# Global Information Assurance Certification Paper

# LEARNING THE HARD WAY: A CASE STUDY OF LEARNING BASIC SECURITY PRACTICES FOR THE MICROSOFT WINDOWS ENVIRONMENT

By Christi Rowekamp

GSEC CERTIFICATION ATTEMPT
Case Study version 1.4b

# LEARNING THE HARD WAY: A CASE STUDY OF LEARNING BASIC SECURITY PRACTICES FOR THE MICROSOFT WINDOWS ENVIRONMENT

September 2001 is when it all started. The pagers start at 7 am and don't stop until 2 am the next day. Our large (400 Windows servers), multi-site, and naive environment had been struck by Nimda. One by one our servers were falling at a rapid pace. We were unprepared, and we needed answers quickly. Answers that would not only enable us to fight the battle in progress, but answers on why we were vulnerable. Business was dramatically affected at each site and upper management needed an explanation. The following case study will not examine Nimda itself, but examine the points of failure in our security practices that caused our vulnerability, explain the immediate actions taken to overcome the virus, and then follow up with our current best practices.

## FAILURE 1- LACK OF DOCUMENTATION

We had just finished a major reorganization effort. The new administrative teams were assembled in a central location, and a project to move the equipment to the same location was in the beginning stages. So we were unprepared in the worst way for Nimda. The documentation for the previous environments was not current or non existent. We did not have a security policy in place, or any written policies for configuration management, administrative control or escalation of support. We did not have any accurate information on server names, locations, configuration, or even the internal applications support contacts. Because of these failures we had no way to effectively target the most critical servers in our clean up efforts. We could not forewarn the proper support groups or report to upper management what servers were at risk and how many had been compromised. We could not dispatch teams to remote sites appropriately because we were unsure of how many servers were in each location and how many could have been compromised.

## FAILURE 2- SERVICE PACKS AND PATCHING

How horrible to find out, and have to admit, that the service packs and security roll ups that would have prevented the Nimda attack (or at least contain it) were available months before the incident.[1] Prior to the Nimda attack the applications groups controlled many of the servers. If they were not comfortable with their testing of a service pack, or felt they could not take the time to test or take the outage on the server for the service pack to be installed, their opinion took precedence over the network administrator. A majority of our Windows 2000 servers had no service pack applied or service pack 1 only. Service Pack 2 was available in May 2001 and would have prevented Nimda. 60% of the Windows NT 4 servers in the environment had service pack 6a, available November 1999, but no security roll up, available July 2001, which would also have prevented vulnerabilities to Nimda.

**FAILURE 3- NO ANTIVIRUS**

Prior to the attack of Nimda, **not one** non messaging server in the enterprise had antivirus installed. It had been the opinion of the administration that the risks and obstacles associated with antivirus were worse than the risk of infection. We could not have been more wrong. Perhaps that would not have been such a ludicrous idea if the servers had been hardened against attack.

**FAILURE 4- CONFIGURATION MANAGEMENT AND STANDARDS**

Prior to Nimda, many people were responsible for the configuration of new servers and maintenance of the current environment. Each administrator, at each site, had their own idea of how a server should be configured based on past experience and current knowledge. Although we have many intelligent administrators, without standards there was no way to measure if the configuration or activity on a server was "normal" or compromised during the incident. In addition, developers with MSDN subscriptions, were building servers on desktops and using them as their daily workstation. The developers, also knowledgeable in their own realm, did not have the server experience necessary to build or maintain a secure server. Applications support personnel had administrative rights on servers, and we had 50+ domain administrators. We did have a password policy in place, but it was limited to domain accounts. Our local account passwords, although strong, did not change regularly.

**FAILURE 5- AUDITING**

Prior to the attack of Nimda, there was not proactive monitoring happening inside the firewall. We had a team of people monitoring at the firewall but internally we did not have a good grasp of what was "normal" on the servers. No event log monitoring was being done on a regular basis, even on the "important" servers. While Nimda was attacking, we were unable to determine if the security log failures were acceptable or not. Many servers did not have auditing configured at all. The system and application event logs were cluttered with other error events, so they too were not helpful.

**CLOSING OPEN DOORS- HOW WE WON THE BATTLE**

Our first step was to get organized. We created strategic teams. We assembled a team at the center that researched the issues, determined what steps needed to be taken next, and provided upper management with communication as we made progress. We assembled server field teams that would eradicate the virus, update the servers with antivirus, service packs and security patches, and provide very basic documentation of the servers in each building. The desktop teams in each building were responsible for the same tasks as the server field teams, but in the desktop environment. My responsibility was to the center team. This section explains how each team contributed to restoring our compromised environment and the tools used to accomplish the tasks.

The center team used sites like CERT.ORG[3], and NAI.COM[2], to gather information on what Nimda did, and how to eradicate the virus.

***NIMDA Fact 1: It starts with an email.*** We immediately had the messaging teams block email that matched the description of the Nimda messages.

***NIMDA Fact 2: It uses IIS to spread.*** We needed to secure the IIS servers first. One of the most helpful tools we used at this point was a product called HYENA[4]. Hyena is a centralized management and export tool for Windows environments. Using Hyena provided us with our basic server list including patch levels, and services in use on both the servers and workstations. This enabled us to provide direction to the field teams and gave us a point of reference for the applications group as they began reporting denial of service. From this master server list, we created a visual "dashboard" for upper management in excel color coding infected servers red, unknown status or unknown location yellow. Remediated or unaffected servers were coded in green. A summary with totals from each group was labeled clearly at the top of the sheet. Upper management was updated with this list every hour. Emails were sent every few hours to update the associates on the details of the virus, how they could help, and when systems were expected to be functional again. After the first 2 hours under attack we had a solid vision of what had been compromised at that point, where a majority of the servers were, and what needed to be done next. The field teams were dispatched immediately. Another useful tool was Dameware Mini Remote Control[5]. Dameware Mini Remote Control helped us remediate the vulnerabilities of systems that were not easily accessible due to location, or had an unknown location (i.e. the servers built by the developers on desktops, etc.) Dameware Mini Remote Control remotely installs its service and allows control over the machine as if you were standing at the console. This tool works for NT4 Workstation and Server, Windows 2000 Pro and Server, and .Net. We were capable of all tasks necessary to secure a server using Dameware Mini Remote Control. With this tool, the center team was able to help secure servers while maintaining communication and guidance using this outstanding tool.

The server field teams first assigned a documenter. This person was responsible for documenting infections and environments including the primary purpose of each server, if it was easily recognizable. The documenter reported progress to the center team and assisted the center team in updating the application support personnel on the status of their servers. Our teams of administrators could not spend the time needed to document all servers during the attack, but did take quick notes on any garish configuration problems. As we applied patches and antivirus to clean up after the attack, not one server, out of 400, had an issue with the service packs or antivirus programs applied. This was a huge victory for the administrators. That fact helped us to regain ownership of the servers, a critical step for future vulnerability prevention and configuration management.

Our desktop field teams quickly confiscated all desktops built with a server OS. They were immediately rebuilt with Windows 2000 Professional. They used SMS to immediately roll out the necessary service packs to the workstations (Windows

2000 Professional or NT4 Workstation) that were running IIS. They also used one person as a central point of contact and documenter to report back to the center team.

Because of our organized approach we had secured all compromised machines within 12 hours of receiving the first page. It took 2 additional hours to complete remediation on known servers. At that point, the field teams were released from duty. The center team stayed 5 more hours to complete the servers accessible only through Dameware Mini Remote Control, to complete reports for management and to monitor the environment. Since this experience we have changed our practices dramatically.

**HOW WE INCREASED SECURITY AWARENESS**

It will never happen to me were no longer words in our vocabulary. Security had not been a priority. No one ever had enough time for research, writing policies or implementation. That had all changed after the rude awaking provided by Nimda in September 2001. I was assigned as the primary Windows 2000 security administrator, the day after our Nimda event. This section will provide a snapshot of how we turned our failures into best practices.

**ACCOMPLISHMENT 1- DOCUMENTATION**

A few months after Nimda, our security team in conjunction with business leaders from the entire enterprise, published a security policy. The policy includes details regarding remote access, password policies, physical security for equipment, data center access and many other guidelines. Not only did it include guidelines for all electronic equipment and data security, it provided consequences for not adhering to the policy. It is available on our company's intranet, and has had appropriate sections emailed to all associates so there is no question about where we stand on security as an organization.

We have also created a master server list. It includes all servers, locations, IP's, primary function, applications support contacts, OS versions and service pack levels. We have tied the server list into our Change Management procedures so it is updated consistently.

We have created a standard Windows 2000 server build. We have 2 documents supporting this standard configuration. One document is for administrators, this will be discussed in more detail in a later section. The second is still in development. It is a general server standards document to be published on the intranet. This will document minimum requirements for application groups, outlining standard maintenance, required maintenance schedules, and configurations that will be permitted on the network. These standards will also be discussed the upcoming sections.

## ACCOMPLISHMENT 2- SERVICE PACKS AND PATCHING

This task was one of my critical initiatives. I began researching patch management and applied what I had learned immediately using tools and services available for free or at a reasonable cost.

I found 2 services that keep me educated. I subscribed to Microsoft's Security Bulletin Service[6]. This service emails administrators each time a Microsoft vulnerability is announced, and keeps the administrator up to date on available Microsoft Operating System and Microsoft Office patches. I also subscribed to CERT.org[7] advisory newsletter a similar service, that announces vulnerabilities, and viruses found on Microsoft or non Microsoft operating systems.

Although I have experimented with many tools for patch management, currently I am most satisfied with HFNETCHKPRO[8]. This is the full version of the Shavlik Technologies tool Hfnetchk[9]. Hfnetchk, available from Microsoft is a command line tool used to check patch levels. HFNETCHKPRO available for purchase from Shavlik Technologies at http://www.shavlik.com/security/prod_hf.asp automates the assessment of Microsoft patch levels, patch application, and reporting. This is done without an agent and is a great value. (There is a free evaluation program called HFNETCHKLT at the same URL) An automated procedure created a Win/Win situation for the company. The company is secure and no additional administrators were required to increase our level of security.
Politically, applying patches has become easier in our environment than pre-Nimda. Nimda provided the administration teams with the undeniable proof that service packs are necessary and need to be applied in a timely manner. The ongoing argument with the applications support teams was now over. Today, a service level agreement has been reached between the server administrators and the applications teams. This agreement allows the administrators to patch, a minimum of once per quarter for the internal servers and once per month on the DMZ servers. It has increased server uptime and made troubleshooting easier when problems do arise.

## ACCOMPLISHMENT 3- ANTIVIRUS

We clearly learned our lesson on antivirus, and officially created a team for antivirus support. This team includes at least one member of our security team, technical design and planning team, server operations team, desktop team and SMS team. We meet when necessary, and have weekly conversations with our antivirus vendor. We have created a multi tiered system for ensuring antivirus is updated and running properly. Updated dat files are pushed weekly via SMS. Hyena reports are run to verify that the services are still running on all servers without error. Scan logs are read regularly on file servers, web servers and any critical servers. Logon scripts check not only if antivirus is present and running, but check to be sure it is current. In addition, we have blocked all dial in access to those who are not running our current antivirus program. This multi tiered approach was proven a success when we did NOT have a single report of the Klez[16] virus in the environment.

**ACCOMPLISHMENT 4 - CONFIGURATION MANAGEMENT**

Immediately after the attack, configuration management became a number one priority. Build procedures were created and documented. Build auditing has been implemented. In addition, repercussions for not adhering to configuration guidelines have been outlined. All server personnel contributed to a standard build document for Windows 2000 and agreed that no more Windows NT4 servers were to be deployed. It is a living document, owned by one person. All administrators can contribute to the document but unless the suggestion is accepted by all, it does not become part of the standard build. This document included general hardware and OS settings, SNMP settings and security considerations. A few of the more basic guidelines we use are listed below:

- ***Disable unnecessary services*** [11 & 12]. - Telnet, Simple Mail Transport Protocol, Automatic Update service, FTP, WWW.
- ***Build Servers based on function.***
- ***If IIS is not required, remove it***. -It is installed by default in Windows 2000
- ***Harden the local security policy***[13]- if Active Directory is not in place. If it is, use Group Policy and security templates to harden the system. For example, do not allow anonymous access, remove users and power users from the log on locally user right (remember how easy it was for us to use Dameware?), and remove the right to do remote shutdowns (this specifically helped us during the Shatrix [14] virus outbreak.)
- ***Terminal Services*** [15]- In application or remote administration mode, increase the RDP encryption, and control who is allowed to use it!
- ***Manage the local administrator password***- it changes every 30 days. (Hyena can be used for this task across multiple servers)
- ***Restrict Access to the Event Logs***- (condensed from the Windows Registry Guide[16] (http://www.winguides.com/registry/display.php/351/)) Event logs can expose configuration information to outsiders. There is no reason why anyone outside of your administrators need access to the logs. This requires an additional registry key for each log.

  **SystemKey:**HKEY_LOCAL_MACHINE\SYSTEM\Current ControlSet\Services\EventLog\
  **ValueName:**RestrictGuestAccess
  **Data Type:** DWORD Value
  **Value Data:** 0=guest access, 1=restricted access

In addition to using a standard configuration, all new servers go through an audit process to ensure all configurations adhere to the standards document and all security vulnerabilities have been remediated. The auditor uses the build document to audit the configuration first. The audit is first done by the building team (not the builder), and then a final audit is done by the design and planning security administrator. During the final audit, a free vulnerability scanning tool is

used called Nessus[17]. Nessus is a client-server product that scans for all vulnerabilities on multiple operating systems, from port scanning, to denial of service simulations. (There is an option to do "safe checks" so you can prevent an actual outage while still testing for vulnerabilities.) The ability to write plug in's for the tool in C is also available. The reports provided by Nessus include the explanations, and if possible resolutions, for the vulnerabilities detected, with hyperlinks! The audits are one way we control consistency in the environment. We have also started using disk imaging to increase configuration integrity. We are not yet capable of imaging all hardware, but have seen a reduction in TCO since using imaging. It has reduced the time necessary for setting up servers, and ensured that server configurations are identical. We update the images every time a new security patch is released or a vulnerability discovered. All servers inside the firewall are scheduled for quarterly maintenance to keep security vulnerabilities to a minimum and patches up to date. The servers in the DMZ are updated monthly, and we take advantage of any reasonable, scheduled server outage to update patches more frequently. Finally, we resolved our abundance of administrators with domain admin privileges. Today we have 15 domain admins, down for the original 50+.

The desktop and developer built servers were also addressed. Server operating systems may only be built on server hardware. All servers need to be secured in the server data center and must be built by server administrators. Any rogue servers found will be confiscated and immediate action with HR will be taken. This has also proven to reduce TCO. The developers have reliable hardware and a standard configuration in which to build their applications. With this controlled development environment, their applications have been more reliable and easier to support.

**ACCOMPLISHMENT 5 – AUDITING**

Auditing our event logs has provided our teams with a new knowledge of our environment. One site we have used is http://www.eventid.net[18]. This site is dedicated to event log entries, possible causes and resolutions. We use Event Comb[19], included in the Microsoft Security toolkit, to search for systems that have logged specific events. We have used this auditing and general event log monitoring to determine unusual behavior. Because of auditing we have the ability to determine that application X may cause events that are similar to virus Y or that systems, X, Y, and Z, have a specific vulnerability. We have used it to monitor changes made that have not passed through our Change Management process, and to determine who was responsible for the change. We have also used it to resolve enterprise wide configuration issues, and it has simplified troubleshooting production problems. The security team has used it to track administrative rights abuse. I believe this task alone has contributed to increased stability and supportability in our environment, because it forced us to expand our knowledge of our environment.

## ACCOMPLISHMENT 6 (A BONUS) - EDUCATION

Our Nimda incident pointed out the lack of attention to and knowledge of the world of security by our company. This lack of knowledge and urgency was shared by administrators and management. This was a crucial turning point for our company. Security has become a critical initiative at all levels. We now have budgets for tools and education. We currently have 6 administrators that have gone to SANS training alone, 4 of us are working toward certification. Our SANS training specifically strengthened the knowledge we gained through our Nimda experience, and helped us to see other weaknesses in our procedures before a hacker did. SANS training also gave us the hands on experience necessary to understand tools available and the risk associated in using those tools. I personally found the greatest benefit in talking to other administrators. We were able to share experience and suggestions. We expanded our vision of the impact security, or lack of security, can have not only in our own environment, but how your weaknesses can potentially effect other environments. As administrators, we discuss information security news almost everyday. We have joined Infraguard[20], a coalition between civilians and the FBI to increase information security awareness. One of the benefits of this membership is the NIPC newsletters sent daily on specific topics or areas of business. We have designed a security page and published it on our intranet. The page includes information on viruses and home user security suggestions. Educating the users has had an additional benefit; it has promoted IT- associate relations. Opening the channels of communications has made security a team effort between the associates and IT staff. Shared ownership and communication has contributed to the reduction of virus incidents in the environment and the volume of calls due to hoaxes.

## CONCLUSION

Our large environment was fertile ground for viruses because of our lack of security knowledge and priority. Nimda exposed all of our weaknesses. We had no documentation: no policies in place, no centralized team, no incident survival plans and no basic documentation of the environment. We were far behind on our patching because we had no plan, and no means of managing patch application. Our environment was unmanageable during the attack of Nimda because we did not know our environment well enough. Without auditing and monitoring, or the most basic configuration management, we had no baseline for normal system behavior. Yet, our greatest failure was our ignorance and "it will never happen" to me attitude. We've learned some security basics the hard way, and then continued the learning through research, and training classes. Today, the administration teams own the servers. A "checks and balances" tiered approach to security exists and has prevented many incidents. These best practices have made our environment more stable and support has been easier for developers and administrators alike. Through the Nimda attack we discovered our failures and brought security to the forefront of our day to day tasks where they belong.

***Footnotes***

[1] http://support.microsoft.com/default.aspx?ID=FH;EN-US;sp&FR=0&SD=GN&LN=EN-US&CT=SD&SE=NONA
[2] http://vil.nai.com/vil/content/v_99209.htm
[3] http://www.cert.org/advisories/CA-2001-26.html
[4] http://www.systemtools.com/hyena/
[5] http://www.dameware.com/products/
[6] http://register.microsoft.com/regsys/pic.asp
[7] http://www.cert.org/contact_cert/certmaillist.html
[8] http://www.shavlik.com/security/prod_hf.asp
[9] http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/tools/hfnetchk.asp
[10] http://vil.nai.com/vil/content/v_99367.htm
[11] http://www.microsoft.com/mspress/security/tips/031402.asp
[12] http://www.zdnet.com/products/stories/reviews/ 0,4161,2564838-1,00.html
[13] http://www.windows2000faq.com/Articles/Index.cfm?ArticleID=15317
[14] http://vil.nai.com/vil/content/v_99291.htm
[15] http://www.advanced-concepts.com/Products/terminal_services.htm
[16] http://www.winguides.com/registry/display.php/351/
[17] http://www.nessus.org/download.html
[18] http://www.eventid.net
[19] http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tips/Manage.asp
[20] http://www.infragard.net/

Last Updated: July 25th, 2017

# Upcoming Training


| | | | |
|---|---|---|---|
| SANS Boston 2017 | Boston, MA | Aug 07, 2017 - Aug 12, 2017 | Live Event |
| SANS Prague 2017 | Prague, Czech Republic | Aug 07, 2017 - Aug 12, 2017 | Live Event |
| Community SANS Omaha SEC401* | Omaha, NE | Aug 14, 2017 - Aug 19, 2017 | Community SANS |
| 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 |
| SANS Adelaide 2017 | Adelaide, Australia | Aug 21, 2017 - Aug 26, 2017 | Live Event |
| 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 |
| 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 San Francisco Fall 2017 | San Francisco, CA | Sep 05, 2017 - Sep 10, 2017 | Live Event |
| SANS Tampa - Clearwater 2017 | Clearwater, FL | 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 |
| Community SANS Albany SEC401 | Albany, NY | Sep 11, 2017 - Sep 16, 2017 | Community SANS |
| Mentor Session - SEC401 | Ventura, CA | Sep 11, 2017 - Oct 12, 2017 | Mentor |
| 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 |
| 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 |
| 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 |
| 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 |
| Community SANS Sacramento SEC401 | Sacramento, CA | Oct 02, 2017 - Oct 07, 2017 | Community SANS |
| Mentor Session - SEC401 | Arlington, VA | Oct 04, 2017 - Nov 15, 2017 | Mentor |
| SANS Phoenix-Mesa 2017 | Mesa, AZ | Oct 09, 2017 - Oct 14, 2017 | Live Event |