



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

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

Interested in learning more?

Check out the list of upcoming events offering
"Network Monitoring and Threat Detection In-Depth (Security 503)"
at <http://www.giac.org/registration/gcia>

Capitol SANS Conference

GIAC Level 2 Intrusion Detection

Practical for

John Cusick

February 20, 2001

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

<u>Acknowledgements</u>	3
<u>Assignment 1 – Four Detects</u>	4
<u>Detect 1</u>	5
<u>1. Source of trace:</u>	5
<u>2. Detect generated by:</u>	5
<u>3. Probability source address was spoofed:</u>	7
<u>4. Description of attack:</u>	8
<u>5. Attack mechanism:</u>	8
<u>6. Correlations:</u>	8
<u>7. Evidence of active targeting:</u>	8
<u>8. Severity:</u>	9
<u>9. Defensive recommendation:</u>	9
<u>10. Test question:</u>	9
<u>Detect 2</u>	10
<u>1. Source of trace:</u>	10
<u>2. Detect generated by:</u>	10
<u>3. Probability source address was spoofed:</u>	11
<u>4. Description of attack:</u>	11
<u>5. Attack mechanism:</u>	11
<u>6. Correlations:</u>	12
<u>7. Evidence of active targeting:</u>	12
<u>8. Severity:</u>	12
<u>9. Defensive recommendation:</u>	12
<u>10. Test question:</u>	12
<u>Detect 3</u>	13
<u>1. Source of trace:</u>	13
<u>2. Detect generated by:</u>	13
<u>3. Probability source address was spoofed:</u>	13
<u>4. Description of Attack:</u>	14
<u>5. Attack mechanism:</u>	14
<u>6. Correlations:</u>	14
<u>7. Evidence of active targeting:</u>	14
<u>8. Severity:</u>	15
<u>9. Defensive recommendation:</u>	15
<u>10. Test question:</u>	15
<u>Detect 4:</u>	16
<u>1. Source of trace:</u>	16
<u>2. Detect generated by:</u>	16
<u>3. Probability source address was spoofed:</u>	16
<u>4. Description of attack:</u>	17
<u>5. Attack mechanism:</u>	17
<u>6. Correlations:</u>	17
<u>7. Evidence of active targeting:</u>	17
<u>8. Severity:</u>	17

<u>9. Defensive recommendation:</u>	17
<u>10. Test question:</u>	18
<u>Assignment 2 - "Analyze This"</u>	19
<u>Introduction</u>	19
<u>Most Active Hosts</u>	21
<u>Alert Definitions and Potential Severity Assessments</u>	23
<u>SUN RPC High Port Access!</u>	24
<u>TCP SMTP Source Port Traffic</u>	25
<u>Back Orifice</u>	26
<u>WinGate 1080 Attempt</u>	26
<u>SITE EXEC - Possible wu-ftpd exploit - GIAC000623</u>	26
<u>External RPC Call</u>	27
<u>Watchlist 000220 IL-ISDNNET-990517</u>	27
<u>SMB Name Wildcard</u>	27
<u>Attempted Sun RPC high port access</u>	28
<u>Watchlist 000222 NET-NCFC</u>	28
<u>spp_portskans</u>	29
<u>Tiny Fragments - Possible Hostile Activity</u>	29
<u>Probable NMAP Fingerprint Attempt</u>	29
<u>Queso Fingerprint</u>	30
<u>Null Scan!</u>	30
<u>NMAP TCP Ping!</u>	30
<u>SYN-FIN scan</u>	31
<u>OOS Alerts</u>	31
<u>Connect to 515 from Inside</u>	31
<u>Broadcast ping to subnet 70</u>	32
<u>Happy 99 Virus</u>	32
<u>SNMP Public Access</u>	32
<u>Analysis of Potentially More Severe Alerts</u>	32
<u>SUN RPC High Port Access!</u>	35
<u>TCP SMTP Source Port Traffic</u>	37
<u>Back Orifice</u>	38
<u>Source IP</u>	38
<u>Destination IP</u>	38
<u>WinGate 1080 Attempt</u>	39
<u>SITE EXEC - Possible wu-ftpd exploit - GIAC000623</u>	41
<u>External RPC Call</u>	42
<u>Watchlist 000220 IL-ISDNNET-990517</u>	43
<u>SMB Name Wildcard</u>	44
<u>Attempted Sun RPC high port access</u>	45
<u>Watchlist 000222 NET-NCFC</u>	46
<u>Conclusion</u>	47
<u>Assignment 3 - Analysis process</u>	48

References

56

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

Acknowledgements

I would like to thank a number of people because, without them, I wouldn't likely have undertaken this endeavor, and, had I done so, certainly wouldn't have completed it!

First, my thanks to Stephen Northcutt for his undying enthusiasm about this intrusion detection business and his ability put together a team to teach others the nuts and bolts of it. Without him we would not be where we are today.

Second, I would like to thank all the other analysts for the effort and information they put into their practicals. I gleaned insights and ideas from many of them that have led me to better understand the data I had before me.

Finally, but most of all, I would like to thank my wife and "partner for life," Christi. Without her encouragement, support and assistance, I simply wouldn't be doing this!

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

Assignment 1 – Four Detects

Each of the following four detects was taken from my home network connected to the Internet via an @home cable modem. The detects were generated by Snort running on a Linux system.

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

Detect 1

```
[**]IDS7 - MISC-Source Port Traffic 53 TCP [**]  
02/07-14:55:00.109665 209.4.187.39:53->MY.NET.209.56:53  
TCP TTL:239 TOS:0x0 ID:59556 IpLen:20 DgmLen:40  
*****S* Seq: 0x406C3E33 Ack: 0x207954CF Win: 0x28 TcpLen: 20
```

1. Source of trace:

Local network

2. Detect generated by:

Snort (v 1.7) running on Linux kernel 2.2.17 in full alert mode:

```
snort -i eth1 -d -D -l /var/log/snort -c /etc/snortrules.conf  
  
-i = sniff on ethernet 1 (external interface)  
-d = dump application layer  
-D = run in daemon mode  
-l = log packets to this directory  
-c = use this rules file
```

Ruleset: combination of vision.conf (Monday, Feb. 5) and snortfull.conf (Snort 1.7.0 Ruleset 01/25/2001)

Filter that triggered alert:

```
alert TCP $EXTERNAL 53 -> $INTERNAL :1023 (msg:"IDS7 - MISC-Source Port Traffic  
53 TCP"; flags: S; )
```

Snort alerts use the following log file format:

```
[**]IDS7 - MISC-Source Port Traffic 53 TCP[**]  
02/07-14:55:00.109665 209.4.187.39:53-> MY.NET.209.56:53  
TCP TTL:239 TOS:0x0 ID:59556 IpLen:20 DgmLen:40  
*****S* Seq: 0x406C3E33 Ack: 0x207954CF Win: 0x28 TcpLen: 20
```

→ **[**]IDS7 - MISC-Source Port Traffic 53 TCP[**]** is the Snort signature.

```
[**]IDS7 - MISC-Source Port Traffic 53 TCP [**]  
02/07-14:55:00.109665 209.4.187.39:53->MY.NET.209.56:53  
TCP TTL:239 TOS:0x0 ID:59556 IpLen:20 DgmLen:40  
*****S* Seq: 0x406C3E33 Ack: 0x207954CF Win: 0x28 TcpLen: 20
```

→ **02/07-14:55:00.109665** is the system date (mo/date) and time (hour:minute:second:millisecond)


```
[**]IDS7 - MISC-Source Port Traffic 53 TCP [**]  
02/07-14:55:00.109665 209.4.187.39:53->MY.NET.209.56:53  
TCP TTL:239 TOS:0x0 ID:59556 IpLen:20 DgmLen:40  
*****S* Seq: 0x406C3E33 Ack: 0x207954CF Win: 0x28 TcpLen: 20
```

→ **209.4.187.39:53** is the source host address and IP port number (53)

```
[**]IDS7 - MISC-Source Port Traffic 53 TCP [**]  
02/07-14:55:00.109665 209.4.187.39:53->MY.NET.209.56:53  
TCP TTL:239 TOS:0x0 ID:59556 IpLen:20 DgmLen:40  
*****S* Seq: 0x406C3E33 Ack: 0x207954CF Win: 0x28 TcpLen: 20
```

→ **->** indicates the direction the packet traveled

```
[**]IDS7 - MISC-Source Port Traffic 53 TCP [**]  
02/07-14:55:00.109665 209.4.187.39:53->MY.NET.209.56:53  
TCP TTL:239 TOS:0x0 ID:59556 IpLen:20 DgmLen:40  
*****S* Seq: 0x406C3E33 Ack: 0x207954CF Win: 0x28 TcpLen: 20
```

→ **MY.NET.209.56:53** is the destination host address and IP port number (53)

```
[**]IDS7 - MISC-Source Port Traffic 53 TCP [**]  
02/07-14:55:00.109665 209.4.187.39:53->MY.NET.209.56:53  
TCP TTL:239 TOS:0x0 ID:59556 IpLen:20 DgmLen:40  
*****S* Seq: 0x406C3E33 Ack: 0x207954CF Win: 0x28 TcpLen: 20
```

→ **TCP TTL:239** indicates the protocol and datagram's time to live (TTL) value

```
[**]IDS7 - MISC-Source Port Traffic 53 TCP [**]  
02/07-14:55:00.109665 209.4.187.39:53->MY.NET.209.56:53  
TCP TTL:239 TOS:0x0 ID:59556 IpLen:20 DgmLen:40  
*****S* Seq: 0x406C3E33 Ack: 0x207954CF Win: 0x28 TcpLen: 20
```

→ **TOS:0x0** indicates the type of service

```
[**]IDS7 - MISC-Source Port Traffic 53 TCP [**]  
02/07-14:55:00.109665 209.4.187.39:53->MY.NET.209.56:53  
TCP TTL:239 TOS:0x0 ID:59556 IpLen:20 DgmLen:40  
*****S* Seq: 0x406C3E33 Ack: 0x207954CF Win: 0x28 TcpLen: 20
```

→ **ID:59556** is the datagram identification number

```
[**]IDS7 - MISC-Source Port Traffic 53 TCP [**]02/07-14:55:00.109665  
209.4.187.39:53->MY.NET.209.56:53  
TCP TTL:239 TOS:0x0 ID:59556 IpLen:20 DgmLen:40  
*****S* Seq: 0x406C3E33 Ack: 0x207954CF Win: 0x28 TcpLen: 20
```

→ **IpLen:20** is the ip header length in bytes

```
[**]IDS7 - MISC-Source Port Traffic 53 TCP [**]  
02/07-14:55:00.109665 209.4.187.39:53->MY.NET.209.56:53  
TCP TTL:239 TOS:0x0 ID:59556 IpLen:20 DgmLen:40  
*****S* Seq: 0x406C3E33 Ack: 0x207954CF Win: 0x28 TcpLen: 20
```

→ **DgmLen:40** is the total IP datagram length in bytes

```
[**]IDS7 - MISC-Source Port Traffic 53 TCP [**]  
02/07-14:55:00.109665 209.4.187.39:53->MY.NET.209.56:53  
TCP TTL:239 TOS:0x0 ID:59556 IpLen:20 DgmLen:40  
*****S* Seq: 0x406C3E33 Ack: 0x207954CF Win: 0x28 TcpLen: 20
```

→ *******S*** indicates the TCP flags that are set, where

```
F = Fin  
S = Syn  
R = Reset  
P = Push  
A = Ack  
U = Urgent  
2 = Reserved bits 2  
1 = Reserved bits 1
```

```
[**]IDS7 - MISC-Source Port Traffic 53 TCP [**]  
02/07-14:55:00.109665 209.4.187.39:53->MY.NET.209.56:53  
TCP TTL:239 TOS:0x0 ID:59556 IpLen:20 DgmLen:40  
*****S* Seq: 0x406C3E33 Ack: 0x207954CF Win: 0x28 TcpLen: 20
```

→ **Seq: 0x406C3E33** is the source host's TCP sequence number (Hex)

```
[**]IDS7 - MISC-Source Port Traffic 53 TCP [**]  
02/07-14:55:00.109665 209.4.187.39:53->MY.NET.209.56:53  
TCP TTL:239 TOS:0x0 ID:59556 IpLen:20 DgmLen:40  
*****S* Seq: 0x406C3E33 Ack: 0x207954CF Win: 0x28 TcpLen: 20
```

→ **Ack: 0x207954CF** is the TCP acknowledgement number (Hex)

```
[**]IDS7 - MISC-Source Port Traffic 53 TCP [**]  
02/07-14:55:00.109665 209.4.187.39:53->MY.NET.209.56:53  
TCP TTL:239 TOS:0x0 ID:59556 IpLen:20 DgmLen:40  
*****S* Seq: 0x406C3E33 Ack: 0x207954CF Win: 0x28 TcpLen: 20
```

→ **Win: 0x28** is the TCP window size (Hex)

```
[**]IDS7 - MISC-Source Port Traffic 53 TCP [**]  
02/07-14:55:00.109665 209.4.187.39:53->MY.NET.209.56:53  
TCP TTL:239 TOS:0x0 ID:59556 IpLen:20 DgmLen:40*****S* Seq: 0x406C3E33 Ack:  
0x207954CF Win: 0x28 TcpLen: 20
```

→ **TcpLen: 20** is the TCP header length in bytes

3. Probability source address was spoofed:

It is likely the source address was spoofed. It would certainly have been possible to craft such a packet. For one, the high TTL value (239) makes the packet suspect.

Using [Andrew's Webserver at Triumph](#) and the [ARIN database](#) the source address was determined to be an ISDN connection with an ISP in Florida. Although attempts to contact the address were unsuccessful, the ARIN lookup does not suggest it is running DNS.

The snort portscan log indicated no portscan activity around the time of the alert.

Under normal circumstances, the Syn flag setting with source and destination ports 53 would indicate a zone transfer.

The active Snort ruleset includes the following rules:

```
alert TCP $EXTERNAL any -> $INTERNAL 53 (msg:"IDS212 - dns-zone-transfer";  
flags: A+; content: "|FC|"; offset: 13; )  
  
alert UDP $EXTERNAL any -> $INTERNAL 53 (msg:"IDS278 - named-probe-version";  
content: "|07|version|04|bind"; nocase; offset: 12; depth: 32; )
```

DNS was not running on the destination host and these two rules generated no detects making it unlikely a zone transfer was requested.

In fact, the destination host responded with a destination unreachable as this snort alert indicates:

```
[**]  
ICMP Destination Unreachable (Undefined Code!) [**]  
02/07-14:55:00.109807 MY.NET.209.56 -> 209.4.187.39  
ICMP TTL:255 TOS:0xC0 ID:81 IpLen:20 DgmLen:88  
Type:3 Code:3 DESTINATION UNREACHABLE: PORT UNREACHABLE
```

4. Description of attack:

The attacker is attempting to make a connection using source port 53 to destination port 53. If the connection had been established, a zone transfer or buffer overflow may have been attempted.

There are no CVE numbers for this attack.

5. Attack mechanism:

This attack is a stimulus in the form of an ICMP echo request packet, with a particular ID, seeking a response from friendly hosts.

By attempting to initiate a connection to port 53, the attack is attempting to exploit known vulnerabilities in DNS. It is, after all, among the top ten vulnerabilities referenced by Randy Marchany in his [Presentation on the Top Ten Vulnerabilities](#).

6. Correlations:

No correlations of this attack were evident on the local network. A review of a previous student's practical ([Jussi Kallio](#)) revealed a similar alert. This one differed in that it was preceded by NMAP pings.

A web search using Google identified [Neohapsis archives](#) that revealed this attack has been used with different destination ports in portscans, picked up by the Snort portscan preprocessor.

7. Evidence of active targeting:

This would seem to be a "wrong number." There is no evidence of reconnaissance to identify this host as a likely target. DNS is not running on the host.

8. Severity:

Applying the formula:

$(\text{System Criticality} + \text{Attack Lethality}) - (\text{System Countermeasures} + \text{Network Countermeasures}) = \text{Severity}$

$$(5 + 5) - (5 + 4) = 1$$

System criticality: **5** - system is a firewall

Attack lethality: **5** - goal is root access or DNS zone transfer

System countermeasures: **5** - current OS with patches applied

Network countermeasures: **4** - firewall (non stateful) blocking all packets or external origin

9. Defensive recommendation:

Defenses appear to have been adequate. Converting to a stateful firewall is recommended.

10. Test question:

Consider the following packet:

```
02/07-14:55:00.109665 209.4.187.39:53-> MY.NET.209.56:53
TCP TTL:239 TOS:0x0 ID:59556 IpLen:20 DgmLen:40
*****S* Seq: 0x406C3E33 Ack: 0x207954CF Win: 0x28 TcpLen: 20
```

Normally, a client query to a DNS server

- a) uses UDP, source port 53 and a destination port > 1023
- b) uses TCP, source port 53 and a destination port > 1023
- c) uses UDP, a source port > 1023 and destination port 53
- d) uses TCP, a source port > 1023 and destination port 53

Answer: c

Detect 2

```
[**] IDS193 - ddos-stacheldraht server-spoof [**]  
01/18-13:10:32.521402 64.229.236.188 -> LOCAL.NET.27.210  
ICMP TTL:240 TOS:0x0 ID:16641 IpLen:20 DgmLen:32 DF  
Type:8 Code:0 ID:666 Seq:1 ECHO
```

1. Source of trace:

Local network

2. Detect generated by:

The IDS, rule set and IP part of the log file format are explained above in Detect1.
Snort ICMP alerts differ from Snort TCP alerts in the following emphasized fields:

```
[**] IDS193 - ddos-stacheldraht server-spoof [**]  
01/18-13:10:32.521402 64.229.236.188 -> LOCAL.NET.27.210  
ICMP TTL:240 TOS:0x0 ID:16641 IpLen:20 DgmLen:32 DF  
Type:8 Code:0 ID:666 Seq:1 ECHO
```

These will now be explained individually:

```
[**] IDS193 - ddos-stacheldraht server-spoof [**]  
01/18-13:10:32.521402 64.229.236.188 -> LOCAL.NET.27.210  
ICMP TTL:240 TOS:0x0 ID:16641 IpLen:20 DgmLen:32 DF  
Type:8 Code:0 ID:666 Seq:1 ECHO
```

→ **DF** indicates the don't fragment bit is set.

```
[**] IDS193 - ddos-stacheldraht server-spoof [**]  
01/18-13:10:32.521402 64.229.236.188 -> LOCAL.NET.27.210  
ICMP TTL:240 TOS:0x0 ID:16641 IpLen:20 DgmLen:32 DF  
Type:8 Code:0 ID:666 Seq:1 ECHO
```

→ **Type:8** indicates the ICMP type (Echo request).

```
[**] IDS193 - ddos-stacheldraht server-spoof [**]  
01/18-13:10:32.521402 64.229.236.188 -> LOCAL.NET.27.210  
ICMP TTL:240 TOS:0x0 ID:16641 IpLen:20 DgmLen:32 DF  
Type:8 Code:0 ID:666 Seq:1 ECHO
```

→ **Code:0**

```
[**] IDS193 - ddos-stacheldraht server-spoof [**]  
01/18-13:10:32.521402 64.229.236.188 -> LOCAL.NET.27.210  
ICMP TTL:240 TOS:0x0 ID:16641 IpLen:20 DgmLen:32 DF  
Type:8 Code:0 ID:666 Seq:1 ECHO
```

→ **ID:666** indicates the ICMP identification.

```
[**] IDS193 - ddos-stacheldraht server-spoof [**]  
01/18-13:10:32.521402 64.229.236.188 -> LOCAL.NET.27.210  
ICMP TTL:240 TOS:0x0 ID:16641 IpLen:20 DgmLen:32 DF  
Type:8 Code:0 ID:666 Seq:1 ECHO
```

→ Seq:1

```
[**] IDS193 - ddos-stacheldraht server-spoof [**]  
01/18-13:10:32.521402 64.229.236.188 -> LOCAL.NET.27.210  
ICMP TTL:240 TOS:0x0 ID:16641 IpLen:20 DgmLen:32 DF  
Type:8 Code:0 ID:666 Seq:1 ECHO
```

→ ECHO

The specific Snort rule that caused this alarm is:

```
alert ICMP any any -> any any (msg:"IDS193 - ddos-stacheldraht server-spoof";  
itype: 8; icmp_id: 666; )
```

3. Probability source address was spoofed:

This source address was probably not spoofed. The source address is likely an agent querying a list of handlers to determine if the handlers are up. In this case, the source address could not be spoofed as it would need to receive the echo reply packets from the queried handlers.

Using [Andrew's Webserver at Triumph](#) and the [ARIN database](#), the source address appears to be a PPP connection assigned to an ISP in Quebec City, Canada.

There is a possibility the address was spoofed if this does not, in fact, involve stacheldraht agents and handlers communicating with one another. This would involve the use of crafted packets with a tool such as Icmpenum v 1.1 available from [Simple Nomad](#).

[Other tools](#) discussed in this paper enable replies to spoofed addresses to be picked up by promiscuous listeners.

4. Description of attack:

The “attacker,” in this case, probably an innocent victim, is sending ICMP echo requests to other hosts suspected to be participating in its attack network.

The attack is fully described by [David Dittrich](#) at the University of Washington.

It is a candidate for inclusion in the [CVE list](#).

5. Attack mechanism:

This attack is a stimulus in the form of an ICMP echo request packet, with a particular ID, seeking a response from friendly hosts. It is part of a denial of service “tribal” network.

6. Correlations:

Since the destination is not within my local network (it appears to be a neighbor on my ISP's cable subnet), I am unable to provide local correlations.

A web search offered some information by [Phil Wood](#) who indicates the alert is set off "by a Macintosh freeware package which has some relationship to Napster."

7. Evidence of active targeting:

This attack appears to be directed at specific hosts thought to be fellow handlers or agents.

8. Severity:

Since the destination is not on my local network, it is difficult to accurately apply the formula:

$(\text{System Criticality} + \text{Attack Lethality}) - (\text{System Countermeasures} + \text{Network Countermeasures}) = \text{Severity}$

$$(3+4) - (3+0) = 4$$

System criticality: **3** - system is unknown
Attack lethality: **4** - goal is Denial of Service
System countermeasures: **3** - unknown
Network countermeasures: **0** - likely non-existent because host is targeted

9. Defensive recommendation:

The destination host should be secured behind a firewall and then scanned for any trojans that may be on it.

10. Test question:

Consider the following alert:

```
[**] IDS193 - ddos-stacheldraht server-spoof [**]  
01/18-13:10:32.521402 64.229.236.188 -> LOCAL.NET.27.210  
ICMP TTL:240 TOS:0x0 ID:16641 IpLen:20 DgmLen:32 DF  
Type:8 Code:0 ID:666 Seq:1 ECHO
```

Which of the following best describes the packet that triggered this alert:

- a) It is a stimulus, in that it is an ICMP echo request
- b) It is a stimulus, in that it is an ICMP echo reply
- c) It is a response, in that it is an ICMP echo reply
- d) None of the above

Answer: a

Detect 3

```
[**]  
IDS183  DDoS - TFN client command LE [**]  
01/24-07:46:59.032641 198.133.219.25 -> MY.NET.209.56  
ICMP TTL:244 TOS:0x0 ID:826 IpLen:20 DgmLen:84  
Type:0 Code:0 ID:51201 Seq:0 ECHO REPLY
```

1. Source of trace:

Local network

2. Detect generated by:

The IDS, rule set and the log file format is explained above.

The following rule generated the alert:

```
alert ICMP $EXTERNAL any -> $INTERNAL any (msg:"IDS183 - DDoS - TFN client  
command LE"; itype: 0; icmp_id: 51201; icmp_seq: 0; )
```

3. Probability source address was spoofed:

Highly unlikely. While it could have been spoofed, an investigation of the source address found it to be a web page at Cisco:



4. Description of Attack:

Had this been an actual attack, the echo reply packet would have sent instructions to a Trinoo File Network (TFN) server. The echo reply would have been more likely to get through a firewall than an echo request would have and would not have generated an echo reply response on the destination host.

The TFN denial of service tool is described by [David Dittrich](#) at the University of Washington.

It is a candidate for inclusion in the [CVE list](#).

5. Attack mechanism:

Although it appears to be a response, this attack is a stimulus in the form of an ICMP echo reply packet, with a particular ID (51201). This ID corresponds to command value 456, which would spawn a shell on the destination host. Denial of Service commands could then be sent to the host.

6. Correlations:

Since it was a false alarm, there was really nothing to correlate. In this particular case, a review of the alert log documented that this alert was in fact a reply to an echo request sent from my network:

```
01/24-07:46:58.995742 MY.NET.209.56 -> 198.133.219.25
ICMP TTL:64 TOS:0x0 ID:826 IpLen:20 DgmLen:84
Type:8 Code:0 ID:51201 Seq:0 ECHO
```

The time on this stimulus packet (07:46:58) immediately precedes the alarmed detect packet above (07:46:59) and the ICMP sequence numbers are equivalent.

In fact, further review of the logs reveals another pair of ICMP packets immediately following this exchange:

```
01/24-07:46:59.994824 MY.NET.209.56 -> 198.133.219.25
ICMP TTL:64 TOS:0x0 ID:827 IpLen:20 DgmLen:84
Type:8 Code:0 ID:51201 Seq:1 ECHO

01/24-07:47:00.034165 198.133.219.25 -> MY.NET.209.56
ICMP TTL:244 TOS:0x0 ID:827 IpLen:20 DgmLen:84 Type:0 Code:0 ID:51201 Seq:1 ECHO
REPLY
```

Note the times, ICMP types and sequence numbers reflect this pattern. These weren't alarmed on because the ICMP sequence number was 1, instead of 0.

I did note a previous [GCIA reporting](#) of the alert also appeared to be a false alarm:

7. Evidence of active targeting:

None in this case. False alarm.

8. Severity:

Using the formula, and assuming it had been a real attack:

$(\text{System Criticality} + \text{Attack Lethality}) - (\text{System Countermeasures} + \text{Network Countermeasures}) = \text{Severity}$

$$(5 + 4) - (5 + 4) = 0$$

System criticality: **5** - system is a firewall
Attack lethality: **4** - goal is to use system in denial of service
System countermeasures: **5** - current OS with patches applied
Network countermeasures: **4** - firewall (non stateful) blocking all packets or external origin

9. Defensive recommendation:

The host should be secured behind a firewall, and the analyst should not be alarmed by false alarms!

10. Test question:

Consider the following packet sequence:

01/24-07:46:58.995742 MY.NET.209.56 -> 198.133.219.25
ICMP TTL:64 TOS:0x0 ID:826 IpLen:20 DgmLen:84
Type:8 Code:0 ID:51201 Seq:0 ECHO

01/24-07:46:59.032641 198.133.219.25 -> MY.NET.209.56
ICMP TTL:244 TOS:0x0 ID:826 IpLen:20 DgmLen:84
Type:0 Code:0 ID:51201 Seq:0 ECHO REPLY

01/24-07:46:59.994824 MY.NET.209.56 -> 198.133.219.25
ICMP TTL:64 TOS:0x0 ID:827 IpLen:20 DgmLen:84
Type:8 Code:0 ID:51201 Seq:1 ECHO

01/24-07:47:00.034165 198.133.219.25 -> MY.NET.209.56
ICMP TTL:244 TOS:0x0 ID:827 IpLen:20 DgmLen:84
Type:0 Code:0 ID:51201 Seq:1 ECHO REPLY

If you know these packets were exchanged during a session where MY.NET.209.56 was browsing a web page at 198.133.219.25, what is a likely explanation for them:

- a) MY.NET.209.56 is attempting to establish an FTP connection with 198.133.219.25.
- b) MY.NET.209.56 believes it has lost a connection with the host at 198.133.219.25 and is attempting to see if the host is still alive.
- c) 198.133.219.25 is initiating a transfer of data to MY.NET.209.56.
- d) 198.133.219.25 believes it has lost a connection with the host at MY.NET.209.56 and is attempting to see if that host is still alive.

Answer: b

Detect 4:

```
[**]  
IDS284 - MISC - Shellcode X86 Setgid0 [**]  
01/18-11:40:31.601973 128.208.34.102:554-> MY.NET.209.56:64924  
TCP TTL:121 TOS:0x0 ID:32772 IpLen:20 DgmLen:798 DF  
***AP*** Seq: 0x197679C2 Ack: 0x5A05C322 Win: 0x2238 TcpLen: 20
```

```
[**]  
IDS284 - MISC - Shellcode X86 Setgid0 [**]  
01/18-11:40:33.583511 128.208.34.102:554-> MY.NET.209.56:64986  
TCP TTL:121 TOS:0x0 ID:47373 IpLen:20 DgmLen:1500 DF  
***AP*** Seq: 0x1962174B Ack: 0x7F474EB6 Win: 0x1D1C TcpLen: 20
```

1. Source of trace:

Local network

2. Detect generated by:

The IDS, rule set and the log file format are explained above.

The particular rule that triggered the alarm is:

```
alert TCP $EXTERNAL any -> $INTERNAL any (msg:"IDS284 - MISC - Shellcode X86  
Setgid0"; flags: PA; content: "|B0B5 CD80|"; )
```

3. Probability source address was spoofed:

Highly unlikely. The exploit is designed for the attacker to gain root shell access to the victim, so the attacker would want maintain the channel of communication to exploit that access.

In this particular case, I attempted to determine who the host was by using the ARIN whois database. I found the address was assigned to a block of addresses at the University of Washington.

Reviewing the alert logs, I noticed this source address was also an alert destination (127 times)!

Each of the 127 alerts were as follows:

```
[**]  
ICMP Destination Unreachable (Undefined Code!) [**]  
01/16-19:33:21.947516 MY.NET.209.56 -> 128.208.34.102  
ICMP TTL:255 TOS:0xC0 ID:8268 IpLen:20 DgmLen:576  
Type:3 Code:3 DESTINATION UNREACHABLE: PORT UNREACHABLE
```

Each of these alerts was a reply to the following:

```
128.208.34.102:27288 -> MY.NET.209.56:6970  
UDP TTL:121 TOS:0x0 ID:64466 IpLen:20 DgmLen:791  
Len: 771
```

This alert is based upon the following rule:

```
alert ICMP any any -> any any (msg:"ICMP Destination Unreachable (Undefined Code!)" ; itype: 3; )
```

While various high ports appeared to be used by the outside host, my local host always used port 6970. I checked on what uses port 6970 and found it was RealAudio. I then checked the address of the NPR station I listen to from the University of Washington and found it was the destination host in this detect!

4. Description of attack:

The attack is designed to gain root access to a Unix system.

It is described at:

<http://whitehats.com/IDS/284>

There is no CVE on it.

5. Attack mechanism:

This attack sends data that represents the setgid (0) system call on the x86 platform. This system call enables root access to the system.

6. Correlations:

There were no correlations for this attack. Although I could not find any explicit correlations to the false alarm I experienced, [Max Vision's](#) description of the attack does note "there may be many cases of false alarms where binary data is transferred from outside the network."

This was quite clearly the case in this false alarm...the suspect packets contained real audio data being transmitted from the web site to the internal network.

7. Evidence of active targeting:

Had this not been a false alarm, there would likely be active targeting of a previously identified vulnerable host.

8. Severity:

Applying the formula:

$(\text{System Criticality} + \text{Attack Lethality}) - (\text{System Countermeasures} + \text{Network Countermeasures}) = \text{Severity}$

$$(5 + 5) - (5 + 4) = 1$$

System criticality:	5 - system is a firewall
Attack lethality:	5 - goal is root access
System countermeasures:	5 - current OS with patches applied
Network countermeasures:	4 - firewall (non stateful) blocking all packets or external origin

9. Defensive recommendation:

To defend against this attack, it is recommended that fully patched hosts be secured behind stateful firewalls, and that these hosts apply good password policy.

10. Test question:

Consider the following alarm:

```
[**]  
IDS284 - MISC - Shellcode X86 Setgid0 [**]  
01/18-11:40:31.601973 128.208.34.102:554-> MY.NET.209.56:64924  
TCP TTL:121 TOS:0x0 ID:32772 IpLen:20 DgmLen:798 DF  
***AP*** Seq: 0x197679C2 Ack: 0x5A05C322 Win: 0x2238 TcpLen: 20
```

What application generated this alarm:

- a) tcpdump
- b) BlackIce
- c) shadow
- d) Snort

Answer: d

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

Assignment 2 - "Analyze This"

Introduction

We have reviewed and analyzed the Snort IDS data you forwarded to us. Although you indicated the data spanned a period of one month, we found it actually spanned a four month period from August 17 through November 23, 2000. We understand data is not available for all dates because of power failures or a full disk.

The data we received may be divided into three types:

- Snort alert data, in the form of 54 "SnortA*.txt" files ("fast" alert format)
- Snort portscan data, in the form of 42 "SnortS*.txt" files (portscan log format)
- Snort alert data, in the form of 19 "OOSche*.txt" files (apparently generated from binary using readback mode with a snap length of 68)

Within each of the three data types, we found two separate files identical to each other that containing the same data for the same date. The redundant files were deleted. The remaining data sets are summarized in Table 1.

Date	Day of Week	Holiday	Alert Files	Port Scan Files	Out of Spec Packet Files
08/17/00	Tue				OOScheck.txt
08/18/00	No Files				
...					
09/25/00					
09/26/00	Sun		SnortA15.txt		
09/27/00	Mon		SnortA13.txt	SnortS14.txt	
09/28/00	Tue		SnortA12.txt	SnortS11.txt	
09/29/00	Wed		SnortA9.txt	SnortS10.txt	
09/30/00	Thurs				
10/01/00	Fri		SnortA8.txt	SnortS7.txt	OOSche6.txt
10/02/00	Sat		SnortA4.txt	SnortS5.txt	OOSche3.txt
10/03/00	Sun		SnortA1e.txt	SnortSCA.txt	OOSche2.txt
10/04/00	Mon		SnortA2.txt		OOSche29.txt
10/05/00	Tue		SnortA28.txt	SnortS27.txt	
10/06/00	Wed		SnortA26.txt		
10/07/00	Thurs		SnortA25.txt	SnortS20.txt	OOSche24.txt
10/08/00	Fri		SnortA22.txt	SnortS21.txt	
10/09/00	Sat	Columbus Day	SnortA19.txt	SnortS13.txt	
10/10/00	Sun		SnortA10.txt	SnortS8.txt	OOSche25.txt
10/11/00	Mon		SnortA23.txt	SnortS22.txt	
10/12/00	Tue		SnortA20.txt	SnortS12.txt	
10/13/00	Wed		SnortA7.txt	SnortS6.txt	

10/14/00	Thurs		SnortA5.txt	SnortS4.txt	OOSche10.txt
10/15/00	Fri		SnortA11.txt	SnortS9.txt	
10/16/00	Sat		SnortA3.txt	SnortS2.txt	
10/17/00	Sun				
10/18/00	Mon		SnortA42.txt	SnortS41.txt	OOSche7.txt
10/19/00	Tue		SnortA40.txt	SnortS39.txt	
10/20/00	Wed		SnortA31.txt		
10/21/00	Thurs		SnortA33.txt	SnortS32.txt	
10/22/00	Fri		SnortA38.txt	SnortS37.txt	
10/23/00	Sat		SnortA35.txt	SnortS36.txt	OOSche34.txt
10/24/00	Sun		SnortA29.txt	SnortS30.txt	
10/25/00	Mon		SnortA24.txt	SnortS24.txt	
10/26/00	Tue		SnortA21.txt	SnortS15.txt	OOSche4.txt
10/27/00	Wed		SnortA27.txt		
10/28/00	Thurs		SnortA36.txt	SnortS35.txt	
10/29/00	Fri		SnortA39.txt	SnortS38.txt	
10/30/00	Sat		SnortA34.txt	SnortS33.txt	
10/31/00	Sun	Halloween	SnortA30.txt	SnortS31.txt	
11/01/00	Mon		SnortA6.txt	SnortS3.txt	
11/02/00	Tue			SnortS45.txt	
11/03/00	Wed		SnortA37.txt	SnortS34.txt	OOSche44.txt
11/04/00	Thurs		SnortA43.txt	SnortS42.txt	OOSche046.txt
11/05/00	Fri		SnortA41.txt		
11/06/00	Sat		SnortA44.txt		
11/07/00	Sun	Election Day	SnortA32.txt	SnortS16.txt	OOSche17.txt
11/08/00	Mon		SnortA53.txt		
11/09/00	Tue		SnortA52.txt		
11/10/00	Wed		SnortA46.txt	SnortS47.txt	OOSche45.txt
11/11/00	Thurs	Veterans Day	SnortA48.txt	SnortS49.txt	OOSche50.txt
11/12/00	Fri		SnortA51.txt		
11/13/00	Sat		SnortA49.txt	SnortS48.txt	
11/14/00	Sun		SnortA45.txt	SnortS17.txt	
11/15/00	Mon				
11/16/00	Tue		SnortA59.txt		
11/17/00	Wed		SnortA55.txt	SnortS56.txt	
11/18/00	Thurs			SnortS58.txt	
11/19/00	Fri		SnortA57.txt		
11/20/00	Sat		SnortA54.txt		
11/21/00	Sun		SnortA50.txt		
11/22/00	Mon		SnortA47.txt		OOSche20.txt
11/23/00	Tue	Thanksgiving Day		SnortS18.txt	OOSche19.txt

Table 1 – All Snort Files

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

For purposes of this analysis the “SnortA” data will be referred to as “alerts”, the “SnortS” data will be referred to as “scans” and the “OOSche” data will be referred to as “OOS”.

To facilitate the analysis, your designation of “MY.NET” was changed to “10.1”. Thus, our analyses will refer to your IP addresses as if they were in the “10.1” rather than “My.NET” network space.

Most Active Hosts

To obtain a sense which source and destination hosts were causing the most Snort detects, the data was sorted by frequency of activity. Table 2 summarizes alert data for the 10 most active destination hosts, while Table 3 does the same for the 10 most active source hosts.

Total Alerts: 150328		Total No. Days: 53	
Destination IP	# Alerts	% Month	
10.1.6.7	5767	3.84%	
10.1.223.98	3939	2.62%	
10.1.214.170	1367	0.91%	
10.1.211.146	4813	3.20%	
10.1.206.90	3915	2.60%	
10.1.203.142	1639	1.09%	
10.1.218.142	1459	0.97%	
10.1.202.22	951	0.63%	
10.1.201.174	800	0.53%	
10.1.100.230	797	0.53%	

Table 2 - Most Frequent Destination Hosts – Alerts

Source IP	Name	# Alerts	% Month
160.78.49.191	ema.chim.unipr.it	7199	4.8%
208.61.4.207	adsl-61-4-207.mia.bellsouth.net	6635	4.4%
159.226.45.3	aphy.iphy.ac.cn	6160	4.1%
212.179.95.5	cable-95005.bezeqint.net	5683	3.8%
209.92.40.32	dslcv1-32.fast.net	4967	3.3%
212.179.79.2	none (ISDN Net Ltd. - Israel)	3950	2.6%
212.179.44.115	bzq-44-115.bezeqint.net	3938	2.6%
63.195.56.20	dsl.snfc21.pacbell.net	3897	2.6%
130.89.229.48	cal032044.student.utwente.nl	3860	2.6%
212.179.27.6	clnt-27006.bezeqint.net	3666	2.4%

Table 3 - Most Frequent Source Hosts - Alert

The alerts are discussed in more detail below. Table 4 summarizes scan data for the 10 most active destination hosts, while Table 5 does the same for the 10 most active source hosts.

Total Scans: 310477		Total No. Days: 41
Destination IP	# Alerts	% Month
10.1.220.2	11906	3.83%
10.1.218.50	2357	0.76%
10.1.206.94	1796	0.58%
10.1.120.36	1588	0.51%
10.1.205.214	1586	0.51%
10.1.162.77	1753	0.56%
10.1.253.114	1477	0.48%
10.1.60.16	1293	0.42%
10.1.215.210	1365	0.44%
10.1.140.57	1218	0.39%

Table 4 - Most Frequent Destination Hosts – Scans

Total Scans: 310477		Total No. Days: 41	
Source IP	Name	# Alerts	% Month
66.9.27.254	none (Intellispace, Inc. - New York)	20649	6.65%
194.244.78.145	none (Electrolux Zanussi - Italy)	11904	3.83%
63.88.175.201	www.multilateral.com	11718	3.77%
62.252.21.241	pc241-gui4.cable.ntl.com	13057	4.21%
62.157.23.237	p3e9d17ed.dip.t-dialin.net	9641	3.11%
62.96.169.86	m-dialin-86.addcom.de	8939	2.88%
24.23.151.112	cx673530-a.vbch1.va.home.com	8763	2.82%
64.50.161.162	none (CapuNet, LLC - Rockville, MD)	8635	2.78%
63.248.55.245	3ff837f5.dsl.flashcom.net	8561	2.76%
160.78.49.191	ema.chim.unipr.it	7192	2.32%

Table 5 - Most Frequent Source Hosts – Scans

Detects of scans generally result from foreign hosts scanning networks to conduct various forms of reconnaissance – mapping networks, fingerprinting Oss, and locating active services and open ports.

Considering the above two tables, scanning does not seem directed at a particular destination host. And, as *Bayerkohler* found when comparing the data he analyzed with that analyzed by *Zeltser*, the destinations are not remaining constant through time.

Scanning source 66.9.27.254 seemed to be particularly active. No DNS information is available for this host, but it does not seem to be particularly associated with subsequent alerts. Similar to the scanning destinations, none of the sources are the same as they were under *Bayerkohler* or

Zeltser.

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

We recommend that scanning activity continue to be monitored on a regular basis for patterns and relationships with other detect data.

Table 6 summarizes OOS data for the 10 most active destination hosts. Table 7 does the same for the 10 most active source hosts.

Total OOS Alerts: 60119 Total No. Days: 18		
Destination IP	# Alerts	% Month
10.1.217.46	243	0.40%
207.172.3.46	168	0.28%
207.172.3.46	37	0.06%
10.1.211.146	12	0.02%
10.1.201.130	9	0.01%
10.1.106.126	9	0.01%
10.1.201.102	9	0.01%
10.1.223.251	8	0.01%
10.1.214.90	8	0.01%

Table 6 - Most Frequent Destination Hosts – OOS

Total OOS Alerts: 60119		Total No. Days: 18	
Source IP	<u>Name</u>	# Alerts	% Month
208.61.4.207	adsl-61-4-207.mia.bellsouth.net	8431	14.0%
209.92.40.32	dslev1-32.fast.net	5750	9.6%
130.89.229.48	cal032044.student.utwente.nl	5551	9.2%
63.195.56.20	dsl.snfc21.pacbell.net	4749	7.9%
203.32.161.197	adnet.imgserv.com	4511	7.5%
193.64.114.10	net10.printeq.fi	4292	7.1%
210.113.89.200	none (Korea Telcom - Korea)	4224	7.0%
195.103.69.159	proxy.guest.net	3968	6.6%
210.101.101.110	none (Korea Telcom - Korea)	3254	5.4%
212.0.107.107	none (Telson - Spain)	3103	5.2%

Table 7 - Most Frequent Source Hosts - OOS

The OOS data is considered below with the alert data.

Alert Definitions and Potential Severity Assessments

The most frequent detect activity is not necessarily the most severe activity. The sophisticated intruder proceeds quietly without generating lots of noise or notoriety.

Considering the data you have provided us, earlier data previously provided other analysts, and your decisions to define particular alerts based upon source or destination addresses and/or ports, we have attempted to roughly rank the potential severity of the alerts and OOS using the formula proposed by Stephen Northcutt in Network Intrusion Detection, An Analyst's Handbook:

$$\text{Severity} = (\text{Criticality} + \text{Lethality}) - (\text{System Countermeasures} + \text{Network Countermeasures})$$

The potential severity calculations for each type of alert, including the OOS's are summarized in Table 8 and further detailed below.

Name	(Criticality)	(+ Lethality)	(- System Countermeasures)	(+ Network Countermeasures)	= Severity
SUN RPC highport access!	5	5	3	2	5
TCP SMTP source port traffic	4	5	3	1	5
Back Orifice	3	5	3	1	4
WinGate 1080 attempt	2	3	1	1	3
Site exec - possible wu-ftpd exploit - GIAC000623	2	5	3	1	3
External RPC call	3	5	3	2	3
Watchlist 000220 IL-ISDNNET-990517	3	3	3	1	2
SMB name wildcard	3	5	3	3	2
Attempted Sun RPC high port access	3	5	3	3	2
Watchlist 000222 NET-NCFC	3	3	3	1	2
Spp_portscans	2	1	1	1	1
Tiny Fragments - possible Hostile Activity	2	1	1	1	1
Probable NMAP fingerprint attempt	2	1	1	1	1
Queso fingerprint	2	1	1	1	1
Null scan!	2	1	1	1	1
NMAP TCP ping!	2	1	1	1	1
SYN-FIN scan!	2	1	1	1	1
OOS alert	2	1	1	1	1
Connect to 515 from inside	3	3	3	3	0
Broadcast ping to subnet 70	2	1	3	1	-1
Happy 99 Virus	2	2	3	3	-2
SNMP public access	3	2	3	4	-2

Table 8 - Potential Severity Calculations for All Alerts

SUN RPC High Port Access!

This detect indicates a remote host has accessed an RPC high port. This can potentially enable root access to the destination hosts.

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

Potential severity assessment:

Criticality:	5 - Unknown function, assumed critical system.
Lethality:	5 - Can gain root access.
System countermeasures:	3 - Unknown, assumed older OS.
Network countermeasures:	: 2 - Appears to be permissive firewall.

A variety of such exploits are CVE entries or candidates. Among them are the following:

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0003>
<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0631>
<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0632>
<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0624>
<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0626>
<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0666>

RPC exploits are #3 on SANS List of the [Top 10 Vulnerabilities](#).

Remote procedure calls (RPC) allow programs on one computer to execute programs on a second computer. They are widely-used to access network services such as shared files in NFS. Multiple vulnerabilities caused by flaws in RPC, are being actively exploited. There is compelling evidence that the vast majority of the distributed denial of service attacks launched during 1999 and early 2000 were executed by systems that had been victimized because they had the RPC vulnerabilities. The broadly successful attack on U.S. military systems during the Solar Sunrise incident also exploited an RPC flaw found on hundreds of Department of Defense systems.

TCP SMTP Source Port Traffic

This detects traffic from TCP source port 25 which is used by SMTP.

Potential severity assessment:

Criticality:	4 - Would likely be to email host.
Lethality:	5 - Some exploits allow root access.
System countermeasures:	3 - Moderately maintained host assumed.
Network countermeasures:	1 - Assumed not prohibited by firewall.

There are many exploits that take advantage of frequent vulnerabilities in SMTP, some of which allow root access. Among the vulnerabilities and candidates are:

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0512>
<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0531>
<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0261>
<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0095>
<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0096>
<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0203>
<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0207>

Mail vulnerabilities are #5 on SANS List of the [Top 10 Vulnerabilities](#).

Back Orifice

This is designed to detect the likely presence of the Back Orifice trojan on a host. It does so by alerting on attempts to connect to UDP port 31337 on the host. Windows systems are vulnerable to this trojan.

Potential severity assessment:

Criticality:	3 - Unknown function of host.
Lethality:	5 - Can gain root access.
System countermeasures:	3 - Assume moderately maintained hosts.
Network countermeasures:	1 - Appears not be blocked.

It is a candidate under review for CVE:

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0660>

WinGate 1080 Attempt

This detect is designed to alert on scans for systems running SOCKS, or WinGate, a popular firewall/proxy for Windows. Both utilize TCP or UDP port 1080.

These probes are very common. The attacker is usually interested in this service because it may be exploited to bounce connections through the vulnerable host.

Potential severity assessment:

Criticality:	2 - Fingerprinting can be done but chances of damage to hosts minimal
Lethality:	3 - Scans not likely to be attacks, but could lead to access to local network.
System countermeasures:	1 - Worst case assumed.
Network countermeasures:	1 - Assumed not prohibited by firewall.

Although there is no CVS entry the exploit is further described at:

<http://www.whitehats.com/info/IDS175>

SITE EXEC - Possible wu-ftpd exploit - GIAC000623

An attempt has been made to exec a command on an ftp server. Some old versions of wu-ftpd 2.4 and earlier were vulnerable to remote compromise due to poor security restrictions of the site exec command. The detect alerts on "site exec" in the packet content.

Potential severity assessment:

Criticality:	2 - Unknown host function.
Lethality:	5 - Can gain root access.
System countermeasures:	3 - Moderately maintained host assumed.
Network countermeasures:	1 - Unknown.

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

This exploit is a CVE entry:

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0080>

A more detailed analysis is available at:

<http://www.sans.org/infosecFAQ/threats/wu-ftp.htm>

External RPC Call

These detects are generated when external hosts connect to the RPC port (TCP or UDP 111). The portmapper service, which may give attackers information and access to other services, runs at this port.

Potential severity assessment:

Criticality:	3 - Unknown function, assumed moderately critical.
Lethality:	5 - Can gain root access.
System countermeasures:	3 - Unknown, assumed older OS.
Network countermeasures:	2 - Appears to be permissive firewall.

This is a candidate for inclusion on the CVE list because of vulnerabilities associated with the port mapper service.

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0632>

Watchlist 000220 IL-ISDNNET-990517

These detects are generated when an IP connects from a predefined block of network addresses, 212.179.0.0 - 212.179.255.255, an ISDN network in Israel.

The fact that you are monitoring it is indicative of some concern you have regarding traffic from those addresses.

Potential severity assessment:

Criticality:	3 - Unknown host function.
Lethality:	3 - Assume user access possible.
System countermeasures:	3 - Moderately maintained host assumed.
Network countermeasures:	1 - Assumed not prohibited by firewall.

SMB Name Wildcard

This detects attempts to access netbios name services (TCP or UDP ports 137). This is a standard netbios name table retrieval query. Windows machines often exchange these queries as a part of the filesharing protocol to determine netbios names when only IP addresses are known. An attacker could use this same query to extract useful information such as workstation name, domain, and users currently logged in.

Potential severity assessment:

Criticality:	3 - Vulnerable host would be Windows or Samba host.
Lethality:	5 - Can gain root access to system.
System countermeasures:	3 - Moderately maintained host assumed.
Network countermeasures:	3 - Assumed moderate firewall protection.

It is a CVE candidate:

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0621>

This traffic is generally considered noise that should only be considered along with other evidence.

Attempted Sun RPC high port access

This detect is generated by an attempt to contact TCP or UDP port 32771 on a destination host. This port is typically used for RPC services.

Potential severity assessment:

Criticality:	3 - Unknown host function, assumed moderately critical.
Lethality:	5 - Can gain root access.
System countermeasures:	3 - Unknown, assumed older OS.
Network countermeasures:	3 - Assume moderately maintained firewall.

Such exploits are CVE entries or candidates. Among them are the following:

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0003>

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0631>

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0632>

Again RPC exploits are #3 on the SANS Top 10 Vulnerabilities (see above).

Watchlist 000222 NET-NCFC

Similarly these occur when an IP connects from 159.226.0.0 - 159.226.255.255, a block of addresses assigned to The Computer Network Center Chinese Academy of Sciences, Institute of Computing Technology Chinese Academy of Sciences, Beijing, China.

Similar to the previous watchlist, the fact you are monitoring this traffic means it is significant to you.

Potential severity assessment:

Criticality:	3 - Unknown host function.
Lethality:	3 - Assume user access possible.
System countermeasures:	3 - Moderately maintained host assumed.

Network countermeasures: 1 - Assumed not prohibited by firewall.

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

spp_portscans

Normally reconnaissance, these detects are generated by the Snort portscan preprocessor. They are generated when a host attempts to access a certain number or ports within a predetermined time level (the default is 3 ports within 5 seconds). Unless they were false alarms, they would indicate scanning for open ports.

Potential severity assessment:

Criticality:	2 - Fingerprinting can be done but chances of damage to hosts is minimal
Lethality:	1 - Scans are not likely to be attacks.
System countermeasures:	1 - Worst case is assumed.
Network countermeasures:	1 - Assumed not prohibited by firewall.

Tiny Fragments - Possible Hostile Activity

This is likely due to a Snort preprocessor that detects on fragments smaller than a specified threshold (the default is 128).

Potential severity assessment:

Criticality:	2 - Fingerprinting can be done but chances of damage to hosts is minimal
Lethality:	1 - Scans are not likely to be attacks.
System countermeasures:	1 - Worst case is assumed.
Network countermeasures:	1 - Assumed not prohibited by firewall.

Probable NMAP Fingerprint Attempt

This indicates a remote host has sent a TCP packet with the SYN, FIN, URG, and PSH flags set, which may be an attempt using NMAP to fingerprint the destination OS.

Potential severity assessment:

Criticality:	2 - Fingerprinting can be done but chances of damage to hosts is minimal
Lethality:	1 - Scans are not likely to be attacks.
System countermeasures:	1 - Worst case is assumed.
Network countermeasures:	1 - Assumed not prohibited by firewall.

It is a candidate for CVE:

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0454>

Queso Fingerprint

This indicates a remote source has used the Queso tool to attempt to fingerprint the system.

Potential severity assessment:

Criticality:	2 - Fingerprinting can be done but chances of damage to hosts is minimal
Lethality:	1 - Scans are not likely to be attacks.
System countermeasures:	1 - Worst case is assumed.
Network countermeasures:	1 - Assumed not prohibited by firewall.

It is on the same candidate list as NMAP at CVE:

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0454>

Null Scan!

This detects TCP packets with no flags set. These are normally crafted packets to conduct reconnaissance.

Potential severity assessment:

Criticality:	2 - Fingerprinting can be done but chances of damage to hosts is minimal
Lethality:	1 - Scans are not likely to be attacks.
System countermeasures:	1 - Worst case is assumed.
Network countermeasures:	1 - Assumed not prohibited by firewall.

NMAP TCP Ping!

This detect indicates a remote host has used the NMAP portscanning tool to probe the destination by sending an NMAP TCP ping to determine if a host is reachable.

Potential severity assessment:

Criticality:	2 - Fingerprinting can be done but chances of damage to hosts is minimal
Lethality:	1 - Scans are not likely to be attacks.
System countermeasures:	1 - Worst case is assumed.
Network countermeasures:	1 - Assumed not prohibited by firewall.

It is a CVE candidate:

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0523>

SYN-FIN scan

The TCP SYN and FIN flags are both set in the TCP header. This does not occur naturally and probably indicates a crafted packet, most likely to detect the OS.

Potential severity assessment:

Criticality:	2 - Fingerprinting can be done but chances of damage to hosts is minimal
Lethality:	1 - Scans are not likely to be attacks.
System countermeasures:	1 - Worst case is assumed.
Network countermeasures:	1 - Assumed not prohibited by firewall.

OOS Alerts

These alerts were generated by “out of specification” (OOS) packets. These packets were inconsistent with the TCP specifications. Most of them (98.7%) were SYN-FIN packets. The remainder had additional flags set.

Potential severity assessment:

Criticality:	2 - Fingerprinting can be done but chances of damage to hosts minimal
Lethality:	1 - Scans not likely to be attacks.
System countermeasures:	1 - Worst case assumed.
Network countermeasures:	1 - Assumed not prohibited by firewall.

Connect to 515 from Inside

This is generated when a internal host connects to TCP or UDP port 515, the line printer spooler.

Potential severity assessment:

Criticality:	3 - Host function unknown, likely not key host.
Lethality:	3 - Internal user access.
System countermeasures:	3 - Assume moderately maintained host.
Network countermeasures:	3 - Internal access.

This can allow remote attackers to execute arbitrary commands on Linux systems.

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0917>

Broadcast ping to subnet 70

This detects pings broadcast to subnet 10.1.70 on the local network. These normally would be for the purpose of reconnaissance.

Potential severity assessment:

Criticality:	2 - Fingerprinting can be done but chances of damage to hosts minimal
Lethality:	1 - Scans not likely to be attacks.
System countermeasures:	3 - Assumed moderately maintained.
Network countermeasures:	1 - Assumed not prohibited by firewall.

Happy 99 Virus

This detect results from TCP connects to port 25 on the destination host that contain the content "X-Spanska\ Yes". The Happy 99 virus modifies system files on Windows computers and then further propagates in email sent from those hosts.

Potential severity assessment:

Criticality:	2 - Likely Windows workstation.
Lethality:	2 - Changes system files, but does not disable functionality.
System countermeasures:	3 - Assumed older OS, moderately maintained.
Network countermeasures:	3 - Assumed firewall, moderately maintained.

SNMP Public Access

This detects connections to the SNMP ports (TCP or UDP 161) on the host.

Potential severity assessment:

Criticality:	3 - Unknown function of hosts.
Lethality:	2 - Could gain user name and password information.
System countermeasures:	3 - Moderately maintained hosts assumed.
Network countermeasures:	4 - Assumed protected by firewall as all contacts internal.

For Windows NT, this is a candidate because the intruder could obtain a list of user names:

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0499>

Analysis of Potentially More Severe Alerts

Due to time constraints, we determined to focus our more detailed analysis on those alerts we believe could do the most damage to your particular network. These alerts are ranked 2 and above in our potential severity assessments.

The relative frequency of all alerts, including the OOS, are summarized below:

Total # of Alerts: 210447*			Total # Days: 53	
Name	# Alerts	% of Total	# Sources	# Dest
OOS Alerts *	60119	28.57%	210	26438
SYN-FIN scan!	39871	18.95%	0	0
Spp portscans	30997	14.73%	0	0
Watchlist 000220 IL-ISDN-990517	8134	3.87%	61	42
Watchlist 000222 NET-NCFC	4764	2.26%	652	2856
WinGate 1080 attempt	2893	1.37%	4	2892
TCP SMTP source port traffic	2542	1.21%	27	34
Attempted Sun RPC high port access	1813	0.86%	221	2
Broadcast ping to subnet 70	1697	0.81%	41	1035
Back Orifice	468	0.22%	23	2
SNMP public access	277	0.13%	208	201
Null scan!	218	0.10%	34	35
SMB name wildcard	142	0.07%	33	61
Queso fingerprint	96	0.05%	26	26
NMAP TCP ping!	60	0.03%	15	13
SUNRPC highport access!	56	0.03%	2	3
Connect to 515 from inside	15	0.01%	14	14
Probable NMAP fingerprint attempt	13	0.01%	8	5
External RPC call	13	0.01%	3	8
SITE EXEC - possible wu-ftpd exploit - GIAC000623	7	0.00%	4	8
Tiny Fragments - possible hostile activity	2	0.00%	2	2
Happy 99 virus	0	0.00%	56	2
* Includes OOS Alerts, however for only 18 days.				

Table 9 - Alerts, Ranked by Frequency

© SANS

These same alerts, ranked by severity are summarized below:

Total # of Alerts: 210447*			Total # Days: 53		
Name	# Alerts	Severity	% of Total	# Sources	# Dest
TCP SMTP source port traffic	2893	5	1.37%	4	2892
SUNRPC highport access!	60	5	0.03%	15	13
Back Orifice	1697	4	0.81%	41	1035
WinGate 1080 attempt	4764	3	2.26%	652	2856
External RPC call	13	3	0.01%	8	3
SITE EXEC - possible wu-ftpd exploit - GIAC000623	13	3	0.01%	5	8
Watchlist 000220 IL-ISDNNET-990517	30997	2	14.73%	77	116
Watchlist 000222 NET-NCFC	8134	2	3.87%	61	42
Attempted Sun RPC high port access	2542	2	1.21%	27	34
SMB name wildcard	218	2	0.10%	34	35
SYN-FIN scan!	56250	1	26.73%	30	39880
Spp portscans	39871	1	18.95%		
Null scan!	277	1	0.13%	208	201
Queso fingerprint	142	1	0.07%	33	61
NMAP TCP ping!	96	1	0.05%	26	26
Probable NMAP fingerprint attempt	15	1	0.01%	14	14
Tiny Fragments - possible hostile activity	7	1	0.00%	5	6
OOS Alerts*	60119	1	0.00%	210	26438
Connect to 515 from inside	1813	0	0.86%	221	2
Broadcast ping to subnet 70	468	-1	0.22%	23	2
SNMP public access	2	-2	0.00%	2	2
Happy 99 virus	0	-2	0.00%	0	0
* Includes OOS Alerts, however for only 18 days.					

Table 10 - Alerts, Ranked by Potential Severity

Those alerts ranked 2 and above in Table 10 are analyzed in more detail below.

SUN RPC High Port Access!

Source and destination activity for this particular alert is summarized in the following tables:

Source IP	Frequency	% of Total
216.10.12.30	33	55.000%
216.148.218.160	6	10.000%
205.188.3.211	4	6.667%
195.34.28.117	3	5.000%
205.188.3.239	3	5.000%
24.18.90.197	3	5.000%
205.188.4.2	2	3.333%
129.123.6.14	1	1.667%
205.188.1.105	1	1.667%
211.46.110.81	1	1.667%

Table 11 - Sun RPC High Port Access - Most Frequent Sources

Destination IP	Frequency	% of Total
10.1.206.222	21	35.000%
10.1.202.242	20	33.333%
10.1.212.186	4	6.667%
10.1.228.62	3	5.000%
10.1.97.59	3	5.000%
10.1.253.114	2	3.333%
10.1.53.23	2	3.333%
10.1.140.51	1	1.667%
10.1.179.78	1	1.667%
10.1.206.218	1	1.667%

Table 12 - Sun RPC High Port Access - Most Frequent Destinations

The majority (55%) of these alerts were caused by source 216.10.12.30 (gravity.cpanel.net). It is unclear what services it's running but it is assigned to Darkorb Communications, Wilmington, DE. All detects were from source port 2078.

Ten percent of these alerts were triggered by traffic from the second most frequent source host, 216.148.218.160, which resolves to "head.rwc.rhns.redhat.com" and appears to be running Apache. It accessed destination 10.1.206.222 on October 26, and several times during early November. Each time the source port was 443, which should be SSL.

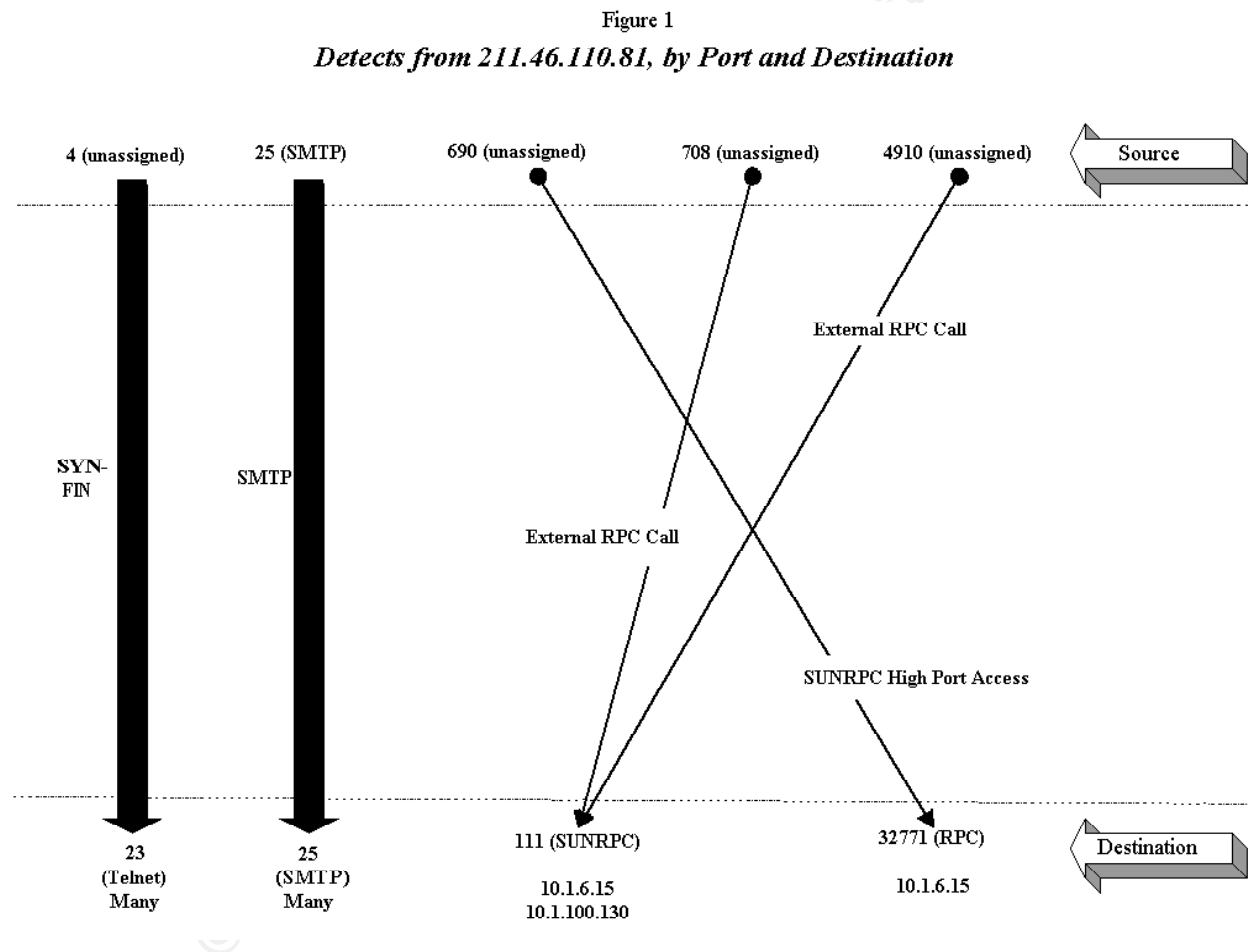
Two destination hosts, 10.1.206.222 and 10.1.202.242, account for over 68% of the alerts.

Other analysts have noted the primary alert destination, 10.1.206.222. [Teri Bidwell](#) notes this was a scan destination for the same source, 63.248.55.245. [Mark Gryparis](#) associates traffic to this server with Napster.

Of particular interest is the tenth source in Table 11, 211.46.110.81. Although it was responsible for only one SUN RPC High Port Access detect, its association with other detects may be revealing.

There is no DNS information available for this source. It is among a netblock assigned to the Korean Network Information Center and appears to be running Sendmail 8.9.3. It is responsible for 2068 alert instances involving 2048 distinct destination hosts in a less than 24 hour period November 10 and 11. Most of these were TCP SMTP source port traffic (see discussion on next alert that follows), followed by SYN-FIN scans, two external RPC calls, and concluding with one SUN RPC high port access.

It's helpful to graph the alert activity generated by this source, 211.46.110.81.



If this source is in fact a mail server or client, why did it send a large number SYN-FIN scans to port 23 over a one day period to a variety of destination hosts? Did someone have an interest in determining if telnet ports were open? And what is going on with the RPC alerts?

Unfortunately, these questions cannot be answered without further review. We only have half the data at best, since we lack information about what packets were sent from the destination hosts to this host.

We recommend the following:

- An immediate review of any packet data available for hosts on your network sending to 211.46.110.81 as a destination
- A review of the services running on all destination hosts identified in Table 12
- Elimination of any SUN High Port services running on externally exposed hosts where at all possible
- Ensure all hosts are updated with current OS patches
- Isolate hosts running SUN High Port services behind a stateful firewall

TCP SMTP Source Port Traffic

Over the 53 day period, there were only 4 source hosts that generated this alert, yet there were over 2800 destinations were involved (see Table 2 above). Source and destination activity for this particular alert is summarized in the following tables:

Source IP	Frequency	% of Total
211.46.110.81	1789	61.839%
24.7.227.215	1096	37.885%
194.67.168.11	6	0.207%
194.88.77.240	2	0.069%

Table 13 - TCP SMTP Source Port Traffic - Most Frequent Sources

Source IP	Frequency	% of Total
10.1.1.2	2	0.069%
10.1.10.103	2	0.069%
10.1.10.44	2	0.069%
10.1.104.129	2	0.069%
10.1.109.202	2	0.069%
10.1.109.218	2	0.069%
10.1.11.35	2	0.069%
10.1.110.18	2	0.069%
10.1.110.232	2	0.069%
10.1.112.182	2	0.069%

Table 14 - TCP SMTP Source Port Traffic - Most Frequent Destinations

Nearly 62% of the alerts were from source IP 211.46.110.81, which we discussed in some detail above.

99.7% of the destination host ports were also SMTP (25). We presume these alerts are generated by the exchange of mail between your network and the various external hosts.

Significantly, the most frequent host also generated alerts for SYN-FIN scans, external RPC calls, and ultimately Sun RPC high port access to 10.1.6.15 on November 11. The vulnerability of this destination host was noted previously by several analysts who documented external RPC calls. See, for example, Brent Deterding, Claudio Silotto, Curt Blais and Kevin Orkin

Most of the other alerts were from 24.7.227.215. Again, no DNS information is available, but it is among the addresses assigned to the @home net block. This host does not appear to be running SMTP. We presume this host was scanning for open ports.

In addition to the recommendations made above, we recommend the following:

- verifying each of the destination hosts are, in fact, mail servers
- disabling SMTP on any hosts not requiring that service
- ensuring all mail servers are fully patched with current SMTP and OS patches
- consider moving all mail servers to a DMZ outside your local networks

Back Orifice

1,697 alerts were generated during the period from 41 sources to 1,035 destinations (Table 9). Source and destination activity for this particular alert is summarized in the following tables:

Source IP	Frequency	% of Total
62.136.90.120	306	18.032%
63.46.46.143	291	17.148%
203.148.182.108	111	6.541%
213.43.69.72	99	5.834%
203.155.130.111	79	4.655%
209.94.199.186	78	4.596%
203.148.183.44	75	4.420%
213.43.69.126	75	4.420%
168.120.12.33	70	4.125%
209.94.199.141	69	4.066%

Table 15 - Back Orifice - Most Frequent Sources

Destination IP	Frequency	% of Total
10.1.97.208	7	0.412%
10.1.98.150	7	0.412%
10.1.97.142	6	0.354%
10.1.98.119	6	0.354%
10.1.98.151	6	0.354%
10.1.98.77	6	0.354%
10.1.98.81	6	0.354%
10.1.98.82	6	0.354%
10.1.97.115	5	0.295%
10.1.97.118	5	0.295%

Table 16 - Back Orifice - Most Frequent Destinations

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

Over 50% of the detects were attributable to the top 5 source hosts.

62.136.90.120 (modem-120.dextroamphetam.dialup.pol.co.uk) with the Planet Online ISP in London generated 306 alerts, apparently scanning 189 destinations during a 90 minute period October 14. 63.46.46.143 (1cust143.tnt2.sierra-vista.az.da.uu.net) with the UUNET generated 291 alerts, apparently scanning 291 destinations on October 29. 203.148.182.108 assigned to the A-Net ISP in Bangkok, Thailand generated 111 alerts, apparently scanning some 100 destination hosts October 20.

Judging from the wide distribution of destinations, most of the activity here appears to be reconnaissance scanning for port 31337. Previous analysts have noted similar scanning behavior, but to different hosts. See, for example, [Dan Eberlein's](#) analysis.

We recommend:

- identifying your Windows hosts that may be subject to these scans
- scheduled running of trojan detection / removal software (such as “The Cleaner” <http://www.moosoft.com/>) on these hosts
- ensuring these systems are fully patched
- putting these systems behind a stateful firewall, if possible, and blocking traffic to port 31337

WinGate 1080 Attempt

This alert was generated 4,764 times over the period, from 652 source hosts to 2,856 destinations (Table 9).

Source and destination activity for this particular alert is summarized in the following tables:

Source IP	Frequency	% of Total
63.193.210.208	1883	39.526%
208.194.161.155	220	4.618%
198.63.2.192	179	3.757%
204.117.70.5	154	3.233%
64.86.5.250	135	2.834%
207.114.4.46	129	2.708%
212.72.75.236	113	2.372%
63.26.7.170	95	1.994%
24.169.61.162	89	1.868%
168.120.16.250	72	1.511%

Table 17 - WinGate 1080 Attempt - Most Frequent Sources

Destination IP	Frequency	% of Total
10.1.206.118	372	7.809%
10.1.225.154	126	2.645%
10.1.60.11	67	1.406%
10.1.60.8	64	1.343%
10.1.60.16	39	0.819%
10.1.203.78	34	0.714%
10.1.60.38	33	0.693%
10.1.53.91	29	0.609%
10.1.222.102	24	0.504%
10.1.53.219	24	0.504%

Table 18 - WinGate 1080 Attempt - Most Frequent Destinations

Nearly 40% of the activity is from source IP, 63.193.210.208 (adsl-63-193-210-208.dsl.snfc21.pacbell.net) which appears to be running Apache and an ESMTP service. All 1883 detects to 1837 separate destination hosts were during a 5 minute period around 7 p.m. on one day, October 5.

The next most frequent source IP, 208.194.161.155 (proxy.monitor.twisted.ma.us.dal.net) appears to be running Apache and Sendmail 8.9.3 at the First Internet Alliance in Hopington, MA. No particular pattern is discernible. Multiple destination hosts are involved and contacts are regular throughout September and October.

The destinations appear to be proxy servers and many of these detects appear to be reconnaissance scans in an effort to find vulnerable hosts. As [Lenny Zeltser](#) and others have noted, this activity is probably due to internet chat, a subject that is further discussed below under “Attempted SUN RPC High Port Access.” The scans are conducted by IRC servers to locate systems that may be used to forward their traffic. Zeltser notes that 207.114.4.46, among the top 10 in Table 17, is associated with [undernet.org](#).

We recommend the following:

- reviewing the destination hosts to verify their proxy server functionality
- disabling SOCKS or WinGate on any hosts that do not require it
- eliminating ICQ traffic from your network
- ensuring all hosts are updated with current OS patches
- placing internet proxy servers in a DMZ
- placing intranet proxy servers behind a stateful firewall

SITE EXEC - Possible wu-ftpd exploit - GIAC000623

A total of 13 alerts were generated, all during October. Five sources and eight destinations were involved. Source and destination activity for this particular alert is summarized in the following tables:

Source IP	Frequency	% of Total
208.61.44.215	9	69.231%
24.31.88.99	2	15.385%
202.9.188.89	1	7.692%
63.202.13.20	1	7.692%

Table 19 - Possible Wu-ftpd Exploit - Most Frequent Sources

Destination IP	Frequency	% of Total
10.1.205.94	4	30.769%
10.1.130.242	3	23.077%
10.1.221.82	2	15.385%
10.1.100.209	1	7.692%
10.1.130.81	1	7.692%
10.1.97.206	1	7.692%
10.1.99.130	1	7.692%

Table 20 - Possible Wu-ftpd Exploit - Most Frequent Destinations

208.61.44.215 (adsl-61-44-215.mia.bellsouth.net) accounted for nearly 70% of the alerts. On October 1, three alerts were generated to destination host 10.1.205.94, the first at 06:17:23 and the next two at 07:46:18 and 07:46:19. Three alerts were also generated to 10.1.130.242 within a 4 second period. Individual alerts were associated with destinations 10.1.97.206, 10.1.99.130 and 10.1.130.81.

24.31.88.99 (a24b31n88client99.hawaii.rr.com) accounted for 2 alerts. Both were to 10.1.221.82 on October 16, within a couple of minute period.

202.9.188.89, assigned to the Dishnet in Chennai, India, had one detect to 10.1.205.94 October 7.

63.202.13.20 (adsl-63-202-13-20.dsl.snfc21.pacbell.net) had one detect to 10.1.100.209 October 4. It's notable that this source address has been involved in Queso fingerprinting of multiple destination hosts.

Previous analysts have noted earlier scanning activity against destination 10.1.205.94 ([Jason Baeder](#), [Gilbert Green](#), [Kathryn Lucas](#) and [Joseph R Rach](#)). This activity was associated with different source IPs than were involved here.

We recommend the following:

- reviewing the destination hosts to determine if they are susceptible to this exploit (Linux boxes are particularly vulnerable)
- ensuring all hosts are updated with current OS patches
- placing any required internet FTP servers in a DMZ

- wherever possible, placing hosts behind a stateful firewall

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

External RPC Call

Source and destination activity for this particular alert is summarized in the following tables:

Source IP	Frequency	% of Total
63.162.239.69	3	23.077%
200.191.80.181	2	15.385%
200.191.80.206	2	15.385%
211.46.110.81	2	15.385%
12.34.21.196	1	7.692%
24.23.151.112	1	7.692%
24.7.227.215	1	7.692%
38.200.223.8	1	7.692%

Table 21 - External RPC Call - Most Frequent Sources

Destination IP	Frequency	% of Total
10.1.6.15	9	69.231%
10.1.100.130	3	23.077%
10.1.15.127	1	7.692%

Table 22 - External RPC Call - Most Frequent Destinations

There were 8 sources with 3 destinations generating this alert.

DNS and registration information was reviewed for each source host. No particular pattern is discernable. Two hosts, 211.46.110.81 (discussed above) and 24.7.227.215 (on the @home network), were actively involved in scanning activity.

Other analysts have noted this alert with these destination hosts. [David Whyte](#), [Dale Ross](#) and [Kevin Orkin](#) found earlier detects to 10.1.6.15. Orkin also noted the detects with 10.1.100.130 and 10.1.15.127 (which also had SMB related alerts).

We recommend the following:

- disabling the portmapper service on all hosts that don't require it
- reviewing the services provided by each of the three destination hosts above
- ensuring all hosts are updated with current OS patches
- ensuring SMB services are not provided on external interfaces
- securing hosts using portmapper services behind a stateful firewall

Watchlist 000220 IL-ISDNNET-990517

30,977 alerts were generated from 77 source hosts accessing 116 separate destinations during the period (Table 9). Source and destination activity for this particular alert is summarized in the following tables:

Source IP	Frequency	% of Total
212.179.95.5	5683	18.334%
212.179.79.2	3950	12.743%
212.179.27.6	3666	11.827%
212.179.72.226	795	2.565%
212.179.30.113	579	1.868%
212.179.24.136	475	1.532%
212.179.56.5	439	1.416%
212.179.23.95	415	1.339%
212.179.45.241	272	0.878%
212.179.19.134	214	0.690%

Table 23 - Watchlist 0220 - Most Frequent Sources

Destination IP	Frequency	% of Total
10.1.211.146	4810	15.518%
10.1.223.98	3938	12.704%
10.1.206.90	3914	12.627%
10.1.203.142	1638	5.284%
10.1.218.142	1459	4.707%
10.1.214.170	1353	4.365%
10.1.202.22	950	3.065%
10.1.201.174	796	2.568%
10.1.214.74	667	2.152%
10.1.209.106	648	2.091%

Table 24 - Watchlist 0220 - Most Frequent Destinations

They are not among the most active scanning sources. Of 310,477 total scans during the period there were only 7 from this network:

```
Oct 13 05:51:59 212.179.41.24:1031 > 10.1.214.170:6699 INVALIDACK ***FRPAU
Oct 13 05:55:25 212.179.41.24:1031 > 10.1.214.170:6699 UNKNOWN *1*F**A* RESERVEDB
Oct 13 05:55:30 212.179.41.24:1031 > 10.1.214.170:6699 INVALIDACK 21S*RPA* RESERV
Oct 13 06:00:18 212.179.41.24:1031 > 10.1.214.170:6699 NOACK *1*FR**U RESERVEDBIT
Oct 13 06:16:29 212.179.41.24:1031 > 10.1.214.170:6699 VECNA *1*F*P** RESERVEDBIT
Nov 11 07:51:31 212.179.27.6:0 > 10.1.206.90:1498 NOACK 2***RP**RESERVEDBITS
Nov 11 08:20:06 212.179.27.6:2078 > 10.1.206.90:4619 NOACK *1**RP**RESERVEDBITS
```

31% of the alerts from this network were to hosts on destination port 6699, a port known to be used by Napster, which may pose security risks. See, for example:

<http://www.sans.org/infosecFAQ/threats/napster.htm>

The association of Napster traffic with these sources has been noted by previous analysts, such as [Guy Bruneau](#) and [Kathryn Lucas](#).

There were no additional potentially severe alerts, nor were there any OOS alerts, from these sources.

We recommend the following:

- review the destination hosts in the 7 anomalous scans above for possible compromise
- consider whether Napster traffic is warranted on these (or any other) destination hosts
- ensuring all hosts are updated with current OS patches

SMB Name Wildcard

Many were generated from internal source hosts connecting with internal destination hosts. Source and destination activity for this particular alert is summarized in the following tables:

Source IP	Frequency	% of Total
10.1.101.160	93	42.661%
141.157.99.21	33	15.138%
169.254.184.161	24	11.009%
141.157.98.201	20	9.174%
10.1.98.154	5	2.294%
10.1.97.207	4	1.835%
129.37.159.177	4	1.835%
10.1.97.120	3	1.376%
130.227.195.57	3	1.376%
10.1.101.113	2	0.917%

Table 25 - SMB Name Wildcard - Most Frequent Sources

Destination IP	Frequency	% of Total
10.1.101.192	93	42.661%
10.1.6.15	53	24.312%
10.1.101.53	9	4.128%
10.1.101.117	7	3.211%
10.1.101.153	7	3.211%
10.1.100.130	4	1.835%
10.1.101.147	4	1.835%
10.1.101.89	4	1.835%
10.1.101.113	3	1.376%
10.1.101.145	3	1.376%

Table 26 - SMB Name Wildcard - Most Frequent Destinations

Note that nearly 50% of the activity among the top 10 sources was within your own network.

No particular pattern was noted with regard to the various external source hosts.

Other analysts have noted this netbios name activity (see, for example, [Robert Currie](#) and [Karen Frederick](#)) and it is a well known phenomenon in the Windows environment. [Guy Bruneau](#) observed significant activity between 10.1.101.160 and 10.1.101.192.

We recommend the following:

- ensuring all hosts are updated with current OS patches
- consider blocking netbios traffic at the firewall
- securing hosts running netbios services behind a stateful firewall

Attempted Sun RPC high port access

99.6% of these alerts originated from source port 4000. Considering that the top 10 source IPs are addresses associated with the AOL ICQ network, it is likely these alerts were generated by ICQ traffic. Source and destination activity for this particular alert is summarized in the following tables:

Source IP	Frequency	% of Total
205.188.153.108	628	24.705%
205.188.153.107	517	20.338%
205.188.153.116	435	17.113%
205.188.153.109	334	13.139%
205.188.153.101	110	4.327%
205.188.153.102	101	3.973%
205.188.153.99	98	3.855%
205.188.153.104	91	3.580%
205.188.153.110	59	2.321%
205.188.153.100	51	2.006%

Table 27 - Attempted Sun RPC High Port Access - Most Frequent Sources

Destination IP	Frequency	% of Total
10.1.221.246	488	19.197%
10.1.225.210	435	17.113%
10.1.217.214	365	14.359%
10.1.206.222	299	11.762%
10.1.222.98	187	7.356%
10.1.226.74	154	6.058%
10.1.228.42	132	5.193%
10.1.227.50	97	3.816%
10.1.152.198	61	2.400%
10.1.223.18	53	2.085%

Table 28 - Attempted Sun RPC High Port Access - Most Frequent Destinations

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

The top two source hosts, 205.188.153.108 and 205.188.153.107, resolve to “fes-d012.icq.aol.com” and “fes-d011.icq.aol.com” respectively. The “icq” in their names is a clue to what they’re doing.

With few exceptions all this traffic is from source port 4000, which is used by [ICQ](#) (UDP). (This port (both TCP and UDP) is also used by [Terabase](#), but we don’t believe that’s the case here.)

Occasional non-AOL e.g. 63.83.105.226, assigned to UUNET, on November 6 had three contacts within a 2 second period from source port 2629 which is used by [Sitara Server](#) to destination 10.1.205.130. This is a QOS device, which may in fact be utilized by UUNET.

There are a variety of trojans that are spread through the ICQ channel, such as ICQ 2000, ICQ IP Sniffer, ICQ Nail, ICQ Relay, ICQ Revenge and ICQ Trojan. See <http://www.moosoft.com/tdbindex.php>.

[Zeltser](#) and others have noted this ICQ traffic in their analyses.

We recommend the following:

- disabling ICQ traffic internally and blocking it at the firewall
- ensuring all hosts are updated with current OS patches

Watchlist 000222 NET-NCFC

Source and destination activity for this particular alert is summarized in the following tables:

Source IP	Frequency	% of Total
159.226.45.3	6295	77.391%
159.226.91.20	1209	14.864%
159.226.41.166	123	1.512%
159.226.5.77	87	1.070%
159.226.228.1	58	0.713%
159.226.157.1	38	0.467%
159.226.66.130	33	0.406%
159.226.92.10	29	0.357%
159.226.114.1	21	0.258%
159.226.63.200	20	0.246%

Table 29 - Watchlist 000222 - Most Frequent Sources

Destination IP	Frequency	% of Total
10.1.6.7	5793	71.220%
10.1.100.230	1286	15.810%
10.1.253.43	461	5.668%
10.1.253.41	179	2.201%
10.1.253.42	151	1.856%
10.1.99.51	70	0.861%
10.1.100.81	53	0.652%
10.1.145.9	41	0.504%
10.1.145.18	13	0.160%
10.1.6.34	13	0.160%

Table 30 - Watchlist 000222 - Most Frequent Destinations

8,134 alerts were generated by 61 source addresses contacting 42 separate hosts during the period. A detailed review of the data indicates 96% of these alerts were to destination port 25 (SMTP).

No scanning activity was recorded from these sources. There were none of the more severe alerts, nor were there any OOS packet alerts, as a result of activity from these sources.

96% of the traffic was to destination port 25 and appears to be mail.

[Gilbert Green](#) noted some probing activity June 6 of 10.1.6.7 from 198.86.17.38 and 24.23.45.19.

We recommend the following:

- scanning 10.1.6.7 for trojans
- verifying you wish to continue mail exchange with the 159.226 network
- verifying each of the destination hosts are, in fact, mail servers
- disabling SMTP on any hosts not requiring that service
- ensuring all mail servers are fully patched with current SMTP and OS patches
- consider moving all mail servers to a DMZ outside your local networks

Conclusion

Your network is regularly subject to active scanning in search of information that may lead to identifying vulnerable hosts. It is imperative that the network be vigorously monitored to protect against exploits of any identified vulnerabilities.

Specific recommendations have been made above particular to those alerts we consider more severe, given what we know about your network. We encourage you to review and wherever possible implement these recommendations as soon as feasible.

Assignment 3 - Analysis process

A variety of tools and methods were used to assess, organize, modify, analyze and present the data. Both Linux and Windows platforms were used.

The assignment data was downloaded from SANS in zipped file format (snortA.zip, snortS.zip, and OOS.zip). These files were decompressed into separate alert, scan and OOS directories onto a Samba shared folder on a Linux box:

```
/gcia/alert  
/gcia/scan  
/gcia/OOS
```

Using [EditPadPro](#) from a Windows 2000 workstation, the individual files were then perused to gain a general understanding of what they contained, and to note the specific dates covered by the contents of each individual file.

I concluded that “snortA” files contained Snort “fast” alerts, “snortS” files contained Snort portscans and “OOS” files contained “out of spec” packet details. These “OOS” files appeared generated from binary packet capture using “readback” mode and a snap length of 68.

An Excel spreadsheet was created to display dates, days of week and the type(s) of data sets available for each date. Three pairs of apparently duplicate files were noted:

```
snortA14.txt and snortA19.txt  
snortS20.txt and snortS23.txt  
OOSche4.txt and OOSche5.txt
```

Available [tools](#) for summarizing and analyzing Snort data were reviewed, and I concluded that “[Snortsnarf](#)” would be key.

To prepare the data for Snortsnarf, I removed headers from individual files using EditPadPro. I then compared what appeared to be redundant files and deleted the redundancies. From the Windows 2000 command line, the three pairs of apparently redundant files were compared using the compare command:

```
comp file1 file2
```

It was found that:

```
snortA14.txt = snortA19.txt, so snortA14.txt was deleted  
snortS20.txt = snortS23.txt, so snortS23.txt was deleted  
OOSche4.txt = OOSche5.txt, so OOSche5.txt was deleted
```

The Excel spreadsheet created earlier was revised to reflect the deleted *.txt files. The result appears as Table 1 in Assignment 2.

The data was then concatenated into various combinations. From the Linux command line, I concatenated alert and portscan data, sorting the data in chronological order within their respective data directories:

```
sort -o alert *  
sort -M -o portscan *
```

From the OOS directory, the OOS files were concatenated into a file named "alert":

```
cat * > alert
```

Since Snortsnarf will not recognize "MY.NET" in an IP address, I used EditPadPro to search and replace "MY.NET" with "10.1"

From the Linux command line, I then attempted to run Snortsnarf.pl (v.011601.1) on my three data sets, alert, scan, and OOS:

```
cd /usr/local/src/snortsnarf
```

```
./snortsnarf.pl -d /gcia /gcia/alert
```

(The command appeared to complete but indicated "Segmentation Fault".)

```
./snortsnarf.pl -d /gcia/scan /gcia/portscan
```

(The command did not complete, and indicated "Segmentation Fault".)

```
./snortsnarf.pl -d /gcia/OOS /gcia/OOS/alert
```

(The command appeared to run quickly, but no web pages were produced.)

Given only partial success, I decided to split the data up into individual months, thinking that Snortsnarf was choking on the size of some of the data segments. I therefore created separate subdirectories for each data type:

```
/gcia/alert/Sep  
/gcia/alert/Oct  
/gcia/alert/Nov  
/gcia/scan/Sep  
/gcia/scan/Oct  
/gcia/scan/Nov  
/gcia/OOS/Aug  
/gcia/OOS/Oct  
/gcia/OOS/Nov
```

The relevant header stripped snortA.txt, snortS.txt, and OOS*.txt files were copied into their monthly directories, the individual months were concatenated, and the "MY.NET" entries were replaced with "10.1" entries.

Snortsnarf was then rerun on the concatenated files in each of the monthly subdirectories. Although the pesky “Segmentation fault” message did appear several times, the appropriate “index.html” files were created for the monthly alert and scan data. Nothing was created for the OOS data, however.

The main and subordinate Snortsnarf perl scripts were then studied to try to understand what the problem might be with the OOS data. It appeared that, for alert data, Snortsnarf might want to see a first line in the format:

[**] Alert name [**]

So, for OOS files, I again used EditPadPro to insert a new first line for each record:

[**] OOS Alert [**]

Voila! Snortsnarf was successfully rerun on the cumulative and monthly OOS data.

I was concerned that the “Segmentation fault” errors might have meant that some of the resulting Snortsnarf html index files would be incomplete. So I examined the the total and monthly Snortsnarf detect counts, and compared them with the earliest and latest dates in the raw data sets to verify completeness. The totals appeared to be correct.

I was then faced with the problem of how to get the data into a database or spreadsheet format, so that various queries and sorts could be run. Not being particularly proficient at perl or Mysql, I knew I couldn't rely strictly on Snortsnarf's excellent summary information or EditPadPro's search and count features.

The alert and scan files were imported into Microsoft Access using the Get External Data feature. The first step was to import the file into an Access table in fixed format because there were no reliable field markers to bring the data in with field delimiters. The second step was to use the Search and Replace feature in Access to put in field delimiters so that I could then export it into a new file with delimiters. Once this was accomplished, I imported the delimited data into a new Access table using the delimiters (instead of fixed format). This resulted in each field being uniquely defined in each table (alert and scan) so that queries and sorts could be run against them.

The OOS data was a little more difficult to import because each record in the “txt” file was actually a separate record. So, the first problem that had to be resolved was eliminating the line breaks for each record. I used EditPadPro's Regular Expression Search and Replace feature to eliminate the line breaks and prepare the file for importing into Access. The Regular Expression feature is Perl 5 compatible so if you are a Perl master, this feature can be very powerful. If you are not a Perl master, like myself, the help files were easy to follow.

Once this was done, then I was able to follow the steps to import the data into Access in the same manner in which the alerts and scans were imported.

Much of the table manipulation and formatting was done using Microsoft Excel because it is less complicated to use than Access. To get the data from Access to Excel required defining a query

(or series of queries) to isolate the desired records then exporting the resulting table into Excel.

Once the data was in Excel, the tables in Assignment 2 were produced, along with other iterations that did not end up in the final document. These various tables were used to understand the relationships among involved hosts and various alerts and scans

I then began to inspect the data, particularly the alert and OOS, using Snortsnarf. By using a browser on the index.html file created by Snortsnarf, one is presented with a page such as the following:

SILICON DEFENSE **SnortSnarf start page**
All Snort signatures
SnortSnarf v011601.1

11677 alerts found among the files:

- sepalert

Earliest alert at **00:00:52.873106** on 09/26
Latest alert at **23:31:23.130090** on 09/30

Signature (click for definition)	# Alerts	# Sources	# Destinations	Detail link
Tiny Fragments - Possible Hostile Activity	3	2	3	Summary
SUNRPC highport access!	3	1	2	Summary
NMAP TCP ping!	4	2	2	Summary
Null scan!	27	17	17	Summary
Attempted Sun RPC high port access	31	2	2	Summary
Queso fingerprint	32	5	7	Summary
Watchlist 000222 NET-NCFC	192	12	8	Summary
Watchlist 000220 IL-ISDNNET-990517	358	8	13	Summary
WinGate 1080 Attempt	429	91	161	Summary
SYN-FIN scan!	10598	2	9782	Summary

SnortSnarf brought to you courtesy of Silicon Defense
Authors: [Jim Hoagland](#) and [Stuart Stanford](#)
See also the [Snort Page](#) by Marty Roesch
Page generated at Thu Feb 15 11:05:24 2001

By clicking on the Detail Link Summary for one of the detects, your are presented with further detail about the sources and destinations:



SnortSnarf signature page

Attempted Sun RPC high port access

[SnortSnarf v011601.1](#)

31 alerts with this signature among the files:

- [sepalert](#)

Earliest such alert at **08:34:21.306733** on 09/26
Latest such alert at **10:16:30.954977** on 09/30

Attempted Sun RPC high port access [2 sources](#) [2 destinations](#)

Sources triggering this attack signature


Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
205.188.153.105	29	29	1	1
205.188.179.33	2	2	1	1

Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
10.1.220.78	29	29	1	1
10.1.204.134	2	2	1	1

[SnortSnarf](#) brought to you courtesy of [Silicon Defense](#)
Authors: [Jen Hoagland](#) and [Stuart Stanford](#)
See also the [Snort Page](#) by Marty Roesch
Page generated at Thu Feb 15 11:05:24 2001

By clicking on the link to a particular Source or Destination IP, one should then be led to a page with more detail about the particular alerts:



SnortSnarf alert page

Source: **205.188.153.105**

SnortSnarf v011601.1

29 such alerts among the files:

- sepalert

Earliest **08:34:21** 306733 on 09/26
Latest **11:29:11** 794697 on 09/26

1 different signatures are present for 205.188.153.105 as a source

- 29 instances of [Attempted Sun RPC high port access](#)

There are 1 distinct destination IPs in the alerts of the type on this page.

205.188.153.105	Whois lookup at:	ARIN	RIPE	APNIC	Geektools
	DNS lookup at:	Amenes	TRIUMF	Riherds	Princeton

09/26-08:34:21.306733	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-08:34:21.639327	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-08:34:30.309065	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-08:35:19.582726	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-08:36:18.395768	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-08:38:19.373983	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-08:57:17.714289	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-08:59:17.574119	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-09:04:17.116627	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-09:12:16.432173	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-09:18:15.908345	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-09:36:14.452562	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-09:36:18.242702	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-09:40:14.094553	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-09:45:13.651237	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-09:50:13.209342	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-09:55:12.791569	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-10:00:13.289580	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-10:16:12.388802	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-10:29:19.586565	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-10:38:18.853064	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-10:41:18.599309	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-10:47:04.778266	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-11:00:16.958828	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-11:13:17.793755	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-11:20:17.212712	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-11:21:17.153529	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-11:29:05.793460	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771
09/26-11:29:11.794697	[**]	Attempted Sun RPC high port access	[**]	205.188.153.105:4000->10.1.220.78:32771

SnortSnarf brought to you courtesy of Silicon Defense
Authors: [Jon Hoagland](#) and [Stuart Stanford](#)
See also the [Snort Page](#) by Marty Roesch
Page generated at Thu Feb 15 11:06:29 2001

Note the links to the Whois and DNS lookup sites. In this example, we clicked on TRIUMF and the following page resulted:

IP Address Identifier : 205.188.153.105 : 205.188.153.105

Another IP :

Domain Name Service (DNS) says 205.188.153.105 is fes-d009.icq.aol.com

Hostmaster (entity providing name service) is hostmaster@aol.net

[Check SMTP](#) (see if a mailserver, may reveal domain) [Check HTTP](#) (open in browser)

(TRIUMF only) [Check database](#) for a TRIUMF remote user.

[Check ip address in anti-spam databases & RBL](#)

Check [NetworkSolutions](#) for a .com, .net, .edu or .org domain. (Note: other registries now exist).

"Administrative contact" is usually responsible for the machine or webserver, while "Technical Contact" is usually responsible for the network connection and nameserver.

Check [Whois aol.com at whois.nic.com](#)

Netblock looks like ARIN

Check [ARIN](#), [RIPE](#), [APNIC](#), [AUNIC](#), [KRNIC](#), [TWNIC](#)

which should reveal who is responsible for the network connection. Follow links to best matching netblock.

Try an [RWHOIS](#) query ([rwhois.net](#)); [Web form here](#).

Try [Geektools whois proxy](#)

See Also

[Sam Spade](#) (lots of tools)

Try [checkdomain.com](#) on ip or [host](#)

See [Domain name registries around the world](#) for a very complete list.

[Contacting Host Owners](#) (SANS)

[Finding Site Contacts](#) (CERT)

Note the convenient links to determine if SMTP or HTTP is running on the particular examined host.

In this fashion, Snortsnarf was used to analyze activity of source and destination detect hosts, and look up source host information.. DNS lookups were most often obtained from [TRIUMF](#).

Network registration information was obtained from [ARIN](#) for US hosts, while [RIPE](#) was used for European hosts, and [APNIC](#) was used for Asia-Pacific hosts.

I soon realized what the "Segmentation faults" were all about. For some data, clicking on a particular source or destination IP would bring up a "page not found" error. It was apparent that Snortsnarf was unable to complete all the individual web pages for each particular alert IP.

To research particular detects, I most often used [Max Vision's Whitehats](#) site which I find one of the most comprehensive. It links quickly to CVE information, if available, about each alert.

To further explore what trojans and exploits are associated with what ports, a variety of sites were used:

<http://www.tlsecurity.com/trojanh.htm>

<http://www.moosoft.com/tdbindex.php>

<http://www.networkice.com/advice/Exploits/Ports/>

http://members.cotse.com/dlf/man/ports/ports0_500.htm

<http://www.henninger.net/downloads/ccna/tools/assigned-numbers.pdf>

To create Figure 1 (Assignment 2), a graphical representation of activity between selected hosts and ports, I examined “[The Brain](#)” which was used in [Lenny Zeltser's](#) excellent practical. I found it was not quite suitable for my purpose, so instead used...

In conclusion, this assignment made me realize my inadequacies when it comes to analyzing large amounts of inter-related data! Access and Excel were slow and limited in capabilities. I'm determined to learn Perl and MySQL. I suspect had I know them, this task would have been easier and been completed earlier!

© SANS Institute 2000 - 2005. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or by any information storage or retrieval system, without the prior written permission of SANS Institute.

References

Graham, R. 2000. "FAQ: Firewall Forensics (What am I seeing?)."
<http://www.robertgraham.com/pubs/firewall-seen.html>

Northcutt, S. 1999. *Network Intrusion Detection, An Analyst's Handbook*. New Riders Publishing, Indianapolis, Indiana.

Northcutt, S., Cooper, M., Fearnow, M., and Frederick, K. 2001. *Intrusion Signatures and Analysis*. New Riders Publishing, Indianapolis, Indiana.

SANS. 2001. "How to Eliminate the Ten Most Critical Internet Security Threats: The Experts Consensus." <http://www.sans.org/topten.htm>

Stevens, W. R. 1994. *TCP/IP Illustrated, Volume 1*. Addison-Wesley, Reading, Massachusetts.

Previous student practicals at: <http://www.sans.org/y2k/analysts.htm>

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