our vendors have learned the importance that we place on maintaining our servers in a fully patched and secure state. I can't say that it is because of our persistence, but many of the vendors that we deal with seem to be quicker at approving patches than they once were! I learned a great deal about the best way to manage a network while deploying this product. I use the product to continue learning more, particularly taking advantage of the links available within the Shavlik HFNETCHKPRO product for researching information relating to specific patches.

There are thousands of people around the world who are at this very moment trying to find a way to exploit any vulnerability they can expose within some version of a Windows operating system. I am confident that with our current implementation of Shavlik HFNETCHKPRO we are up to the challenge presented by those who would attempt to release a destructive piece of software into our network.

References

http://windowsupdate.microsoft.com

http://www.microsoft.com/technet/security/bulletin/MS02-039.mspx

http://www.pcworld.com/news/article/0,aid,108988,00.asp

http://www.shavlik.com/hfn_windows.aspx

http://www.shavlik.com/pdownloadform4.aspx

http://www.cve.mitre.org/

http://ntbugtraq.ntadvice.com/

http://www.microsoft.com/technet/security/bulletin/advance.mspx

http://www.cert.org

http://www.sans.org/