Global Information Assurance Certification Paper

Case Study: Using the Information Security Cycle to Protect a Small Network

GSEC Practical, Version 1.4b, Option 2
Submitted May 1, 2004
Monique Rae Nelson

Abstract

The purpose of this paper was to implement the steps of the Information Security Cycle (prevention, detection, response) to protect a small business network. I was able to create a security plan that included receiving alerts and bulletins, enhancing firewall rules, vulnerability scanning, implementing an intrusion detection system, monitoring network traffic using network sniffers, improving event log analysis, network mapping, and creating an incident handling procedure. The steps that were taken allowed me to improve network security with minimal cost to the company. The tools used have allowed me to be more informed about the security of the network and also to be more efficient with my time. This case study has provided me the opportunity to be more proactive instead of reactive with regards to network security.

Before Snapshot

We are a small software development company that has twenty employees. The company is made up of six departments: administration, marketing, sales, software development, project management, and technical support. Most users are highly skilled with a computer and are able to maintain their own computers with regards to installing critical updates, not opening questionable emails, etc. When anything looks suspicious, they contact me immediately. Social-engineering is not much of an issue because of our company size and the users knowledge of what to watch out for.

There are fifty-seven devices that make up the Local Area Network (LAN) and De-Militarized Zone (DMZ). The number of devices attached to the network is higher than other small companies due to the nature of the business. The company has never had any type of formal security plan for the systems or network. Without a plan in place, vulnerabilities are not addressed in a timely manner and successful attacks may go undetected. A systematic, cost effective, and time-saving procedure to secure the small network needs to be developed. By using the information security cycle (prevention, detection, and response), I will construct a comprehensive plan to better protect the network (LAN and DMZ) and create a balanced approach to security.

At this time, the highest security priority is to minimize outside attacks from the Internet to our network. Any security process that is put in place must not only provide a way to enhance the current security configuration, but do so with as little administrative time as possible. Setting up and configuring automated tools and scripts will be a high priority. Automation will allow me to do more in less time.

I will be handling this entire case study on my own, as I am the only IT person at my company. While I have been a network administrator for approximately three years, I have not had the network security knowledge to address these concerns. I have read articles (online and in print), but have not taken any formal network security training until now. The SANS course has helped me tremendously in creating a base of knowledge from which to draw. I also find that my previous law enforcement experience helps me with the investigative aspect of network security.

Network Description

There are multiple vulnerable systems currently on our network. We have Mail, FTP, and Web servers available through the Internet. All computers on our network have access to the Internet. All servers are either running Windows 2000 Standard Server or Windows 2003 Server. Other network services/software available are Active Directory (2003), Antivirus (corporate and Exchange Server versions), DHCP, DNS, Exchange Server 2003, file servers, FTP, GIS mapping software, IIS 5 and 6, Outlook Web Access (OWA), print server, SQL Server, and WINS.

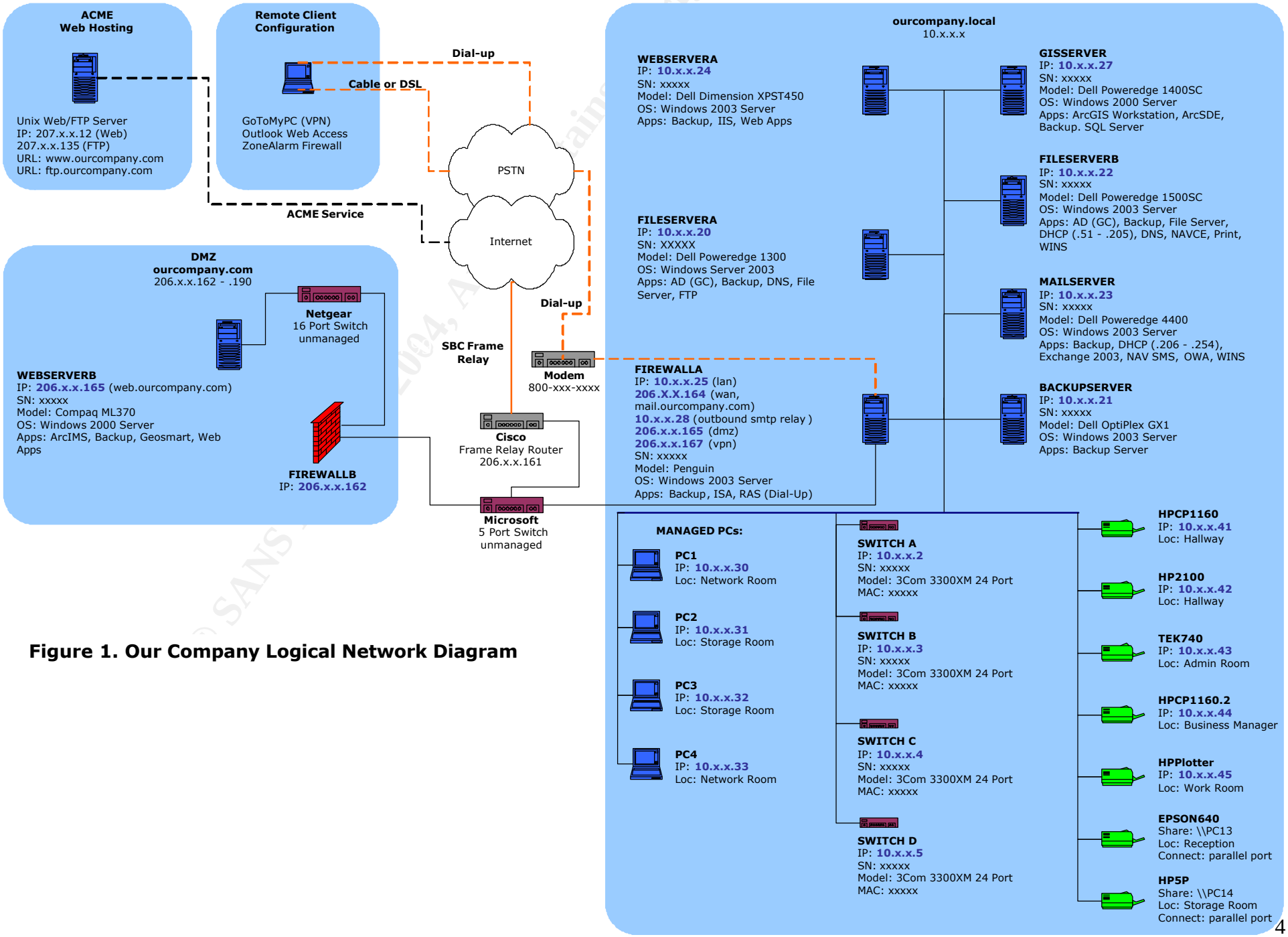

Figure 1. Our Company Logical Network Diagram

Security Position

The current security position is more of a 'Security Lite' approach. Only a few preventative steps are performed. The Antivirus software is set to automatically update the virus definition files daily to try to keep the possibility of a virus infection down to a minimum. The definition files are downloaded from Symantec to MAILSERVER and FILESERVERB. Then FILESERVERB pushes out the updates to all of the Antivirus software clients attached to the network. Emails that contain attachments with certain file extensions (*.exe, *.bat, *.vbs, *.txt, *.com, *.pif, and *.scr) are deleted automatically before they reach their recipient, because they are rarely valid file types for our business to be receiving from anyone. If a new vulnerability is announced and I know about it, critical systems are patched immediately. Otherwise, critical servers are manually scanned using Microsoft Windows Update[1] service for missing critical updates at least monthly.

During Snapshot

Due to the limited funds available to purchase 3rd party commercial tools, I located inexpensive, open-source, or freely available scripts and software. My idea was to create a toolbox for myself that would help me stay on top of the security issues and also become more efficient with my time. I looked for tools that were automated and scriptable, if at all possible. Before I started installing any tools or scripts, I received permission from the President of the company to perform vulnerability scans, network sniffing, and create snapshots of any servers.

Step One: Prevention

The information security cycle, which has three steps, was used to create a comprehensive plan to secure our network. The first step is Prevention, in which the goal is to prevent attacks from happening. It is also the most cost-effective countermeasure. Items included in the prevention plan are alerts and bulletins, Antivirus software, firewall rules, and vulnerability scanning.

1. Alerts and Bulletins

Due to the many different types of email alerts and bulletins, my focus was on receiving information about new virus threats, vendor-specific bulletins for new product vulnerabilities (Cisco and Microsoft), and vendor-neutral bulletins specifically covering network and software application security vulnerabilities where the vendor does not have a bulletin service available (3Com, NetGear, and Zoom).

Virus Alerts:

I subscribed to two sources of new virus information. If one of the alerts turns out to be more comprehensive than the other, I may decide to choose the more complete alert notification and discontinue the other. The two sources are:

a. Symantec Security Alert[2] emails
b. Panda Software's Virus Alert Newsletter[3]

Vendor-Specific Bulletins:

I subscribed to Cisco's and Microsoft's bulletins because it is important to get direct information from the vendors. Cisco issues Security Advisories for security issues that directly impact Cisco products when action is necessary to repair the Cisco product. Cisco sends out Security Notices for issues that require a response to information posted to a public forum, or recommendations to mitigate general problems affecting network stability. Microsoft notifies subscribers when they release an important security bulletin or virus alert, and also makes subscribers aware that they might need to take action to guard against a circulating threat. The two vendor-specific bulletins are:

a. Cisco Security Notices[4]
b. Microsoft Security Notification Service[5]

Vendor-Neutral Bulletins:

Lastly, I subscribed to a vendor-neutral bulletin that provides information about current security issues, vulnerabilities, and exploits for software applications and hardware devices. I also have an account on the SANS web site, so that I can build a personal SANS portal page. The portal page contains different kinds of security-related information from which I can pick and choose from. The two vendor-neutral bulletins are:

a. US-CERT Technical Cyber Security Alert[6]
b. SANS Portal[7] which includes "@RISK: The Consensus Security Alert" and other helpful information

2. Antivirus Software

It is important to prevent known viruses, worms, and Trojan programs from infecting the company systems. Currently, Symantec Antivirus Corporate Edition is used to protect the personal computers and network servers. Symantec Mail Security for Microsoft Exchange is used to protect the Exchange Server from virus threats. If a new virus definition file is available, all servers and clients are updated with the definition on a daily basis. Using this combination, there have not been any virus outbreaks so far. Therefore, I plan to use the Symantec products to protect the systems from future virus threats.

3. Firewall Rules

In the past, the firewall configuration contained rules that allowed only a few protocols (smtp, pop3, imap4) from the Internet to the LAN, but was allowing all outbound traffic from the LAN to the Internet. A new firewall was purchased that provides a user-friendly interface and also allows quick review of historical or live

network traffic for all network adapters installed. When installing the new firewall, additional rules were added to protect the LAN and prevent a compromised computer on the LAN from attacking a computer(s) on the Internet. The principle "Deny all unless explicitly permitted" was utilized when developing the new rule set. The new firewall rules created are as follows:

a. Allow Outbound HTTP [LAN to WAN]: 80, TCP
b. Allow Outbound HTTPS [LAN to WAN]: 443, TCP
c. Allow Outbound SMTP [Email Server to WAN]: 25, TCP
d. Allow Outbound FTP [LAN to WAN]: 21, TCP
e. Allow Outbound DNS [Internal DNS Servers to External DNS Servers]: 53, TCP & UDP
f. Publish SMTP Server [WAN to Email Server]: 25, TCP
g. Default Rule Deny All Protocols from All Networks to All Networks

4. Vulnerability Scanning

The type of vulnerability scanner I was looking for had to scan the Windows operating system and other Microsoft products. The scanner functions needed to:

a. Identify active systems
b. Identify ports that are available on the active systems
c. Identify the running services (including the version and vulnerabilities)
d. Identify unpatched vulnerabilities
e. Identify misconfigured or poor security settings

I found several potential vulnerability scanners that would perform the necessary tasks. I chose Microsoft Baseline Security Analyzer v1.2 (MBSA) [8] based on its cost (freely available) and because MBSA could run via a script, batch file, or a Graphical User Interface (GUI).

Microsoft Baseline Security Analyzer v1.2 (MBSA)

MBSA can perform local or remote scans of Windows systems. MBSA runs on Windows 2000, Windows XP, and Windows Server 2003 systems and will scan for common system misconfigurations in the following products: Windows NT 4.0, Windows 2000, Windows XP, Windows Server 2003, Internet Information Server (IIS), SQL Server, Internet Explorer, and Office. MBSA 1.2 will also scan for missing security updates for the following products installed on the network: Windows 2000, Windows XP, Windows Server 2003, IIS, SQL Server, IE, Exchange Server, Windows Media Player, Microsoft Data Access Components (MDAC), MSXML, Microsoft Virtual Machine, and Office.

MBSA fits well with our particular network configuration because we have many Microsoft products throughout the network. MBSA was installed on WEBSERVERA server using the installation procedures from the Microsoft web site[8]. Sample scripts[9] were also downloaded and copied to a newly created c:\scripts\mbsa folder.

MBSA was run once using the GUI so it would create the "SecurityScans" folder under the C:\Documents and Settings\<user name>. The "SecurityScans" folder stores the report .xml files for each computer that is scanned.

Several benchmark tests were completed to determine how much time it would take to run different types of scans:

TYPE	# OF SYSTEMS	SCAN DURATION
All Servers and Managed Systems	11	5 minutes
Client Computers	30	16 minutes
All Servers and Computers	41	23 minutes

Figure 2. MBSA Scan Benchmark Tests

The four reports that will be generated are Server Critical Summary Report, Server Misconfiguration Summary Report, Client Critical Summary Report, and a Client Misconfiguration Summary Report. The steps to setup the automated scans and generate the four report summaries are as follows:

a. Create a "Servers" and a "Clients" folder in C:\Documents and Settings\<user name>\SecurityScans.
b. Copy the contents of rollup.js to "rollup_servers.js" and "rollup_clients.js" in the c:\scripts\mbsa folder.
c. Edit rollup_servers.js to include the new patch for completed server scans.
d. Edit rollup_clients.js to include the new patch for completed client scans.
e. Copy scripts included in Appendix A to c:\scripts\mbsa.
    i. batchMbsa_Servers.bat
    ii. batchMbsa_Clients.bat

Please note in the scanning scripts, the contents of the C:\Documents and Settings\<user name>\SecurityScans\<Servers or Clients> folder is deleted before the scan is actually launched. Therefore, previously scanned computers would not show up in the summary reports.

Daily scans are now performed during non-business hours for all servers and managed systems. The client computers are scanned weekly when most of the staff is at lunch, so as not to impact their productivity. All scanning and reporting scripts are launched using the Windows Task Scheduler.

The summary reports have provided a way to quickly view the status of the servers and client computers. All the custom reports generated are located in the c:\scripts\mbsa folder. Shortcuts were created on my desktop to each custom report. In order to view each scanned computer's individual report, a shortcut was created to the mbsa.exe file. The mbsa.exe shortcut was moved to C:\Documents and Settings\<user name>\SendTo. Additional shortcuts to the

…\SecurityScans\Servers and … SecurityScans\Clients folders were created. To view a scanned computer's individual report in C:\Documents and Settings\<user name>\SecurityScans\<Servers or Clients> without opening the MBSA GUI, right-click the individual report of interest and click "Send To" and then click on the MBSA shortcut.

Check	Critical	Warning	Passed	Note	Error	Informational	Not performed	Total
Windows Security Updates >>				8 (100%)				8
Macro Security >>	1 (13%)	1 (13%)					6 (75%)	8
Windows Media Player Security Updates >>			8 (100%)					8
Exchange Server Security Updates >>			1 (13%)				7 (88%)	8
Office Security Updates >>			1 (13%)				7 (88%)	8
Automatic Updates >>		1 (13%)		7 (88%)				8
SQL Server/MSDE Security Updates >>	1 (13%)	1 (13%)					6 (75%)	8

Figure 3. Sample Summary Report

Note: In order for the whole process to work, rollup.xslt must be in the c:\scripts\mbsa folder as well as the scripts mentioned above.

Step Two: Detection

The second step in the Information Security Cycle is Detection, in which the goal is to detect a network breach. Items included in the detection plan are Intrusion Detection System (IDS), network sniffers, event log analysis, and network mapping.

1. Intrusion Detection System

An intrusion detection system monitors either a network (NID) or a host (HID) for indicators that an attack has been launched. A NID system is not a good choice for our network, as we have a high-speed, switched network. Also, this would add more complexity to the network and require more administrative time. On the other hand, a real-time HID system is too expensive to implement. Instead, I focused on creating a HID system in which the goal was to create a baseline for typical host activity so that we had something to compare future host activity against. Appendix B contains a Visual Basic script (snapshot.vbs) that collects information that needs to be captured from a local or remote system. This information is typically included when conducting a system audit. The collected information may assist in detecting system compromises and may even become legal evidence if a security breach is confirmed. The snapshot.vbs script creates a text file with the following information:

a. Computer name
b. Snapshot start date and time
c. Script name used to create snapshot
d. User running the script
e. Domain
f. Operating system version and service packs
g. Local user account information
h. Local groups and their memberships
i. Local shares
j. Processes and their properties
k. Services and their properties
l. Networking configuration settings
m. Environmental variables
n. Startup commands (startup folder and registry)
o. All file properties on c: drive (size, last modified date, hidden attributes)
p. Snapshot completion date and time

Once the first baseline file was created by the script, subsequent snapshots were scheduled using the Windows Task Scheduler. The script is executed on all servers on a daily basis. All snapshots are generated in a compressed NTFS folder. The naming convention used for the snapshot files was snapshot_<servername>_<date>.txt (i.e. snapshot_GISSERVER_03282004.txt).

If there are any noted differences between the previous day baseline and the current day baseline for the same server, a file is generated by a batch script. The information can be examined via a shortcut to the snapshot folder to verify if the snapshot differences were caused by malicious means or an authorized change. The following are some items that would warrant further investigation:

a. Unauthorized changes
b. Unusual processes and services
c. Unusual files
d. Unusual network usage
e. Unusual scheduled tasks
f. Unusual accounts

2. Network Sniffers

Network sniffers are designed to capture all packets that are sent and received on a network interface of the host that has the network sniffer installed. The challenge was that the company has a switched network. With a switched network, only broadcast and multicast traffic is sent to all ports. In order to capture unicast traffic between two ports, a few connection modifications needed to be made. First, a hub was plugged into an existing switch on the network. Then, the host computer with the network sniffer installed was plugged into the hub. Next, the FIREWALLA's LAN interface was then plugged into the hub. As long as the host computer's

interface card was in promiscuous mode, all traffic moving between FIREWALLA and the LAN could be monitored.

Ethereal v0.10.3

Ethereal[10] is an open source packet sniffer with versions for Windows and Unix. There are several different filters that can be created to filter out TCP, UDP, ICMP, and IGMP to potentially reveal any odd traffic. Ethereal was used to analyze the traffic that was being sent and received through the firewall. The suspicious activity to be identified included the following:

a. Blatantly crafted packets that violate the IP protocol standards (i.e. outbound packets with a source IP that is not in our address space).
b. Loopback address (127.0.0.1) on the network may indicate an improper configuration or an attack
c. Unusual or suspicious use of TCP flags (i.e. SYN FIN, FIN, and no flags set)
d. Sequential port traffic (may indicate a potential port scan)
e. Monitor critical systems (web, DNS, or mail server) for TCP, UDP, ICMP traffic

A baseline was created for typical traffic and a file was saved for future comparison to other traffic captures. A schedule was setup to analyze the network traffic at least weekly.

3. Event Log Analysis

The event logs provide clues to potential security breaches. The difficulty is managing all the separate event logs on each server. Appendix C contains a script created to combat this problem. The script copies all of the event logs from each server and puts the information into an Event Log database (Microsoft Access). Once the information is copied, the script automatically clears the server's event logs. Moving the contents of the event logs and then clearing them makes it harder for malicious users to tamper with the event log contents when trying to cover up their trail. The script runs every hour by using the Windows Task Scheduler on each server.

Field Name	Data Type	Caption
AutoNum	AutoNumber	
RecordNumber	Number (Long Integer)	
Date	Date/Time (Short Date)	
Time	Date/Time (Long Time)	
Type	Text	
SourceName	Text	Source
EventCode	Number (Integer)	Event ID
Message	Memo	Description
ComputerName	Text	PC
User	Text	

Figure 4. Event Log Database Table Structure

Several queries were created in the Event Log database. Now information can quickly be sorted and queried in different ways. For instance, one of the queries only shows audit failures, errors, and warnings and then is sorted by date and time. Developing the Event Log database allows me to identify:

a. Failed logon events
b. System reboots or shutdowns
c. Unauthorized access attempts
d. Unexplained entries

All unknown entries are researched further using the EventId[11] web site or using the Google[12] "Web" or "Groups" search.

4. Network Mapping

A program that would list IP addresses for computers attached to the network, listening TCP ports, network shares (including system and hidden), and detect Media Access Control (MAC) addresses was needed. The SoftPerfect Network Scanner[13] provided all of the functionality required.

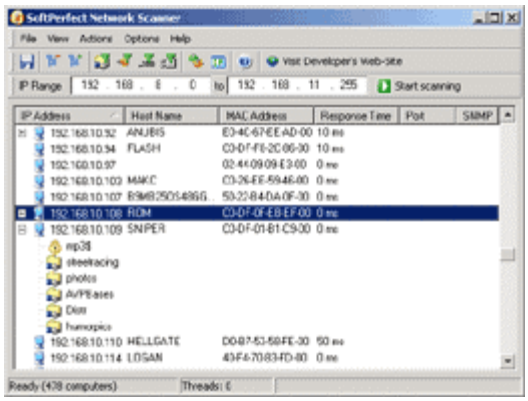

Figure 6. SoftPerfect Network Scanner

The Network Scanner runs via its GUI in addition to being launched through a script or batch file. The idea is to create a network baseline by saving the results of a scan and then comparing it to a previous scan. Appendix D contains a script that is launched daily using the Windows Task Scheduler. The script generated a new scanned text file and compared it to the previous baseline. Any difference between the two files is copied to a new text file for later review. Once a comparison file is created, a review for any unauthorized devices or configurations can be conducted.

Step Three: Response

There are different types of incidents that the company needs to be prepared to handle. Incidents can range from a natural disaster to a malicious network attack and the indicators may vary. Any unexplained entries in the Event Log database, server snapshot files, or network map files can indicate the network has been compromised. To better prepare for an incident, an incident handling procedure was created for the company.

To develop the incident handling procedure, the six steps outlined in Chapter 10: Incident Handling Foundations from SANS Security Essentials with CISSP CBK[14] were followed. The steps include preparation, identification, containment, eradication, recovery, and follow-up. Each step and how it was utilized to develop the incident handling procedure for the company is outlined below.

1. Preparation

A plan was developed to try to prevent the inevitable network attack. Information was collected along with tools to assist during an incident. All items were then organized and placed in a "war bag". The "war bag" contains important items to have close at hand which reduces time required to manage an incident when it occurs. The "war bag" contains the following items:

a. Contact List - included company employees, vendors, and emergency services (local law enforcement computer crime unit, local FBI computer analysis and response team, etc.). Each contact contained the contact's name, phone numbers, pager numbers, fax, and email wherever available. It was also noted if a staff member could respond on short notice.
b. Equipment Inventory List – included computer hardware and software.
c. Location of Off-Site Media Storage – procedure to retrieve copies of software or backups if we needed them.
d. Network Diagram – included a logical and physical network diagrams
e. Password list
f. Check Lists – included how to take a system offline, how to restore a server from a backup tape, etc.
g. Office Supplies – included notepads, pencils, pens, paperclips. It is important to document everything during an incident, including who, what, where, when, and how.

h. Hardware – network cables (straight and cross-over), hub, hard drive, screwdrivers and other PC tools.
i. Software – network security tools

2. Identification

It is critical to verify whether an incident has in fact occurred. The process outlined in "Step Two: Detection" in this paper was used to identify indicators, such as:

a. Alert from the IDS
b. Virus alert with a confirmed virus infection
c. Network sniffer abnormalities (i.e. rogue systems connected to the network)
d. Unexplained entries in the Event Log database
e. Hacker tools installed on a system
f. Snapshot comparison file indicating:

    i. Unauthorized applications in the systems startup process
    ii. Unauthorized processes
    iii. Accounts with unauthorized permission changes

3. Containment

Containment is the method of isolating an incident. There are several different ways to contain an incident. The first step is to secure the area by verifying that no unauthorized persons have physical access to the network. The next step is to backup the affected system on new media for evidentiary reasons. One option is to backup and then remove the hard drive, while keeping the original hard drive as evidence. It is important to be sure to maintain a chain of custody for any evidence collected. The second option is to create two backup copies of the hard drive, while keeping one of the backup copies for evidence and another backup copy for analysis. The third step is to run the snapshot.vbs script to determine what the attacker may have altered on the affected system. The fourth step is to decide whether an affected system should be pulled off of the network or should the entire network be disconnected from the Internet. This step depends on the severity and number of affected systems. The fifth step is to change the existing passwords to avoid allowing the attacker to logon to the affected system. It should be noted that restarting the computer, logging on and logging off should be avoided as it may launch malicious code.

4. Eradication

A determination of the type of attack must be made in order to eradicate the source or cause of an incident. Once the vulnerabilities are identified, contact should be made with the appropriate management personnel and then follow the steps identified by the vendor or resource to resolve the problem. It is important to

identify all systems involved in the attack. If additional vulnerable systems are found, follow the "Containment" steps above before continuing on to "Eradication". After the vulnerabilities are addressed, run MBSA to confirm that all issues are resolved and no new vulnerabilities are added.

5. Recovery

Once the eradication is complete, the system must be recovered and its services restored. Recovery may include restoring the system using a previous known good backup tape or some other means. The steps to recovery are to:

a. Create a snapshot after the recovery of the system.
b. Compare the vulnerable snapshot and the recovered snapshot to determine if the system may still be vulnerable.
a. Document any differences, as it may provide clues to how the attack occurred.
b. Create a full backup of the recovered system on new media as soon as possible.
c. Test the recovered system and ensure that the system's applications are functioning properly.
d. Monitor all recovered systems for the first few hours for re-infection.
e. Put the system back on the network and/or connect the entire network to the Internet after monitored systems appear healthy.
f. Monitor the network to ensure that the incident has been resolved.

6. Follow-Up

To prevent future attacks, a complete review of the incident is critical. All notes need to be compiled into a comprehensive incident log. A summary report needs to be created so that management understands how the event occurred and what steps will be taken in the future to prevent a similar attack. The summary report would contain how the incident occurred, recovery steps, how future attacks would be prevented, how to improve incident response, and money saved by the current incident handling procedure. Any staff members involved in handling the incident would contribute to any logs or reports.

AFTER SNAPSHOT

By using the Information Security Cycle, I have been able to increase the security throughout our network. Many of the new tools used have been automated which allows me to do more work in less time. Now I am able to have more time to be proactive instead of only reactive. In the future, I plan on expanding the functionality for many of the steps previously outlined.

Alerts and Bulletins

The alerts and bulletins have assisted me in staying informed regarding any known vulnerabilities to the network. I have found that one alert or bulletin is not able to provide information on all known vulnerabilities or viruses at any given time. Having comprehensive information has allowed me to keep the staff informed of any new vulnerabilities or viruses. If there are any steps that I need to take to ensure that the network is less vulnerable, I can implement a new procedure immediately. On the other hand, I have learned that receiving too many alerts and bulletins can be problematic because all day can be spent trying to keep up with the enormous amount of information being disseminated. It is important to stay focused on the information pertinent to your network.

Antivirus Software

No changes were made regarding antivirus software because the processes in place work well for the company. The virus definition files are updated daily, if a new definition file is available. Whenever additional steps can be taken to prevent a virus outbreak, I make the necessary changes.

Firewall Rules

The new firewall rules have provided better network security. By limiting the inbound and outbound protocols entering our network, it is easier to manage the monitoring of the network traffic traveling through the LAN and WAN interfaces on our firewall. Each packet is identified by the rule that has allowed or denied its connection in the firewall log. Also, by limiting the protocols leaving our network we can be a better Internet neighbor. We can ensure that if one of our systems becomes compromised, our firewall is able to eliminate most threats going to other systems on the Internet.

Vulnerability Scanning

Using MBSA for vulnerability scanning has made monitoring of the servers and client computers much easier. Having the automated script running the scans and then creating the reports has been very helpful. It is like having an additional network administrator assisting me with gathering the information. The best part about this solution is that it did not cost anything, except a little time on my part to install the software and create the scripts.

In the future, I would like to install and setup Nessus[15]. Nessus is another freely available vulnerability scanner that uses client server technology. The server application runs on a Linux system and the client application runs on various Windows and Unix systems. Nessus may be able to find vulnerabilities that MBSA may miss because the Nessus security checks database can be updated on a daily basis. Vulnerability scanning is very critical to maintaining secure systems both for

the systems that are directly available to the Internet and also in case an attacker is able to get beyond our firewall.

Intrusion Detection System (IDS)

The IDS snapshots have been very helpful in documenting the server's configurations. While the system is a historical and passive system, it does allow me to review any changes to a server that could have been caused by an attacker. I would like to add the capability of emailing any results from the snapshot comparisons to me in the future as well as add more system checks to the script. This solution would make it quicker to review the information. Another option I may look into in the future is using Snort[16]. Snort is an open source NID system. I will need to determine if the network switches that we have on our network can mirror traffic, so that the port that Snort would be listening on can see all the network traffic.

Network Sniffers

Ethereal was a very easy to use application. It made monitoring the network traffic easier and allowed me to save captured data that I could review or compare to other captures later. The down-side was using a hub to make network sniffing possible. The hub added an additional single point of failure and was prone to data loss due to collisions. In the future, I would like to research our network switches to find out what if any functionality they have to allow monitoring of all data traveling on the LAN from a port on a switch.

Event Log Analysis

The Event Log database in Access has been tremendously helpful. It has been much easier and quicker to review server events in the database as opposed to having to go to each server and review its individual logs. I would like to migrate the database to SQL Server for improved performance as the database grows. I would also like to create more queries and reports to be able to view quickly the critical information. If possible, I would like to add the capability to send each event automatically, as it is generated on the server to the database. This eliminates the need to run the export script every hour and makes it more difficult for an attacker to cover their tracks by tampering with the server's event logs. I would also like to create some type of notification system to send an email regarding suspicious events as they occur and are added to the database. This addition will change the database from being a passive system to more of a real-time system.

Network Mapping

Network mapping has allowed me to keep a closer eye on what systems are attached to the network. This type of scan in the past was rarely done and I did not have the means to compare a previous file and then create a new file with any differences. In the future, I would like to change the schedule of creating a network

map file and the comparison from a daily event to an hourly event. The SoftPerfect Network Scanner creates minimal network traffic for the type of scan that is performed. It could very easily be changed from a daily scan to an hourly scan. I would also like to add the capability to email any comparison results to me as well. Otherwise, I still will not know when unauthorized devices have connected to the network, unless I manually look at the comparison file every hour.

Response

Utilizing the six steps outlined in Chapter 10: Incident Handling Foundations from SANS Security Essentials with CISSP CBK[14], I was able to create an incident handling procedure. We now have a strong foundation to build upon where we had nothing before. The incident handling procedure will need to be expanded so that it is more comprehensive (i.e. disaster recovery plan). Adding more checklists would be helpful as they make responding to an incident more efficient and complete. Future reviews of the procedure will be done on a continuous basis as the hardware, software, and personnel changes in our company.

Conclusion

The information security cycle has allowed me to create a fairly comprehensive security plan in a relatively short period of time. Several of the tools were automated and several more will be automated in the future. The new tools have allowed me to be not only more efficient, but also more informed about the security of our network. The SANS Security Essentials course has not only helped me understand network security better, but I also learned about the Linux operating system. I have setup my computer to dual boot with Windows and Linux. Additional Linux computers will be added over the next few months. The Linux computers will be used primarily for network security tasks. Hopefully this paper has provided some ideas and assistance for other network administrators who manage a small Windows network.

References

1. Microsoft Windows Update.
   URL: http://windowsupdate.microsoft.com (17 April 2004).
2. Symantec Security Alert Emails.
   URL: http://nct.symantecstore.com/virusalert (17 April 2004).
3. Panda Software Virus Alert Newsletter.
   URL:
   http://www.pandasoftware.com/register.asp?CodigoProducto=99&TipoLead=2&TipoUsuario=2&Tipo=1&Ref=ww-SUSCRIP&Idioma=2&Country=us&sec=about&Lst_6=true&Lst_5=false&Lst_7=false&Lst_8=false (17 April 2004).
4. Cisco Security Notices.
   URL: http://www.cisco.com/warp/public/707/sec_incident_response.shtml#questions (20 April 2004).
5. Microsoft Security Notification Service.
   URL: http://register.microsoft.com/subscription/subscribeme.asp?ID=135 (17 April 2004).
6. US-CERT Technical Cyber Security Alert.
   URL: http://www.us-cert.gov/cas/techalerts/index.html (17 April 2004).
7. SANS Portal.
   URL: http://portal.sans.org/ (20 April 2004).
8. Microsoft Baseline Security Analyzer (MBSA).
   URL: http://www.microsoft.com/technet/security/tools/mbsahome.mspx (18 April 2004).
9. "Scripting with the Microsoft Baseline Security Analyzer V1.2." 17 March 2004.
   URL: http://www.microsoft.com/technet/security/tools/mbsascript.mspx (22 April 2004).
10. Ethereal.
    URL: http://www.ethereal.com/ (19 April 2004).
11. EventID.
    URL: http://www.eventid.net (22 April 2004).
12. Google.
    URL: http://www.google.com (22 April 2004).
13. SoftPerfect Network Scanner.
    URL: http://www.softperfect.com/products/networkscanner/ (22 April 2004).
14. Cole, E., Fossen, J., Northcutt, S., & Pomeranz, H. SANS Security Essentials with CISSP CBK, Volume 1. The SANS Institute, 2003. 451 – 492.
15. Nessus.
    URL: http://www.nessus.org (23 April 2004).
16. Snort.
    URL: http://www.snort.org/ (23 April 2004).

Appendix A: MBSA Scripts

batchMbsa_Servers.bat

```
@echo off
REM    batchMbsa_Servers.bat
REM    Scan servers and create summary reports script batch file
REM    Copyright (c) 2004 Monique Rae Nelson
REM    Version 1.0 - April 22, 2004
REM
REM    You have a royalty-free right to use, modify, reproduce, or distribute
REM    this batch file in any way you find useful, provided that you agree
REM    that the copyright owner above has no warranty, obligations, or
REM    liability for such use.

REM Scan servers using MBSA
"C:\Program Files\Microsoft Baseline Security Analyzer\mbsacli.exe" /r 192.168.0.20-
192.168.0.40 /o %%c%% /nosum

REM Copy new scan files to Servers folder
copy "%userprofile%\SecurityScans\*.xml" "%userprofile%\SecurityScans\Servers\"

REM Delete previous scanned files
del "%userprofile%\SecurityScans\*.xml"

REM Generate critical Summary Report
cscript.exe /nologo rollup_servers.js -c 115 124 126 127 174 179 212 302 415 417 418
>criticalPatch_Servers_SummaryReport.xml

REM Generate misconfiguration summary report
cscript.exe /nologo rollup_clients.js -c 102 104 106 107 110 117 118 122 201 203 204
205 206 207 213 215 216 217 218 301 307 308 309 311 315 316
>NonCriticalPatch_Clients_SummaryReport.xml
```

batchMbsa_Clients.bat

```
@echo off
REM    batchMbsa_Clients.bat
REM    Scan client computers and create summary reports script batch file
REM    Copyright (c) 2004 Monique Rae Nelson
REM    Version 1.0 - April 22, 2004
REM
REM    You have a royalty-free right to use, modify, reproduce, or distribute
REM    this batch file in any way you find useful, provided that you agree
REM    that the copyright owner above has no warranty, obligations, or
REM    liability for such use.
```

```
REM Scan client computers using MBSA
"C:\Program Files\Microsoft Baseline Security Analyzer\mbsacli.exe" /r 192.168.0.51-
192.168.0.254 /o %%c%% /nosum

REM Copy new scan files to Clients folder
copy "%userprofile%\SecurityScans\*.xml" "%userprofile%\SecurityScans\Clients\"

REM Delete previous scanned files
del "%userprofile%\SecurityScans\*.xml"

REM Generate critical Summary Report
cscript.exe /nologo rollup_clients.js -c 115 124 126 174 179 212 302 415 417 418
>criticalPatch_Clients_SummaryReport.xml

REM Generate misconfiguration summary report
cscript.exe /nologo rollup_clients.js -c 102 104 106 107 110 117 118 122 201 203 204
205 206 207 213 215 216 217 218 301 307 308 309 311 315 316
>NonCriticalPatch_Clients_SummaryReport.xml
```

Appendix B: snapshot.vbs

```
‘=============================================
‘   snapshot.vbs
‘   Create server snapshot vbscript file
'   Use command line to launch, first var = computer name
'   Note: Use Windows Task Scheduler to schedule the script execution
‘   Copyright (c) 2004 Monique Rae Nelson
‘   Version 1.0 - April 22, 2004
‘
‘   You have a royalty-free right to use, modify, reproduce, or distribute
‘   this batch file in any way you find useful, provided that you agree
‘   that the copyright owner above has no warranty, obligations, or
‘   liability for such use.
‘=============================================

' Snapshot.vbs
' Description: create a snapshot file
' Requires: Microsoft Windows Active Directory domain, target computer must have
Windows Management Instrumentation (WMI),
'          Microsoft Windows 2000 or higher
' How To: use command line to launch, first var = computer name
' Example of command line: wscript "C:\Scripts\snapshot.vbs" "ServerA"
' Note: Use Windows Task Scheduler to schedule this file to be executed on a timed
basis.
'
============================================================
========================

' Set directory location for snapshot storage
strSnapshotDirectory = "c:\snapshots\"

' Server/PC Name
Dim ArgObj, strComputer
Set ArgObj = WScript.Arguments
If ArgObj.Count > 0 then
        strComputer = ArgObj(0)
Else
        wscript.echo "ERROR: You must enter the computer name as a command line
argument."
        wscript.quit
End if

strComputer = UCASE(strComputer)

'Clear object out of memory
```

```
set ArgObj = Nothing

' Date
dateToday = UCASE(month(now) & day(now) & year(now))

' Create new snapshot file
Dim filesys, testfile
Set filesys = CreateObject("Scripting.FileSystemObject")
Set snapshotFN= filesys.CreateTextFile(strSnapshotDirectory & "snapshot_" &
strComputer & "_" & dateToday & ".txt", True)
snapshotFN.WriteBlankLines(1)

' Server/PC Name
snapshotFN.WriteLine "Computer: " & strComputer

' Date/Time
snapshotFN.WriteLine "Date and Time: " & UCASE(now)

' Script name
snapshotFN.WriteLine "Script: SNAPSHOT.VBS"

' Username of person running script
Set objNetwork = CreateObject("Wscript.Network")
snapshotFN.WriteLine "User Name: " & UCASE(objNetwork.UserName)

' Domain of person running script
snapshotFN.WriteLine "Domain Name: " & UCASE(objNetwork.UserDomain)

snapshotFN.WriteBlankLines(2)

On Error Resume Next
Set objWMIService = GetObject("winmgmts:" & "{impersonationLevel=impersonate}!\\"
& strComputer & "\root\cimv2")

snapshotFN.WriteLine "--------------------------------- "
snapshotFN.WriteLine " OS VERSION and SERVICE PACK "
snapshotFN.WriteLine "--------------------------------- "
snapshotFN.WriteBlankLines(1)

Set colComputer = objWMIService.ExecQuery ("Select * from
Win32_ComputerSystem")
For Each objComputer in colComputer

        Set colOperatingSystems = objWMIService.ExecQuery ("Select * from
Win32_OperatingSystem")
        For Each objOperatingSystem in colOperatingSystems
```

```
                snapshotFN.WriteLine objOperatingSystem.Caption & " (" &
objOperatingSystem.Version & ") SP " & objOperatingSystem.ServicePackMajorVersion
& "." & objOperatingSystem.ServicePackMinorVersion
        Next

Next

snapshotFN.WriteBlankLines(2)
snapshotFN.WriteLine "---------------------------------- "
snapshotFN.WriteLine " LOCAL USER ACCOUNT INFORMATION "
snapshotFN.WriteLine "---------------------------------- "
snapshotFN.WriteBlankLines(1)

Set colItems = objWMIService.ExecQuery("Select * from Win32_Account Where
LocalAccount = True")
For Each objItem in colItems
   snapshotFN.WriteLine "Name: " & objItem.Name
   snapshotFN.WriteLine "Description: " & objItem.Description
   snapshotFN.WriteLine "Status: " & objItem.Status
   snapshotFN.WriteLine "SID: " & objItem.SID
   snapshotFN.WriteLine "SID Type: " & objItem.SIDType
   snapshotFN.WriteBlankLines(1)
Next

snapshotFN.WriteBlankLines(2)
snapshotFN.WriteLine "---------------------------------- "
snapshotFN.WriteLine " LOCAL GROUPS AND MEMBERSHIPS "
snapshotFN.WriteLine "---------------------------------- "
snapshotFN.WriteBlankLines(1)

Set colGroups = GetObject("WinNT://" & strComputer & "")
colGroups.Filter = Array("group")
For Each objGroup In colGroups
   snapshotFN.WriteLine objGroup.Name
   For Each objUser in objGroup.Members
      snapshotFN.WriteLine vbTab & objUser.Name
   Next
   snapshotFN.WriteBlankLines(1)
Next

snapshotFN.WriteBlankLines(2)
snapshotFN.WriteLine "---------------------------------- "
snapshotFN.WriteLine " LOCAL SHARES "
snapshotFN.WriteLine "---------------------------------- "
snapshotFN.WriteBlankLines(1)
```

```
' Lists all the shared folders on a computer
Set colShares = objWMIService.ExecQuery("Select * from Win32_Share")
For each objShare in colShares
   snapshotFN.WriteLine "Name: " & vbTab & vbTab & objShare.Name
   snapshotFN.WriteLine "Description: " & vbTab & objShare.Description
   snapshotFN.WriteLine "Path: " & vbTab & vbTab & objShare.Path
   snapshotFN.WriteLine "Status: " & vbTab & vbTab & objShare.Status
   snapshotFN.WriteBlankLines(1)
Next

snapshotFN.WriteBlankLines(2)
snapshotFN.WriteLine "--------------------------------- "
snapshotFN.WriteLine " PROCESSES (INCLUDING PROPERTIES) "
snapshotFN.WriteLine "--------------------------------- "
snapshotFN.WriteBlankLines(1)

Set colProcessItems = objWMIService.ExecQuery("Select * from Win32_Process")
For Each objProcessItem in colProcessItems
   snapshotFN.WriteLine "Name: " & objProcessItem.Name
   snapshotFN.WriteLine "ExecutablePath: " & objProcessItem.ExecutablePath
   snapshotFN.WriteLine "ParentProcessId: " & objProcessItem.ParentProcessId
   snapshotFN.WriteLine "Priority: " & objProcessItem.Priority
   'snapshotFN.WriteLine "ProcessId: " & objProcessItem.ProcessId
   snapshotFN.WriteBlankLines(1)
Next

snapshotFN.WriteBlankLines(2)
snapshotFN.WriteLine "--------------------------------- "
snapshotFN.WriteLine " SERVICES "
snapshotFN.WriteLine "--------------------------------- "
snapshotFN.WriteBlankLines(1)

Set colServiceItems = objWMIService.ExecQuery("Select * from Win32_Service")
For Each objServiceItem in colServiceItems
   snapshotFN.WriteLine "Name: " & objServiceItem.Name
   snapshotFN.WriteLine "DisplayName: " & objServiceItem.DisplayName
   snapshotFN.WriteLine "Description: " & objServiceItem.Description
   snapshotFN.WriteLine "PathName: " & objServiceItem.PathName
   snapshotFN.WriteLine "Started: " & objServiceItem.Started
   snapshotFN.WriteLine "StartMode: " & objServiceItem.StartMode
   snapshotFN.WriteLine "State: " & objServiceItem.State
   snapshotFN.WriteLine "Status: " & objServiceItem.Status
   snapshotFN.WriteBlankLines(1)
Next

snapshotFN.WriteBlankLines(2)
```

```
snapshotFN.WriteLine "--------------------------------- "
snapshotFN.WriteLine " NETWORKING CONFIGURATION "
snapshotFN.WriteLine "--------------------------------- "
snapshotFN.WriteBlankLines(1)

Set colNetworkItems = objWMIService.ExecQuery("Select * from
Win32_NetworkAdapterConfiguration",,48)
For Each objNetworkItem in colNetworkItems
        snapshotFN.WriteLine "Description: " & objNetworkItem.Description
        snapshotFN.WriteLine "Default IP Gateway: " &
objNetworkItem.DefaultIPGateway
        snapshotFN.WriteLine "DHCP Enabled: " & objNetworkItem.DHCPEnabled
        snapshotFN.WriteLine "DHCP Lease Expires: " &
objNetworkItem.DHCPLeaseExpires
        snapshotFN.WriteLine "DHCP Lease Obtained: " &
objNetworkItem.DHCPLeaseObtained
        snapshotFN.WriteLine "DHCP Server: " & objNetworkItem.DHCPServer
        snapshotFN.WriteLine "DNS Domain: " & objNetworkItem.DNSDomain
        snapshotFN.WriteLine "DNS Enabled For WINS Resolution: " &
objNetworkItem.DNSEnabledForWINSResolution
        snapshotFN.WriteLine "DNS Host Name: " & objNetworkItem.DNSHostName
        snapshotFN.WriteLine "DNS Server Search Order: " &
objNetworkItem.DNSServerSearchOrder
        snapshotFN.WriteLine "IP Address: " & objNetworkItem.IPAddress
        snapshotFN.WriteLine "IP Enabled: " & objNetworkItem.IPEnabled
        snapshotFN.WriteLine "IP Filter Security Enabled: " &
objNetworkItem.IPFilterSecurityEnabled
        snapshotFN.WriteLine "IPSec Permit IP Protocols: " &
objNetworkItem.IPSecPermitIPProtocols
        snapshotFN.WriteLine "IPSec Permit TCP Ports: " &
objNetworkItem.IPSecPermitTCPPorts
        snapshotFN.WriteLine "IPSec Permit UDP Ports: " &
objNetworkItem.IPSecPermitUDPPorts
        snapshotFN.WriteLine "IP Subnet: " & objNetworkItem.IPSubnet
        snapshotFN.WriteLine "MAC Address: " & objNetworkItem.MACAddress
        snapshotFN.WriteLine "TCPIP Netbios Options: " &
objNetworkItem.TcpipNetbiosOptions
        snapshotFN.WriteLine "WINS Enable LMHosts Lookup: " &
objNetworkItem.WINSEnableLMHostsLookup
        snapshotFN.WriteLine "WINS Host Lookup File: " &
objNetworkItem.WINSHostLookupFile
        snapshotFN.WriteLine "WINS Primary Server: " &
objNetworkItem.WINSPrimaryServer
        snapshotFN.WriteLine "WINS Secondary Server: " &
objNetworkItem.WINSSecondaryServer
        snapshotFN.WriteBlankLines(1)
```

```
Next

snapshotFN.WriteBlankLines(2)
snapshotFN.WriteLine "--------------------------------- "
snapshotFN.WriteLine " ENVIRONMENTAL VARIABLES "
snapshotFN.WriteLine "--------------------------------- "
snapshotFN.WriteBlankLines(1)

Set colItems = objWMIService.ExecQuery("Select * from Win32_Environment")
For Each objItem in colItems
   snapshotFN.WriteLine "Description: " & objItem.Description
   snapshotFN.WriteLine "Name: " & objItem.Name
   snapshotFN.WriteLine "System Variable: " & objItem.SystemVariable
   snapshotFN.WriteLine "User Name: " & objItem.UserName
   snapshotFN.WriteLine "Variable Value: " & objItem.VariableValue
   snapshotFN.WriteBlankLines(1)
Next

snapshotFN.WriteBlankLines(2)
snapshotFN.WriteLine "--------------------------------- "
snapshotFN.WriteLine "       STARTUP COMMANDS "
snapshotFN.WriteLine " (STARTUP FOLDER & REGISTRY)"
snapshotFN.WriteLine "--------------------------------- "
snapshotFN.WriteBlankLines(1)

Set colStartupCommands = objWMIService.ExecQuery ("Select * from
Win32_StartupCommand")
For Each objStartupCommand in colStartupCommands
   snapshotFN.WriteLine "Command: " & objStartupCommand.Command
   snapshotFN.WriteLine "Description: " & objStartupCommand.Description
   snapshotFN.WriteLine "Location: " & objStartupCommand.Location
   snapshotFN.WriteLine "Name: " & objStartupCommand.Name
   snapshotFN.WriteLine "User: " & objStartupCommand.User
        snapshotFN.WriteBlankLines(1)
Next

snapshotFN.WriteBlankLines(2)
snapshotFN.WriteLine "--------------------------------- "
snapshotFN.WriteLine "              C: FILES "
snapshotFN.WriteLine " (SIZE, LAST MOD. DATE, HIDDEN ATTRIB.) "
snapshotFN.WriteLine "--------------------------------- "
snapshotFN.WriteBlankLines(1)

Set colFiles = objWMIService.ExecQuery("Select * from CIM_DataFile where
Drive='c:'")
For Each objFile in colFiles
```

```
        strDateTimeWritten = objFile.LastModified
        strYear = left(strDateTimeWritten, 4)
        strMonth = mid(strDateTimeWritten, 3,2)
        strDay = mid(strDateTimeWritten, 7,2)
        strHour = mid(strDateTimeWritten, 9,2)
        strMin = mid(strDateTimeWritten, 11,2)
        strSec = mid(strDateTimeWritten, 13,2)
        dateWritten = cDate(strMonth & "/" & strDay & "/" & strYear)
        timeWritten = cDate(strHour & ":" & strMin & ":" & strSec)
        snapshotFN.WriteLine objFile.Name & " -- " & objFile.FileSize & " bytes -- " &
dateWritten & " " & timeWritten & " -- " & objFile.Hidden
Next

snapshotFN.WriteBlankLines(3)

' Date/Time
snapshotFN.WriteLine "Snapshot Completed: " & UCASE(now)

snapshotFN.WriteBlankLines(5)

' Close new snapshot file
snapshotFN.Close

Wscript.quit
```

Appendix C: evLogToDb.vbs

```
‘==========================================
‘   evLogToDb.vbs
‘   Copy local event logs to Access database vbscript file
‘   Copyright (c) 2004 Monique Rae Nelson
‘   Version 1.0 - April 22, 2004
‘
‘   You have a royalty-free right to use, modify, reproduce, or distribute
‘   this batch file in any way you find useful, provided that you agree
‘   that the copyright owner above has no warranty, obligations, or
‘   liability for such use.
‘==========================================

' Copy records from local event logs to an Access Database using an ODBC connection
Set objConn = CreateObject("ADODB.Connection")
Set objRS = CreateObject("ADODB.Recordset")
objConn.Open "DSN=EventLogs;"
objRS.CursorLocation = 3
objRS.Open "SELECT * FROM EventTable" , objConn, 3, 3
strComputer = "."
Set objWMIService = GetObject("winmgmts:" _
   & "{impersonationLevel=impersonate}!\\" & strComputer & "\root\cimv2")
objWMIService.Security_.Privileges.AddAsString "SeSecurityPrivilege", True

On Error Resume Next

Set colRetrievedEvents = objWMIService.ExecQuery _
   ("Select * from Win32_NTLogEvent")
For Each objEvent in colRetrievedEvents
   objRS.AddNew
   objRS("RecordNumber") = objEvent.RecordNumber
        ' change date and time format to standardized format acceptable to Access
        strDateTimeWritten = objEvent.TimeWritten
        strYear = left(strDateTimeWritten, 4)
        strMonth = mid(strDateTimeWritten, 3,2)
        strDay = mid(strDateTimeWritten, 7,2)
        strHour = mid(strDateTimeWritten, 9,2)
        strMin = mid(strDateTimeWritten, 11,2)
        strSec = mid(strDateTimeWritten, 13,2)
        dateWritten = cDate(strMonth & "/" & strDay & "/" & strYear)
        timeWritten = cDate(strHour & ":" & strMin & ":" & strSec)
        objRS("Date") = dateWritten
        objRS("Time") = timeWritten
   objRS("Type") = objEvent.Type
   objRS("SourceName") = objEvent.SourceName
```

```
    objRS("Message") = objEvent.Message
    objRS("EventCode") = objEvent.EventCode
    objRS("ComputerName") = objEvent.ComputerName
    objRS("User") = objEvent.User
    objRS.Update
Next
objRS.Close
objConn.Close

' Clear Event Log
Set objWMIService =
GetObject("winmgmts:{impersonationLevel=impersonate,(Backup)}!\\" & _
       strComputer & "\root\cimv2")
objWMIService.Security_.Privileges.AddAsString "SeSecurityPrivilege", True
Set colLogFiles = objWMIService.ExecQuery _
    ("Select * from Win32_NTEventLogFile")
For Each objLogfile in colLogFiles
          objLogFile.ClearEventLog()
Next

wscript.quit
```

Appendix D: run_networkMapper.bat

```
@echo off
REM ============================================
REM    run_networkMapper.bat
REM    Create network map script batch file
REM    Copyright (c) 2004 Monique Rae Nelson
REM    Version 1.0 - April 22, 2004
REM
REM    You have a royalty-free right to use, modify, reproduce, or distribute
REM    this batch file in any way you find useful, provided that you agree
REM    that the copyright owner above has no warranty, obligations, or
REM    liability for such use.
REM ============================================

REM Delete old scan
del "C:\Tools\networkScanner\scan_old.txt"

REM Rename new scan to old
ren scan_new.txt scan_old.txt

REM Create new scan
 "C:\Tools\networkScanner\netscan.exe" /auto:scan_new.txt /hide

REM Compare new scan results to old scan results and create comparison file with
differences
fc.exe c:\tools\networkScanner\scan_new.txt c:\tools\networkScanner\scan_old.txt >
c:\tools\networkScanner\scan_compare.txt
```