Global Information Assurance Certification Paper

SANS Intrusion Detection in Depth
GCIA Practical Assignment
Version 2.9
SANSFIRE, Washington D.C.
July 30 – August 4 2001

David B. Leach

Table of Contents

1. Assignment 1 – Network Detects

1.1 Detect #1 – Low and Slow Network Scan

The following network scan was initially detected starting on 28 March 2001 and continued until 11 April 2001. Over those 15 days, 141.213.10.159 attempted to scan 126 consecutive hosts on my network. Only samples of the entire scan are included to demonstrate the pattern.

29 March 2001

07:37:09: Deny inbound icmp src 141.213.10.159 dst MY.NET.131.16 (type 8, code 0)
07:37:17: Deny inbound icmp src 141.213.10.159 dst MY.NET.131.16 (type 8, code 0)
07:37:24: Deny inbound icmp src 141.213.10.159 dst MY.NET.131.16 (type 8, code 0)
07:41:22: Inbound TCP connection denied from 141.213.10.159/2380 to MY.NET.131.16/1884 flags SYN
07:41:25: Inbound TCP connection denied from 141.213.10.159/2380 to MY.NET.131.16/1884 flags SYN
07:41:31: Inbound TCP connection denied from 141.213.10.159/2380 to MY.NET.131.16/1884 flags SYN
07:41:43: Inbound TCP connection denied from 141.213.10.159/2380 to MY.NET.131.16/1884 flags SYN

5 April 2001

04:48:29: Deny inbound icmp src 141.213.10.159 dst MY.NET.131.83 (type 8, code 0)
04:48:37: Deny inbound icmp src 141.213.10.159 dst MY.NET.131.83 (type 8, code 0)
04:48:44: Deny inbound icmp src 141.213.10.159 dst MY.NET.131.83 (type 8, code 0)
04:54:19: Inbound TCP connection denied from 141.213.10.159/2331 to MY.NET.131.83/1884 flags SYN
04:54:22: Inbound TCP connection denied from 141.213.10.159/2331 to MY.NET.131.83/1884 flags SYN
04:54:28: Inbound TCP connection denied from 141.213.10.159/2331 to MY.NET.131.83/1884 flags SYN
04:54:40: Inbound TCP connection denied from 141.213.10.159/2331 to MY.NET.131.83/1884 flags SYN
05:01:15: Inbound TCP connection denied from 141.213.10.159/2518 to MY.NET.131.83/80 flags SYN
05:01:18: Inbound TCP connection denied from 141.213.10.159/2518 to MY.NET.131.83/80 flags SYN
05:01:24: Inbound TCP connection denied from 141.213.10.159/2518 to MY.NET.131.83/80 flags SYN
05:01:36: Inbound TCP connection denied from 141.213.10.159/2518 to MY.NET.131.83/80 flags SYN

06:31:35: Deny inbound icmp src 141.213.10.159 dst MY.NET.131.84 (type 8, code 0)
06:31:43: Deny inbound icmp src 141.213.10.159 dst MY.NET.131.84 (type 8, code 0)
06:31:50: Deny inbound icmp src 141.213.10.159 dst MY.NET.131.84 (type 8, code 0)
06:38:59: Inbound TCP connection denied from 141.213.10.159/4564 to MY.NET.131.84/1884 flags SYN
06:39:01: Inbound TCP connection denied from 141.213.10.159/4564 to MY.NET.131.84/1884 flags SYN
06:39:07: Inbound TCP connection denied from 141.213.10.159/4564 to MY.NET.131.84/1884 flags SYN
06:39:20: Inbound TCP connection denied from 141.213.10.159/4564 to MY.NET.131.84/1884 flags SYN
06:45:29: Inbound TCP connection denied from 141.213.10.159/4755 to MY.NET.131.84/80 flags SYN
06:45:32: Inbound TCP connection denied from 141.213.10.159/4755 to MY.NET.131.84/80 flags SYN
06:45:38: Inbound TCP connection denied from 141.213.10.159/4755 to MY.NET.131.84/80 flags SYN
06:45:50: Inbound TCP connection denied from 141.213.10.159/4755 to MY.NET.131.84/80 flags SYN

1.1.1 Source of Trace

My Organization's Network

1.1.2 Detect was Generated By

Cisco PIX Firewall version 5.3 with rule set that denies all but specific services on particular hosts. The firewall also blocks inbound and outbound ICMP traffic. The format of the PIX log records is: Date, Time, Message #, Message. However I have edited out the date and some other non-pertinent information for ease of reading. See Cisco's online PIX documentation for more details on reading PIX logs. Timestamps are GMT –5:00 (Eastern Standard Time.)

1.1.3 Probability the Source Address was Spoofed

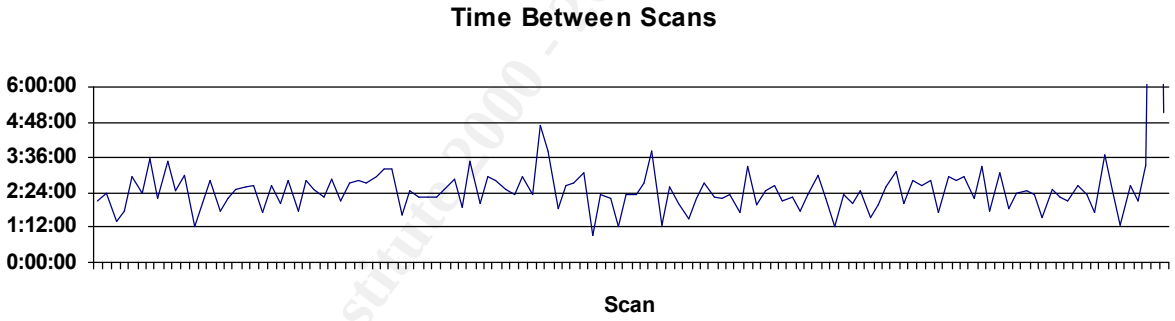
It is unlikely that the source address was spoofed. This appears to be an attempt at reconnaissance and spoofing the source address would not return information to the originator unless they could take advantage of source routing.

1.1.4 Description of Attack

This is appears to be an attempt to a) map hosts on my network, b) identify web servers and c) identify servers responding on TCP port 1884. There are many web server exploits; however, there is not enough information to determine exactly which vulnerability the attacker is attempting to find since the connection was denied by the firewall. According to IANA, port 1884 is used for Internet Distance Mapping Service (idmaps) and there are no vulnerabilities listed for this service in Mitre's CVE database nor in ISS's X-Force database. Additionally there are no known Trojans using port 1884 according to Simovit's Trojan List.

1.1.5 Attack Mechanism

The attacker first pings (ICMP Type 8, Code 0) expecting a response to see if the host exists then waits an average of 5 minutes before attempting a connection on TCP port 1884. Approximately 6 minutes later the attacker attempts a connection on TCP port 80. Finally, using Microsoft Excel to chart the times between each scan (see below), the attacker waits a seemingly random amount of time before attempting the scan on the next sequential host. The only exception was a 28 hour and 12 minute period where the scanning stopped completely which is the out-of-range spike on the graph.



Since there is a relatively long period between scans (1-5 hours), the attacker may be trying to avoid detection or could be busy scanning other hosts. Notice that the source port jumps 2,046 in the 1.5 hours between the 1st and 2nd iteration but only jumps 7 in the 3 hours between the 2nd and 3rd iteration. This may support the theory that the attacking host is scanning other networks in parallel with mine but I cannot draw any definite conclusions without further information.

Also note the pattern of four TCP SYN packets at 3, 6 and 12-second intervals to a particular host/port pair. This is consistent with TCP retries although I did not have access to the packets to further verify that the sequence numbers were the same for each retry.

Finally note that when my public web server (MY.NET.131.16) was scanned on 29 March 2001 the attempted TCP connection on port 80 was allowed by the firewall however there was no corresponding log entry in my web server. Since my web server is set to log all activity, I

assume that the TCP 3-way handshake was not completed but I cannot verify this since I had no IDS or packet sniffer available on that network segment at that time.

1.1.6 Correlations

141.213.10.159 was not found in the lists of attacking source IP addresses maintained by Dshield.org and Incidents.org and a check of the ARIN WHOIS Database showed that this IP address was registered to the University of Michigan's Computer Aided Engineering Network (CAEN). After contacting the CAEN system administrator listed in the ARIN database, I was informed that the scans are the result of a research project being conducted by CAEN to develop a method for a host to quickly determine the distance to another host (see http://idmaps.eecs.umich.edu/index.php.) I should also note that the CAEN administrator was nice enough to stop the scan against my network when I requested it.

1.1.7 Evidence of Active Targeting

There is some evidence of active targeting in that the attacker is scanning a particular subnet on my network.

1.1.8 Severity

Criticality (C)	3	Attacker is targeting a range of hosts on my DMZ.
Lethality (L)	2	Does not appear to be after a specific vulnerability.
System Countermeasures (S)	5	Web servers have latest security patches.
Network Countermeasures (N)	5	Perimeter firewall blocks inbound ICMP and port 1884. Port 80 traffic is only allowed for specific web servers.
Severity = (C+L)-(S+N)	**-5**	

1.1.9 Defensive Recommendation

Since the perimeter firewall blocks all traffic except to the legitimate web servers on port 80 and the web servers have the latest security patches implemented, no additional defenses are recommended.

1.1.10 Multiple Choice Test Question

Given the following firewall log sample, what is the best evidence that these are TCP retries?

```
06:38:59: Inbound TCP connection denied from 141.213.10.159/4564 to MY.NET.131.84/1884 flags SYN
06:39:01: Inbound TCP connection denied from 141.213.10.159/4564 to MY.NET.131.84/1884 flags SYN
06:39:07: Inbound TCP connection denied from 141.213.10.159/4564 to MY.NET.131.84/1884 flags SYN
06:39:20: Inbound TCP connection denied from 141.213.10.159/4564 to MY.NET.131.84/1884 flags SYN
```

a) Multiple attempts to the same source and destination port.
b) Multiple attempts at 3, 6 and 12-second intervals.
c) Multiple packets with SYN flag set.
d) All of the above.

Answer: D

1.2 Detect #2 – Third Party Effect

The following unsolicited ICMP Host Unreachable messages were detected on my perimeter firewall starting at 13:05 on 4 June 2001 and continuing through 23:19 on 20 June 2001. This is only a sample of the log entries. The fact that these were being "returned" to random existent and non-existent host IP addresses (including broadcast addresses) in my network prompted me to look deeper.

Jun 04 2001 09:19:33 Deny inbound icmp src 157.130.64.54 dst 161.11.131.1 (**type 3, code 13**)

< Additional entries deleted >

Jun 04 2001 15:11:42: Deny inbound icmp src 157.130.215.21 dst MY.NET.131.59 (**type 3, code 1**)
Jun 04 2001 15:11:44: Deny inbound icmp src 157.130.215.21 dst MY.NET.131.114 (type 3, code 1)
Jun 04 2001 15:12:44: Deny inbound icmp src 157.130.215.21 dst MY.NET.131.60 (type 3, code 1)
Jun 04 2001 15:17:27: Deny inbound icmp src 157.130.215.21 dst MY.NET.131.107 (type 3, code 1)
Jun 04 2001 15:21:21: Dst IP is network/broadcast IP, translation creation failed for icmp src 157.130.52.209 dst MY.NET.131.127 (type 3, code 1)
Jun 04 2001 15:21:21: Deny inbound icmp src 157.130.52.209 dst MY.NET.131.127 (type 3, code 1)
Jun 04 2001 15:21:34: Deny inbound icmp src 157.130.52.209 dst MY.NET.131.114 (type 3, code 1)
Jun 04 2001 16:00:55: Deny inbound icmp src 157.130.215.21 dst MY.NET.131.93 (type 3, code 1)
Jun 04 2001 16:01:21: Deny inbound icmp src 157.130.215.21 dst MY.NET.131.10 (type 3, code 1)
Jun 04 2001 16:10:37: Deny inbound icmp src 157.130.215.21 dst MY.NET.131.95 (type 3, code 1)
Jun 04 2001 16:26:59: Deny inbound icmp src 157.130.215.21 dst MY.NET.131.26 (type 3, code 1)
Jun 04 2001 16:28:17: Deny inbound icmp src 157.130.215.21 dst MY.NET.131.63 (type 3, code 1)
Jun 04 2001 16:30:34: Deny inbound icmp src 157.130.52.209 dst MY.NET.131.34 (type 3, code 1)
Jun 04 2001 16:35:50: Deny inbound icmp src 157.130.215.21 dst MY.NET.131.82 (type 3, code 1)
Jun 04 2001 16:38:44: Deny inbound icmp src 157.130.215.21 dst MY.NET.131.26 (type 3, code 1)
Jun 04 2001 16:38:56: Deny inbound icmp src 157.130.215.21 dst MY.NET.131.65 (type 3, code 1)
Jun 04 2001 16:44:34: Deny inbound icmp src 157.130.215.21 dst MY.NET.131.8 (type 3, code 1)
Jun 04 2001 16:50:26: Dst IP is network/broadcast IP, translation creation failed for icmp src 157.130.215.21 dst MY.NET.131.127 (type 3, code 1)
Jun 04 2001 16:50:26: Deny inbound icmp src 157.130.215.21 dst MY.NET.131.127 (type 3, code 1)

The occurrences of these packets were as follows:

Date	Source IP	# Packets
June 4	157.130.215.21	100
	157.130.52.209	16
	157.130.241.17	2
	157.130.182.213	4
	157.130.243.149	4
June 5	157.130.215.21	208
	157.130.52.209	70
	157.130.182.213	17
June 6	157.130.215.21	21
	157.130.52.209	23
	157.130.182.213	7
June 15	157.130.215.21	1
June 20	157.130.52.209	1

The following is the result of a WHOIS query at ARIN for the above source IP addresses:

> UUNET Technologies, Inc. (NET-UUNETCUSTB40)
> 3060 Williams Drive
> Fairfax, VA 22031
> US
>
> Netname: UUNETCUSTB40
> Netblock: 157.130.0.0 - 157.130.255.255

1.2.1 Source of Trace
My Organization's Network

1.2.2 Detect was Generated By
Cisco PIX Firewall version 5.3 with rule set that denies all inbound and outbound ICMP traffic. The format of the PIX log records is: Date, Time, Message #, Message. However I have edited out some non-pertinent information for ease of reading. See Cisco's online PIX documentation for more details on reading PIX logs. Timestamps are GMT –5:00 (Eastern Standard Time)

1.2.3 Probability the Source Address was is Spoofed
If the intent was a denial of service attack against my network, then these packets could contain spoofed source addresses. However, at a maximum of 25 packets per hour it is unlikely my network is the target. Therefore I believe these packets are 3rd party effect in which case the attacker is spoofing source addresses to UUNET but the responses to me are probably not spoofed.

1.2.4 Description of Attack
It appears that an unknown party may be attempting a denial of service attack (DOS) against UUNET's network and trying to cover their tracks by crafting packets with spoofed addresses from my network. Alternatively the attacker could be doing reconnaissance and intermingling spoofed packets with their own valid packets to avoid detection. Unfortunately I cannot determine the type of attack without more information such as the packets UUNET was seeing. I may have been able to determine the original protocol from the ICMP packets being returned to my network since they should contain the original IP header plus 64 bits of source data (Postel, RFC792, pp. 4-5) but I had no means to capture packets outside the firewall.

However, I suspect that the packets being sent to UUNET were ICMP. If they were TCP or UDP packets, I would expect to see TCP or UDP response packets from UUNET and I do not. So why do I only see ICMP Host Unreachable (and one "Communication administratively prohibited by filtering") packets and not any other ICMP traffic? A possible answer is that the router on that particular subnet may be a CISCO router that has an ACL implemented that blocks outbound ICMP but the router does not have "no ip unreachables" set. According to Cisco's documentation on Configuring Access Control Lists:

> **Caution** By default, the router sends Internet Control Message Protocol (ICMP) unreachables when a packet is denied by an access group; these access-group denied packets are not dropped in the hardware but are bridged to the router so that it can generate the ICMP-unreachable message. To drop access-group denied packets in the hardware, you must disable ICMP unreachables using the **no ip unreachables** interface configuration command. Note that the **ip unreachables** command is enabled by default.

1.2.5 Attack Mechanism

Third Party Effects are the result of an attacker (the source) sending packets to a victim (the destination) containing source IP addresses belonging to a third party. The scenario is as follows:

a) Attacker crafts packet with source IP of MY.NET.x.x
b) Attacker sends packet to victim
c) Victim processess packet and sends response to MY.NET.x.x
d) MY.NET.x.x receives packet and tries to process it

The reasons for doing this can be to mask reconnaissance or to perform a DOS attack (Bejtlich, p. 2). Typically a DOS attack involves very heavy traffic, which I am not seeing. However, since a partner organization is also seeing similar traffic (see 1.2.6 Correlations), it is likely the attacker is spoofing addresses from many other networks so we are only seeing a small portion of it.

Since the attacker will not see the responses to his/her crafted packets, they can choose any random IP addresses to spoof. In addition, the attacker may first use a utility that will check for non-existent hosts to use for spoofing which will make some DOS attacks like SYN floods more effective (Bejtlich, p. 5). Therefore it is possible that my firewall has actually logged a "host sweep" from the actual attacker but this is nearly impossible to correlate with the actual attack.

1.2.6 Correlations

A check of our Shadow sensor showed that there were no traffic between my network and the 157.130.0.0 network. In addition, I contacted an associate at a partner organization who saw the following, unsolicited ICMP Type 3, Code 1 messages from UUNET coming into his network. The common source addresses are bolded.

Date	Source IP	# Packets
June 4	157.130.52.209	18
	157.130.182.213	5
	157.130.215.21	90
June 5	**157.130.52.209**	54
	157.130.182.213	18
	157.130.215.21	173
June 8	157.130.143.153	10
June 15	**157.130.52.209**	3

The anomalous traffic was reported to UUNET who responded that this was normal traffic however it ended shortly there after.

1.2.7 Evidence of Active Targeting

There is no evidence that my network is being actively targeted.

1.2.8 Severity

Criticality (C)	3	Appears to be random targets on DMZ
Lethality (L)	0	ICMP error messages do not cause a response and there are not enough to cause degradation in service.
System Countermeasures (S)	4	All systems have current security patches applied.
Network Countermeasures (N)	5	Firewall blocks inbound ICMP traffic
Severity = (C+L)-(S+N)	**-6**	

1.2.9 Defensive Recommendation

No additional defense is needed since my firewall is already configured to block ICMP traffic.

1.2.10 Multiple Choice Test Question

Which of the following is a key indication of a Third Party Effect?

a) Spoofed destination address
b) Spoofed source address
c) ICMP Host Unreachable messages
d) ICMP Echo Reply messages

Answer: B

1.3 Detect #3 – Web Server Sending to Broadcast Address?

The following traffic was detected on 16 June 2001. In addition to the IP address shown below there were three other IP addresses that exhibited the same behavior: 24.29.57.0, 4.54.48.255 and 62.10.156.0

```
MY.NET.131.16 > 66.108.49.0
17:26:24.767254 MY.NET.131.16.80 > 66.108.49.0.1450: S 4027036726:4027036726(0) ack 4677322
                win 6144  (DF)
17:26:24.826189 MY.NET.131.16.80 > 66.108.49.0.1450: . 4027036727:4027038107(1380) ack 4677587
                win 5879 (DF)
17:26:24.826299 MY.NET.131.16.80 > 66.108.49.0.1450: P 4027038107:4027039400(1293) ack
                4677587 win 5879 (DF)

        < Additional entries deleted >

17:26:31.586660 MY.NET.131.16.80 > 66.108.49.0.1450: . 4027177984:4027179364(1380) ack 4687513
                win 31905 (DF)
17:26:31.586777 MY.NET.131.16.80 > 66.108.49.0.1450: . 4027179364:4027180744(1380) ack 4687513
                win 31905 (DF)
17:26:31.639280 MY.NET.131.16.80 > 66.108.49.0.1450: . 4027180744:4027182124(1380) ack 4687513
                win 31905 (DF)
17:26:31.639339 MY.NET.131.16.80 > 66.108.49.0.1450: FP 4027182124:4027182777(653) ack
                4687513 win 31905 (DF)
17:26:31.698661 MY.NET.131.16.80 > 66.108.49.0.1450: . ack 4687514 win 31904 (DF)
```

The following is the edited results of a WHOIS query at ARIN for the above source IP addresses:

```
24.29.57.0          ServiceCo LLC - Road Runner (NET-ROAD-RUNNER-1)
                    13241 Woodland Park Road
                    Herndon, VA 20171
                    US
                    Netblock: 24.24.0.0 - 24.30.95.255

4.54.48.255 BBN Planet (NET-SATNET)
                    150 Cambridge Park Dr.
                    Cambridge, MA 02138
                    US
                    Netblock: 4.0.0.0 - 4.255.255.255

66.108.49.0         ROADRUNNER-NYC (NETBLK-ROADRUNNER-NYC-1)
                    13241 Woodland Park Road
                    Herndon, VA 20171
                    US
                    Netblock: 66.108.0.0 - 66.108.255.255
```

The following is the edited results of a WHOIS query at RIPE for the above source IP address:

```
62.10.156.0         inetnum:    62.10.0.0 - 62.11.255.255
                    descr:      Tiscali SpA
                    descr:      PROVIDER
                    country:    IT
                    address:    Piazza del carmine, 22
                    address:    09124 Cagliari
```

address: Italy

1.3.1 Source of Trace
My Organization's Network

1.3.2 Detect was Generated By
Shadow v1.6 Intrusion Detection System hourly report using the default filters. The particular Shadow sensor that generated this alert is inside my perimeter firewall. Timestamps are GMT – 5:00 (Eastern Standard Time).

1.3.3 Probability the Source Address was Spoofed
Since there appears to be a successful TCP connection established and data is flowing in both directions (see Correlations), the source address is probably not spoofed.

1.3.4 Description of Attack
Shadow reported the above because it appeared the destination was a broadcast address. However this appears to be a false positive.

1.3.5 Attack Mechanism
Shadow's default IP filters cause an alert if the last octet of the destination IP address is "255" (broadcast) or "0" (older BSD broadcast) which is what caused the above traffic to show up in the hourly reports. Since TCP is a connection-oriented protocol, broadcast addresses do not make sense (Stevens, p. 169) so I believe the source addresses are actually legitimate host addresses from a network using subnetting that results in more than 255 hosts.

Note that all the source IP addresses belong to Internet Service Providers (ISP) that offer broadband Internet service. Since many broadband ISPs allocate 3-5 addresses per home user (i.e. customer) it is conceivable that a suburban community of more than 50 homes would require more than 255 host addresses on a single subnet. This could be accomplished with a subnet mask such as 0xfffffe00 that would provide up to 510 hosts (e.g. 66.108.48.0 through 66.108.49.254).

1.3.6 Correlations
Looking at the TCPDUMP files that the Shadow sensor collected I found that in all cases the normal TCP 3-way handshake was completed and, at the end of the session, the connection was terminated gracefully as shown in the example below:

```
17:26:24.642347 66.108.49.0.1449 > MY.NET.131.16.80: S 4677195:4677195(0) win 8192 <mss
                1380,nop,nop,sackOK> (DF)
17:26:24.642462 MY.NET.131.16.80 > 66.108.49.0.1449: S 4018602916:4018602916(0) ack 4677196
                win 6144 <mss 1380> (DF)
17:26:24.680161 66.108.49.0.1449 > MY.NET.131.16.80: . ack 1 win 8280 (DF)

                < Entries showing data flowing in both directions deleted >

17:32:55.102601 MY.NET.131.16.80 > 66.108.49.0.1472: FP 18319:18493(174) ack 520 win 5625 (DF)
17:32:55.175610 66.108.49.0.1472 > MY.NET.131.16.80: . ack 18494 win 8106 (DF)
17:32:55.206480 66.108.49.0.1472 > MY.NET.131.16.80: F 520:520(0) ack 18494 win 8106 (DF)
17:32:55.206548 MY.NET.131.16.80 > 66.108.49.0.1472: . ack 521 win 5624 (DF)
```

Additionally, a check of the web server logs showed normal web server traffic. There was no evidence of attempts to find or exploit any potential vulnerability.

1.3.7 Evidence of Active Targeting

This traffic was targeted to a specific host on my network but there was no indication of malicious intent.

1.3.8 Severity

Criticality (C)	4	Traffic was addressed to specific host
Lethality (L)	0	No attack detected
System Countermeasures (S)	5	Specific host had current security patches
Network Countermeasures (N)	2	The firewall allows traffic to port 80 on this host
Severity = (C+L)-(S+N)	**-3**	

1.3.9 Defensive Recommendation

Since this turned out to be a false positive, no additional defenses are recommended. However, to reduce the chances of future false positives, the Shadow IP filters could be changed as follows:

```
Old:    ip[19] = 0xff
        ip[19] = 0x00

New:    (ip[19] = 0xff) and (ip[9] != 6)
        ip[19] = 0x00 and (ip[9] != 6)
```

1.3.10 Multiple Choice Test Question

Which of the following subnet mask would result in more than 255 host IP addresses?

a) 0xffffff00
b) 0xfffffe00
c) 0xfffff000
d) Both A and B
e) Both B and C

Answer: E

1.4 Detect #4 – IANA Reserved Addresses

The following traffic was detected on 14 August 2001 inside my perimeter firewall. It was directed at one of my public web servers. Note that only the packets of interest are shown below:

10:58:08.905209 **106.1.0.0**.40194 > MY.NET.131.12.80: S 2431636879:2431636879(0) win 512
10:58:09.146763 **98.1.0.0**.40194 > MY.NET.131.12.80: S 897741404:897741404(0) win 512
11:00:49.403871 **2.0.0.0**.16324 > MY.NET.131.11.80: S 766581951:766581951(0) win 512
11:03:06.520078 **2.1.0.0**.16898 > MY.NET.131.11.80: S 1864106716:1864106716(0) win 512
11:03:12.262564 **1.0.0.0**.51201 > MY.NET.131.11.80: S 2933329890:2933329890(0) win 512
14:36:56.965845 **1.0.0.0**.32271 > MY.NET.131.12.80: S 2303974715:2303974715(0) win 512

1.4.1 Source of Trace

My Organization's Network

1.4.2 Detect was Generated By

Detected by Shadow v1.6 Intrusion Detection System hourly reports using the default filters. Timestamps are GMT –5:00 (Eastern Standard Time).

1.4.3 Probability the Source Address was Spoofed

The source IP addresses are probably spoofed since they are IANA Reserved address. Also note the same source port (40194) used from two different source addresses. What's the chance of that happening naturally?

1.4.4 Description of Attack

This could be an attempt at reconnaissance using techniques described by Tom Chmeilarski (http://www.sans.org/newlook/resources/IDFAQ/spoofed_IP.htm) and Kevin Van Dixon (http://www.sans.org/infosecFAQ/intrusion/spoof.htm). However in both those cases the spoofed address is of a live host (called a sensor) which is not the case here. Alternatively this may be an attempt at a TCP SYN flood DOS attack against my web server but there are not enough packets to sustain the attack.

1.4.5 Attack Mechanism

The attacker successfully initiated the first half of a TCP connection since in all cases my web server responded with a SYN/ACK. However the final ACK from the three-way handshake was not received, causing the web server to wait for a specific period. If a sufficient number of half-open connections were attempted (as little as 10 per minute) this attack could have prevented my web server from responding to any new requests.

1.4.6 Correlations

A check of the perimeter firewall logs showed there was no recognizable activity that may have coincided with the above packets. In taking a more detailed look at the packets collected by the Shadow sensor, using tcpdump (with the "-v" option) I found that all the spoofed packets had the same Time To Live (TTL). This is an indication that they probably originated from the same source.

10:58:08.905209 106.1.0.0.**40194** > MY.NET.131.12.80: S 2431636879:2431636879(0) win 512 <mss 1380> (**ttl 51**, id 40904)

10:58:09.146763 98.1.0.0.**40194** > MY.NET.131.12.80: S 897741404:897741404(0) win 512 <mss 1380> (**ttl 51**, id 40929)
11:00:49.403871 2.0.0.0.16324 > MY.NET.131.11.80: S 766581951:766581951(0) win 512 <mss 1380> (**ttl 51**, id 46604)
11:03:06.520078 2.1.0.0.16898 > MY.NET.131.11.80: S 1864106716:1864106716(0) win 512 <mss 1380> (**ttl 51**, id 53288)
11:03:12.262564 1.0.0.0.51201 > MY.NET.131.11.80: S 2933329890:2933329890(0) win 512 <mss 1380> (**ttl 51**, id 53405)
14:36:56.965845 1.0.0.0.32271 > MY.NET.131.12.80: S 2303974715:2303974715(0) win 512 <mss 1380> (**ttl 51**, id 42004)

Thinking that I may be able to correlate this back to a live host, I reran tcpdump against the same Shadow file looking for all packets that contained a TTL of 51. After running the results through UniqueIP.pl (see Appendix B) I found 29 different possibilities. Of those, three showed up on the Incidents.org database but the traffic on my network from them looked legitimate.

Finally, note that this is similar to Detect #4 in Graham Stork's practical (http://www.sans.org/y2k/practical/graham_stork_GCIA.doc) except for different addresses and less volume.

1.4.7 Evidence of Active Targeting

Given the spoofed address and specific host destination, there is strong evidence of active targeting.

1.4.8 Severity

Criticality (C)	4	Traffic addressed specifically to my web server
Lethality (L)	3	Unknown what exploit since connection failed.
System Countermeasures (S)	4	Web server has latest security patches.
Network Countermeasures (N)	0	Firewall allows traffic from IANA reserved addresses.
Severity = (C+L)-(S+N)	**3**	

1.4.9 Defensive Recommendation

Reconfigure perimeter firewall to block inbound packets from IANA reserved addresses as recommended in the SANS Top Ten List Appendix B, #1.

1.4.10 Multiple Choice Test Question

Assuming you are receiving the following packet once every 6 seconds, what is the most likely intent?

11:00:49.403871 **2.0.0.0**.16324 > MY.NET.131.11.80: S 766581951:766581951(0) win 512

a) Smurf attack
b) Packet craft.
c) Web server exploit.
d) SYN flood

Answer: D

1.5 Detect #5 – Denial of Service Attack

The following traffic was detected on 17 September 2001 starting at 08:45 and continuing through 10:14. This is only a small sample of the traffic with the source addresses obfuscated since they are probably spoofed.

10:00:28 10.66.161.130 %PIX-2-106001: Inbound TCP connection denied from **a.b.135.1/3127** to **MY.NET.131.16/1959** flags SYN
10:00:29 10.66.161.130 %PIX-2-106001: Inbound TCP connection denied from **c.d.168.180/22840** to **MY.NET.131.16/1959** flags SYN
10:00:29 10.66.161.130 %PIX-2-106001: Inbound TCP connection denied from **e.f.208.11/23453** to **MY.NET.131.16**/1959 flags SYN

1.5.1 Source of Trace

My Organization's Network

1.5.2 Detect was Generated By

Cisco PIX Firewall version 5.3 with rule set that denies all inbound except to specific services on specific hosts. The format of the PIX log records is: Date, Time, Message #, Message. However I have edited out some non-pertinent information for ease of reading. See Cisco's online PIX documentation for more details on reading PIX logs. Timestamps are GMT –5:00 (Eastern Standard Time)

1.5.3 Probability the Source Address was Spoofed

Typically with SYN flood attacks the source address is spoofed.

1.5.4 Description of Attack

1,607 TCP SYN packets were sent to my web-caching server (MY.NET.131.16) from 230 unique sources during a period of 89 minutes. This appears to be a distributed denial of service (DDOS) attack. Since the resulting traffic of approximately 18 packets per second is not enough to cause traffic congestion on my network, this may be a SYN flood attack however I did not have the ability to capture packets outside my firewall to assist in verification.

1.5.5 Attack Mechanism

A SYN flood is a denial of service attack that causes the target host to exhaust its resources thus preventing it from handling legitimate traffic. The attack works as follows:

- The attacker sends TCP packets to the victim with the SYN flag set which is the start of the 3-way handshake. Since the attacker does not intend to complete the connection, (s)he crafts the packet with a spoofed source IP address to cover their tracks.
- The victim responds with a SYN/ACK and keeps track of the half-open connection in memory while waiting for an ACK from the attacker. If a response is not received in a specific amount of time, the victim will release the resources to be reused for other connections. So….
- The attacker continues to send crafted TCP SYN packets at a rate as low as 10 per second.

1.5.6 Correlations

Port 1959 is used by the administration server for Novell's BorderManager Proxy Caching software (see http://support.novell.com/cgi-bin/search/searchtid.cgi?/2953420.htm).

CVE-2001-0486 – Remote attackers can cause a denial of service in Novell BorderManager using TCP flood.
CVE-2000-0152 – Remote attackers can cause denial of service in Novell BorderManager.

Richard Bejtlich's paper on Third Party Effects describes how SYN flood attacks work (http://packetstormsecurity.org/papers/evaluation/nid_3pe_v101.pdf).

CERT posted an advisory on TCP SYN Flooding and IP Spoofing Attacks at http://www.cert.org/advisories/CA-1996-21.html.

An example of a SYN flood was reported to Incidents.org by Jay Swofford at http://www.incidents.org/archives/y2k/080500.htm.

1.5.7 Evidence of Active Targeting

All packets were addressed to a specific port on a specific host so there is strong evidence of active targeting. Furthermore, since the target host is running Novell caching software, it appears that the attacker may have done some prior reconnaissance.

1.5.8 Severity

Criticality (C)	5	The target is my web server
Lethality (L)	4	The attacker appeared to know this was a Novell host and there are known DOS attacks against BorderManager
System Countermeasures (S)	4	The host has the latest security patches applied.
Network Countermeasures (N)	5	The firewall blocks access to port 1959 on this host.
Severity = (C+L)-(S+N)	0	

1.5.9 Defensive Recommendation

Since the firewall blocked access on that port to the host, no additional defenses are recommended.

1.5.10 Multiple Choice Test Question

What are the fewest packets per second (pps) that may sustain a TCP SYN flood against one target host?

a) 1-9 pps
b) 10-20 pps
c) 20-100 pps
d) 100-500 pps

Answer: B

2. Assignment 2 – Describe the State of Intrusion Detection

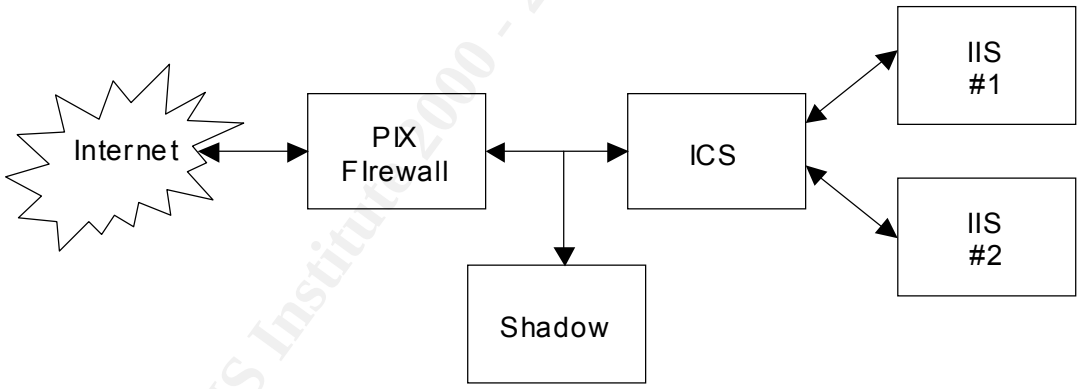
Using Log Files for Intrusion Detection: Challenges Caused by the Lack of Standards

Introduction

Correlating logs from firewalls, web servers and other devices plays a vital role in successfully detecting intrusions but the lack of standardized log formats is a major impediment. When I've asked other intrusion analysts what process and tools they use to process their log files, the common theme is that they a) time-synch their hosts and b) "grow their own" programs for analysis. The latter is typically due to the fact that there are very few products available that can correlate logs from different sources. Adopting standards such as those being developed by the Intrusion Detection Exchange Format Working Group (IDWG) will help accelerate development of log analysis products by addressing issues such as those that follow.

Time Synchronization: Whey NTP Is Not Enough

I regularly review logs from a Cisco PIX firewall, Novell (now Volera) Internet Caching Server (ICS), Microsoft IIS web server and Shadow Intrusion Detection System for signs of intrusions (using "home grown" scripts). While reviewing one of our IIS web server logs for 18 September 2001 I discovered malicious activity that required correlation with logs from other devices to determine the source. The following is a simplified view of the configuration involved:



The first problem I ran into was the source IP address in the IIS log was that of the ICS host. ICS is a web appliance that off-loads resources from web servers by caching static content. It can also be configured to load balance between multiple web servers [1]. Incoming web server requests are directed to the ICS, which changes the source address to itself before passing the request off to one of the back-end web servers. After the back-end web server processes the request, it sends the reply back to the ICS who in turn forwards it back to the originator.

The next step was to look at in the ICS logs using the timestamp from the IIS log. Both record their logs in World Wide Web Consortium (W3C) extended log file format, which specifies that timestamps are GMT [2]. Since neither of the IIS servers nor ICS clocks are synchronized automatically, I compared the system times on the ICS and IIS and found that

(a) both were set to Eastern Standard Time (i.e. GMT –5),
(b) both were set to adjust for Daylight Savings Time (i.e. subtract 1 hour the first Sunday in April and revert back on the last Sunday of October for most of the USA) and
(c) the web server clock (IIS #2) was exactly 7 minutes faster than the ICS clock.

However when I looked into the ICS logs at the appropriately adjusted time, I could not find a matching HTTP request.

After verifying that both the ICS and IIS #2 were logging all requests, I ran a test transaction that could be easily identified in both server logs and found that the log entries were an hour and 7 minutes apart! The IIS web server was correctly logging times in GMT but the ICS logs were not adjusted properly for Daylight Savings Time. Using this new piece of information I was now able to identify the external source IP address that originated the request.

Finally I wanted to see if there had been any other activity directed to any more of my hosts or services from that source by looking at my IDS and firewall logs. Fortunately both the Shadow hosts (analysis and sensors) and log host for the PIX firewall are automatically time-synced using Network Time Protocol (NTP) so this correlation is a little easier except that these logs are in local time so I had to make further time adjustments to correlate with other web server logs.

The point is a lot of time and effort is wasted just trying to match up timestamps between log files. Even if all my hosts were running NTP, I would still have to adjust for the fact that some products report local time while others report GMT.

No Standards for Log Files

As I mentioned previously, W3C has defined a standard format for web server logs, which is supported by most vendors. W3C extended format logs are easily imported into an SQL database for analysis so, assuming servers are time-synched, correlation between multiple web servers becomes easier.

Unfortunately, there is not a compatible format for firewall vendors to follow. Two of the most popular firewalls, Check Point Firewall-1 and Cisco PIX, use proprietary formats. Firewall-1's GUI allows the user to export the log as a text file that is easily imported into a SQL database facilitating analysis [3]. PIX, on the other hand, can be set up to send its logs to a log host via the Unix syslog facility so it is already in text format but the message text is free form and the format varies from message to message [4]. For example the following two messages relay the same information although the stimulus is slightly different:

```
00:06:50: %PIX-2-106001: Inbound TCP connection denied from NOT.MYNET.30.114/1979 to
                         MY.NET.131.24/80 flags SYN  on interface OUTSIDE
00:07:23: %PIX-3-106010: Deny inbound tcp src OUTSIDE:NOT.MYNET.177.28/1184 dst
                         DMZ:MY.NET.131.208/80
```

As a result, the variable, free form text makes parsing out information for import into a database much more difficult.

A couple of vendors have developed standards for handling firewall logs but their purpose seems to be to allow other vendor's products to interface to them. For example, WebTrends has developed the WebTrends Enhanced Log Format (WELF) that allows use of WebTrends's reporting tool with firewall logs including Firewall-1 and PIX [5]. Additionally, Check Point has developed the Open Platform for Security (OPSEC) SDK that allows other vendor products to be managed by Check Point's management interface [6]. Although a step in the right direction, they are not enough for use in intrusion detection.

A Possible Solution

The Internet Engineering Task Force's (IETF) Intrusion Detection Exchange Format Working Group (IDWG) was formed to develop standards for exchange of data between different intrusion detection systems [7]. Their draft standard, called the Intrusion Detection Exchange Format (IDMEF), proposes using a data model implemented in the Extensible Markup Language (XML) to facilitate analysis and correlation among various types of intrusion detection systems [8]. The advantage of using XML is that it has been gaining universal acceptance and support can be found in many programming languages. For example, ActiveState Perl v5.6 includes a built-in XML-Parser module for parsing XML documents (see www.activestate.com). And, as the name implies, using XML for implementing the data model allows it to be extended as new needs arise.

Although the IDMEF addresses a common format for intrusion detection systems to send alerts, it could also be used to provide a standard for log formats. First, it provides a standard for reporting timestamps that allows for discrepancies while making it easier to resolve them. The standard states that

> Times MUST be formatted to include (a) an indication that the time is in Coordinated Universal Time (UTC), or (b) and indication of the difference between the specified time and Coordinated Universal Time. [8]

Although GMT and UTC are basically the same, UTC is based on an atomic clock rather than astronomical measurements used by GMT [9]. Therefore UTC is more accurate and is the preferred standard. The IDMEF specifies that UTC timestamps be in the format: "hh:mm:ssZ" (or hh:mm:ss.ssZ where fractional seconds are required) whereas local timestamps use the format: "hh:mm:ss+hh:mm" representing local time ahead of UTC and "hh:mm:ss-hh:mm" representing local time behind UTC. For example, 3:15 p.m. Eastern Standard Time would be represented by either of the following:

"15:15:00-05:00" or "20:15:00Z"

Reporting timestamps using this format would allow analysts to spend less time normalizing timestamps between different types of logs.

Second, in its present form the IDMEF is already flexible enough to accommodate certain types of log messages. For example the first PIX message above (PIX-2-106001) could be represented using the following IDMEF notation:

```
<IDMEF-Message version="0.5">
    <Alert ident="PIX-2-106001">
        <Analyzer analyzerid="MyPix" manufacturer="Cisco" version="5.3">
            <Node category="unknown"
                <location>Main Network Room</location>
                <name>Internet Firewall</name>
            </Node>
        </Analyzer>
        <CreateTime ntpstamp="0xbf39539a.0x00000000">
            2001-08-31T00:06:50-04:00
        </CreateTime>
        <Source interface="OUTSIDE" spoofed="unknown">
            <Node>
                <Address category="ipv4-addr">
                    <address>NOT.MYNET.30.114</address>
                </Address>
            </Node>
            <Service>
                <port>1979</port>
                <protocol>TCP</protocol>
            </Service>
        </Source>
        <Target>
            <Node>
                <Address category="ipv4-addr">
                    <address>MY.NET.131.24</address>
                </Address>
            </Node>
            <Service>
                <name>http</name>
                <port>80</port>
            </Service>
        </Target>
        <Classification>
            <name>106001</name>
            <url>
                http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_v53/syslog/pixemsgs.htm#36183
            </url>
        </Classification>
        <AdditionalData type="string" meaning="TCP Flags">
            SYN
        </AdditionalData>
    </Alert>
</IDMEF-Message>
```

However there are other types of messages that will not easily fit into the existing classes. The IDWG recognized that they could not define all possible types of message classes so they built in the capability to extend the IDMEF model through the "AdditionalData" class. A simplistic use of the "AdditionalData" class is shown above. In its more complex form, the "AdditionalData" class can be used to define new XML document type definition (DTD) modules. These new modules can, in turn, be included in newer versions of the IDMEF standard as it evolves.

Summary

There are numerous examples of how standards have allowed inter-operability and inter-communication between disparate systems ultimately saving time. The goal of an intrusion detection analyst is to be able to determine whether a compromise has occurred as close to the time of the event as possible. If an intrusion is detected in a timely manner, we can hopefully limit the amount of damage. Since parsing through megabytes of log files in varying formats adds to that time, we need to quickly adopt and implement standards to reduce correlation time thereby reducing overall detection time.

References

[1] "Novell ICS 1.3 Administration Guide" URL: http://www.novell.com/documentation/lg/ics13/index.html (29 September 2001).

[2] Hallam-Baker, Phillip M. and Behlendorf, Brian. "W3C Extended Log File Format" URL: http://www.w3.org/TR/WD-logfile.html (29 September 2001).

[3] "Check Point Firewall-1 Technical Overview" October 2000. URL: http://www.checkpoint.com/products/downloads/fw1-4_1tech.pdf (29 September 2001).

[4] "Cisco PIX Firewall and VPN Configuration Guide Version 6.1" 2001. URL: http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_61/config/index.htm (29 September 2001).

[5] "WebTrends Enhanced Log Format (WELF) for Firewalls & VPNs" URL: http://www.webtrends.com/library/prtnr_welf.doc (29 September 2001).

[6] "OPSEC Alliance Info Sheet" 2001. URL: http://www.checkpoint.com/opsec/info.htm (1 October 2001).

[7] Wood, M. and Erlinger, M. "Intrusion Detection Message Exchange Requirements" 20 February 2001. URL: http://www.silicondefense.com/idwg/draft-ietf-idwg-requirements-05.txt (1 October 2001).

[8] Curry, David A. and Debar, Herve. "Intrusion Detection Message Exchange Format Data Model and Extensible Markup Language (XML) Document Type Definition" 18 September 2001. URL: http://www.silicondefense.com/idwg/draft-ietf-idwg-idmef-xml-04.txt (1 October 2001).

[9] U.S. Naval Observatory. "What is Universal Time" 6 March 2001. URL: http://aa.usno.navy.mil/faq/docs/UT.html (6 October 2001).

[10] Windl, Ulrich, et al. "The NTP FAQ and HOWTO" 18 September 2001. URL: http://www.eecis.udel.edu/~ntp/ntpfaq/NTP-a-faq.htm (29 September 2001).

3. Assignment 3 – "Analyze This" Scenario

The following Snort data was analyzed to provide a security audit for a University. The Snort data was created using a fairly standard Snort rule set. Note that files with a ".B.gz" extension were excluded since they appeared to be duplicate files.

Date	File	# Alerts	Size (Bytes)
10 August 2001	alert.010810	10,537	1,111,902
	oos_Aug.10.2001	21	6,651
	scans.010810	19,796	1,346,756
11 August 2001	alert.010811	24,162	2,875,050
	oos_Aug.11.2001	74	21,652
	scans.010811	1,105,127	78,110,884
12 August 2001	alert.010812	26,855	1,907,644
	oos_Aug.12.2001	108	31,337
	scans.010812	37,173	2,544,944
13 August 2001	alert.010813	25,045	2,750,554
	oos_Aug.13.2001	247	70,725
	scans.010813	61,537	4,181,347
14 August 2001	alert.010814	57,426	5,552,366
	oos_Aug.14.2001	107	30,759
	scans.010814	71,809	5,083,172
15 August 2001	alert.010815	19,651	2,138,434
	oos_Aug.15.2001	108	32,498
	scans.010815	126,142	8,435,268

Top 10 Sources of Alerts

Count	IP	Registered To:	Source
65,583	207.155.118.220	PADS Development 1780 Oak Street #C Torrance, CA 90501 US	ARIN
42,995	205.188.246.121	America Online, Inc 22080 Pacific Blvd Sterling, VA 20166 US	ARIN
28,628	217.229.165.221	Deutsche Telekom AG, Internet service provider Am Kavalleriesand 3 D-64295 Darmstadt Germany	RIPE
16,107	4.33.128.228	BBN Planet 150 Cambridge Park Dr. Cambridge, MA 02138 US	ARIN
16,007	212.187.67.33	chello Broadband GmbH Internet Services Reumannplatz 7 A-1100 Vienna, Austria	RIPE

Count	IP	Registered To:	Source
14,858	166.90.181.9	Level 3 Communications, Inc 1025 Eldorado Blvd Broomfield, Colorado 80021 US	ARIN
13,816	213.26.139.140	UNLIMITED SOFTWARE SRL Centro direzionale Napoli isola E3 piano 12 int. 48 I-80143 Napoli (NA) Italy	RIPE
9,441	209.73.164.70	AltaVista Company 529 Bryant St. Palo Alto, CA 94301 US	ARIN
9,293	217.226.245.152	Deutsche Telekom AG, Internet service provider Deutsche Telekom AG Am Kavalleriesand 3 D-64295 Darmstadt Germany	RIPE
9,268	207.26.210.70	NetEnterprise 1088 Bishop St Honolulu, HI 96813 US	ARIN

Alerts Log Analysis

Snort Alert Data Summary (116,855 Alerts)

Signature	# Alerts	# Sources	# Destinations
External RPC call	54,645	38	33,984
SMB Name Wildcard	21,582	9,340	8,799
Connect to 515 from outside	13,226	14	11,540
Possible Trojan server activity	11,124	2051	6,609
UDP SRC and DST outside network	8,327	63	2,799
SNMP public access	1,542	40	161
Attempted Sun RPC high port access	1,237	116	5
WinGate 1080 Attempt	1,082	74	94
Watchlist 000222 NET NCFC	407	14	20
TCP SRC and DST outside network	139	21	75
STATDX UDP attack	138	19	107
Queso fingerprint	134	23	40
Connect to 515 from inside	77	7	7
Tiny Fragments – Possible Hostile Activity	83	51	67
High port 65535 tcp – possible Red Worm – traffic	69	20	20
NMAP TCP ping!	41	19	19
Port 55850 tcp – Possible myserver activity – ref. 010313-1	35	14	14
Null scan!	32	29	26
High port 65535 upd – possible Red Worm – traffic	29	19	20
SUNRPC highport access!	12	4	4
ICMP SRC and DST outside network	1	1	1
Probably NMAP fingerprint attempt	1	1	1
Samba client access	1	1	1
SYN-FIN scan!	1	1	1

External RPC call

Top 5 Source hosts

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
4.33.128.228	8003	8030	5001	5004
166.90.181.9	7348	7365	5834	5836
209.73.164.70	4745	4759	4060	4067
207.26.210.70	4440	4447	3952	3954
209.142.214.16	4399	4421	3499	3502

Top 5 destination hosts

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.100.165	22	288	1	35
MY.NET.99.196	20	20	4	4
MY.NET.98.70	19	23	6	9
MY.NET.99.132	17	17	5	5
MY.NET.99.238	17	17	5	5

This alarm is triggered by attempts to access the portmapper service (port 111) that could be exploited to gain root access on Unix and Linux systems. Several exploits are outlined in CERT's Incident Note IN-99-04 (http://www.cert.org/incident_notes/IN-99-04.html)
As of 9 September 2001, this is currently the third highest probed port according to Dshield.org's Top 10 Targeted Ports (http://www.dshield.org/topports.html).

Correlations: Detect #2 in Joseph Rach's practical (http://www.sans.org/y2k/practical/Joseph_Rach.html#DETECT2).

Recommended Defense: Disable the portmapper service on hosts that do not require it. For hosts that need to have the portmapper enabled, ensure that all appropriate patches have been applied and that you are only running the specific RPC services you need. SANS Top Ten List #3 (http://www.sans.org/topten.htm) has references on where to find patches from vendors. For hosts that require portmapper be enabled for internal-only applications, a packet filter or firewall should be configured to block port 111 access from unauthorized hosts. David Reece has written an excellent paper entitled "Is blocking port 111 sufficient to protect your systems from RPC attacks?" (http://www.sans.org/newlook/resources/IDFAQ/blocking.htm) which is helpful in configuring your packet filter.

SMB Name Wildcard

Top 5 Source Hosts

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
63.210.163.18	927	927	1	1
213.73.136.151	207	207	1	1
138.88.67.102	180	180	2	2
200.246.11.167	167	167	119	119
141.157.102.11	160	160	6	6

Top 5 Destination Hosts

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.182.91	1333	1366	293	302
MY.NET.253.114	636	639	21	23

MY.NET.69.216	700	750	521	527
MY.NET.100.165	338	374	52	60
MY.NET.253.114	217	221	16	18

This alarm is triggered by attempts to connect to the NETBIOS Name Service (port 137). In many cases this can be normal traffic but it can also be used by an attacker to obtain a list of Windows hosts on the network. This information can, in turn, be used for targeted attacks against specific hosts which can be identified by a subsequent attempt to connect via port 139 (http://www1.dshield.org/ports/port137.html) such as the following:

```
Aug 15 01:32:37 213.107.120.174:4648 -> MY.NET.2.1:139 SYN **S*****
Aug 15 01:32:45 213.107.120.174:4670 -> MY.NET.2.1:139 SYN **S*****
```

Note that the majority of this traffic originated from external addresses and, given that the top 5 sources listed all appear to be ISP's, it is possible that a lot of this traffic may be just home Windows users without firewalls.

Correlations: Detect # 2 from Eric Hacker's practical (http://www.sans.org/y2k/practical/Eric_Hacker.html - anchor9566546)

Defensive Recommendation: Although this traffic appears to be fairly innocuous, the number of false positives could be reduced by filtering port 137 (as well as port 139) traffic at a perimeter packet filter. File and print sharing should only be enabled where needed and locked down to specific directories that are protected with strong passwords. See the SANS Top Ten List #7 for more information (http://www.sans.org/topten.htm).

Connect to 515 from outside

Top 5 Source Hosts

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
212.187.67.33	8162	8162	7047	7047
210.96.22.193	4132	4132	3644	3644
204.210.243.163	867	867	787	787
208.177.252.181	40	40	37	37
255.255.255.255	19	19	19	19

Top 5 Destination Hosts

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.157.249	28	29	2	3
MY.NET.99.104	18	19	2	3
MY.NET.140.113	9	9	2	2
MY.NET.162.64	6	8	2	4
MY.NET.162.82	5	5	2	2

This alarm is triggered by attempts to connect to the Unix print spooler service (port 515) from an external source address. This service contains vulnerabilities that may allow an attacker to gain root access to the host. For example, CERT Advisory CA-2000-22 discusses a buffer overflow vulnerability in the LPRng server that is shipped with many Linux systems that could lead to a denial of printing service or root compromise of the host (http://www.cert.org/advisories/CA-2000-22.html). Also note there are a large number of probes to port 515 reported by SANS Incidents.org (http://www.incidents.org/).

Note the "broadcast" source host listed above. The 19 alerts all show the same source port 31337 to multiple destination addresses spread over the 6 days. This bears further investigation since there is some evidence of packet craft.

Correlations: Detect from Chris Talianek at http://www.sans.org/y2k/012201.htm

Defensive Recommendation: Disable the printer spooler service on hosts that do not require it. Also ensure that the latest patches from the appropriate vendors are applied. Port 515 should also be blocked in the perimeter packet filter or firewall.

Possible Trojan server activity

Top 5 Source Hosts

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
213.26.139.140	5677	5679	5083	5083
MY.NET.98.164	1235	1235	870	870
MY.NET.97.214	792	792	474	474
24.201.107.14	168	168	98	98
MY.NET.97.207	60	60	1	1

Top 5 Destination Hosts

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
213.26.139.140	2478	2478	1659	1659
MY.NET.97.214	221	222	149	150
MY.NET.98.164	221	227	163	167
213.8.227.24	60	60	1	1
24.249.179.176	58	58	1	1

This alarm was triggered because either the source or destination port was 27374, which is commonly used by the SubSeven Trojan. SubSeven is Windows remote control software that will allow an attacker to take over a victim's host without their knowledge. This should be investigated further since it appears that some of the university's hosts have been compromised because they appear to be sending as well as receiving packets to port 27374. Aaron Greenlee has written a paper on SubSeven that would be helpful in further tracking down the problem (http://www.sans.org/infosecFAQ/malicious/subseven_22.htm).

Correlation: Incidents.org reported seeing 1,000-3,000 SubSeven probes a day during the month of August from what appear to be mainly home users (http://www.incidents.org/diary/august2001.php - 303). SANS also describes Sub7 v2.2 in the Windows Security Digest Vol.4 No.3 (http://www.sans.org/newlook/digests/ntarchives/033101.htm)

Defensive Recommendation: First, ensure that all Windows users are running anti-virus software and have the latest virus signature files installed. Second, since one of the primary means for SubSeven to infect a host is via e-mail attachments, install and configure filtering software on e-mail servers to delete executable files. Also, once a SubSeven client is identified, the corresponding ISP should be notified so they can take appropriate action.

UDP SRC and DST outside network

Top 5 Source Hosts

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
64.210.135.86	2349	2349	6	6
192.168.11.127	1828	1828	2	2
169.254.180.216	1170	1170	812	813
10.0.0.2	1120	1120	801	801
169.254.5.245	1004	1006	800	800

Top 5 Destination Hosts

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
10.0.3.2	1849	1849	2	2
192.168.13.30	1828	1828	2	2
64.14.124.200	256	256	2	2
10.0.3.3	244	244	2	2
212.104.156.106	62	62	1	1

These alerts are triggered by UDP traffic where neither the neither source nor destination addresses are in the MY.NET.x.x network. Routers or other equipment that are not configured correctly can cause this. In these cases the source port was 137 and the destination ports were a combination of 137 and 53 so it may also be Windows systems that that are dual-homed to different networks. For example, someone who may have a "back-door" connection to the Internet while connected to the University network.

Defensive Recommendation: Configure egress filtering on the perimeter firewall to block outbound traffic from IANA reserved addresses and source addresses that are not from MY.NET.x.x.

SNMP public access

Top 5 Source hosts

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
24.180.132.123	513	513	6	6
24.180.202.45	305	305	93	93
63.79.238.154	261	261	2	2
64.205.198.196	185	185	2	2
64.244.202.66	176	176	3	3

Top 5 Destination Addresses

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.71.24	284	288	2	5
MY.NET.100.160	219	227	10	17
MY.NET.1.81	188	190	3	5
MY.NET.109.89	137	139	1	3
MY.NET.109.70	124	125	1	2

This alert is triggered when an SNMP request (port 161) is made with a password of "public" which is the default community string. This is one of the top 10 most common threats as indicated on the SANS Top Ten list (see http://www.sans.org/topten.htm #10). If successful, an attacker can use SNMP to gain valuable information about your network and potentially disable or reconfigure devices on your network.

Correlation: See Jose Luis Camacho's paper on SNMP architecture and security at http://www.sans.org/infosecFAQ/netdevices/SNMP_sec.htm. James Romanski's paper on "Using SNMP for Reconnaissance" describes how SNMP can be exploited using the default community strings (http://www.sans.org/newlook/resources/IDFAQ/SNMP.htm).

Defensive Recommendation: Disable external port 161 access in your perimeter firewall. Disable SNMP on hosts and devices that do not require it. Where SNMP is required, use strong passwords (see http://www.sans.org/topten.htm #8) and upgrade to the latest version of SNMP V3 which uses encrypted passwords for authentication.

Attempted Sun RPC high port access

Top 5 Source Hosts

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
205.188.153.99	839	839	1	1
128.59.25.66	249	249	1	1
205.188.153.97	17	17	2	2

208.62.89.170	4	4	1	1
63.150.73.7	4	4	1	1

All Destination Hosts

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.217.190	839	841	2	6
MY.NET.218.14	381	384	113	116
MY.NET.98.168	15	24	1	7
MY.NET.217.82	2	4	1	3

Packets to the portmapper alternative port (32771) trigger this alarm and the same vulnerabilities and defenses as outlined in "External RPC Call" apply. 69% of these alerts originated from AOL using port 4000, which is usually ICQ traffic. Unfortunately ICQ is also a popular mechanism for Trojans to "phone home" but none of this traffic appears to correlate to the traffic seen in "Possible Trojan Server Activity" above. In addition there are a number of known vulnerabilities with ICQ (see CVE-1999-0474, CVE-2000-0522, CAN-2000-1078 and CAN-2001-0367).

Another 20% originated from Columbia University using source port 28000, which is not a well-known port so I can not tell what this traffic may be without more information.

Correlation: The following are the results from ARIN for 205.188.153.x and 128.59.25.66"

```
America Online, Inc (NETBLK-AOL-DTC)
   22080 Pacific Blvd
   Sterling, VA 20166
   US

   Netname: AOL-DTC
   Netblock: 205.188.0.0 - 205.188.255.255

   Coordinator:
      America Online, Inc.  (AOL-NOC-ARIN)  domains@AOL.NET
      703-265-4670

   Domain System inverse mapping provided by:

   DNS-01.NS.AOL.COM             152.163.159.232
   DNS-02.NS.AOL.COM             205.188.157.232

   Record last updated on 27-Apr-1998.
   Database last updated on 10-Sep-2001 23:16:26 EDT.

Columbia University (NET-CU-NET)
   Center for Computing Activities  Watson Labs, 7th Floor  612 West
115th Street
   New York, NY 10025
   US
```

```
Netname: CU-NET
Netblock: 128.59.0.0 - 128.59.255.255

Coordinator:
   Columbia University Computer Operations  (CU-NOC-ARIN)
                net-trouble@columbia.edu
   (212) 854-1919

Domain System inverse mapping provided by:

SAELL.CC.COLUMBIA.EDU  128.59.59.218
DNS2.ITD.UMICH.EDU            141.211.125.15

Record last updated on 19-Mar-2001.
Database last updated on 10-Sep-2001 23:16:26 EDT.
```

Paul Asadoorian also discusses this same traffic from AOL in his practical at http://www.sans.org/y2k/practical/Paul_Asadoorian_GIAC.doc

Defensive Recommendation: In addition to the defenses recommended in "External RPC Call", the traffic from AOL should be investigated further to determine if it is legitimate ICQ or if those systems have been compromised. In particular, look for traffic from MY.NET hosts to AOL on port 4000. Since many people use ICQ for business and personal messaging, it may not be feasible to block its use especially in a University environment.

WinGate 1080 Attempt

Top 5 Source Hosts

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
217.229.165.221	934	1899	868	1393
200.194.96.32	27	27	2	2
217.10.143.54	10	10	5	5
65.162.249.12	10	10	1	1
63.102.226.86	9	9	5	5

Top 5 Destination Hosts

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.75.202	28	30	3	4
MY.NET.60.11	15	26	8	16
MY.NET.217.38	10	14	1	2
MY.NET.60.38	8	30	6	16
MY.NET.98.161	6	9	1	3

This alarm is triggered by attempts to connect to port 1080, which is used by the WinGate proxy server. Attackers will try to find open proxy servers to launch attacks in an attempt to cover their

tracks. Also note that 217.229.165.221 was attempting connections on port 111 to various hosts so it is very probable that this is malicious activity.

Correlation: A search of the Incidents.org CID database found a report on 7 July 2001 of another host in the 217.229.0.0 network that was scanning port 1080, port 111 and others including known Trojan ports (i.e. 12345). According to RIPE, 217.229.0.0 is registered to Deutsche Telekom AG in Germany.

Also see Detect #2 of John Best JR's practical (http://www.sans.org/y2k/practical/John_Best.htm - detect2) for an example of a WinGate scan.

Defensive Recommendation: Block all inbound connection attempts to port 1080 at your perimeter firewall so that people outside your network can not bounce traffic off your proxy server. You may also want to search lists of open proxy servers using any of the popular Internet search engines to see if any of the hosts in MY.NET are publicly listed and focus on blocking those first. Finally, Deutsche TeleKom should be apprised of this activity including the correlation found at Incidents.org.

Watchlist 000222 NET NCFC

Top 5 Source Hosts

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
159.226.41.166	323	323	1	1
159.226.185.107	23	33	6	7
159.226.228.1	18	18	2	2
159.226.21.3	11	11	3	3
159.226.128.8	9	9	1	1
159.226.48.3	6	6	1	1

Top 5 Destination Hosts

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.163.100	323	325	1	3
MY.NET.253.42	21	62	3	12
MY.NET.253.43	16	53	3	10
MY.NET.253.31	9	26	1	9
MY.NET.100.230	6	25	1	10

The alert is triggered by connection attempts from the 159.226.x.x network that is registered to The Computer Network Center Chinese Academy of Sciences in Beijing, China. There are numerous reports in the Incidents.org database of scans, including scans on known Trojan ports, originating from this network. Specifically, the second of the Top 5 Source Hosts above, 159.226.185.107, is reported 94 times from 08 August 2001 through 21 August 2001 (as of 15

September 2001). Note also that this address is only attempting connections on ports 139, 12345 and 27374 to various hosts indicating scans for Trojans.

Also note that the top source host, 159.226.41.166, is specifically targeting MY.NET.163.100 from source port 23 (telnet) to various destination ports 3478, 3486, 3516, 3652 and 4101 which are not listed as well-know ports.

Correlation: Crist Clark noted similar activity involving 159.226.228.1, 159.226.21.3 and 159.226.128.8 (see http://www.sans.org/y2k/practical/Crist_Clark_GCIA.html#analysis) and Chris Kuethe saw similar activity involving 159.226.41.166 although on different ports (see http://www.sans.org/y2k/practical/chris_kuethe_gcia.html#2.14).

Defensive Recommendation: Further investigation should be done on the traffic from 159.226.41.166 to MY.NET.163.100 to determine if this is legitimate traffic. Since the traffic from 159.226.185.107 appears to be of malicious intent, ensure that all anti-virus software is up-to-date and double-check that the destination hosts are not infected. Contact the Academy of Sciences and let them know of the traffic, particularly from 159.226.185.107. As a last resort, you may want to consider blocking connection attempts from this network at your perimeter firewall unless you do legitimate business with them.

TCP SRC and DST outside network

Top 5 Source Hosts

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
172.131.19.34	69	69	40	40
169.254.101.152	32	60	5	20
24.180.139.57	9	9	8	8
192.168.1.2	7	7	3	3
172.170.180.165	4	4	3	3

Top 5 Destination Hosts

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
216.155.60.12	29	29	2	2
172.18.1.92	4	4	1	1
24.29.150.80	3	3	1	1
24.141.208.81	3	3	1	1
64.12.28.125	3	1	1	1

See explanation and defensive recommendations from "UDP SRC and DST outside network" alerts. Note that some of the traffic appears to be from KaZaA (port 1214), IRC (port 6665) and AOL (5190).

STATDX UDP attack

Top 5 Source Hosts

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
4.33.128.228	27	8030	17	5004
209.142.214.16	22	4421	19	3502
166.90.181.9	17	7365	14	5836
209.73.164.70	14	4759	13	4067
148.235.152.179	13	2216	13	1824

Top 5 Destination Hosts

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.162.233	5	17	2	4
MY.NET.75.140	4	10	3	4
MY.NET.110.75	3	10	2	3
MY.NET.111.161	2	10	1	4
MY.NET.158.49	2	10	2	6

This alert indicates someone looking to exploit vulnerable Unix rpc.statd services. The destination hosts had all been previously probed on port 111. Note that 4.33.128.228 was also the top source host in "External RPC call" alerts above. This address is registered to:

BBN Planet (NET-SATNET)
150 Cambridge Park Dr.
Cambridge, MA 02138
US

Netname: SATNET
Netblock: 4.0.0.0 - 4.255.255.255
Maintainer: BBNP

Coordinator:
Soulia, Cindy (CS15-ARIN) csoulia@genuity.net
800-632-7638

Domain System inverse mapping provided by:

NIC.NEAR.NET	192.52.71.4
VIENNA1-DNS-AUTH1.BBNPLANET.COM	4.1.16.4
NIC3.BARRNET.NET	131.119.245.6

ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

Record last updated on 17-Feb-1999.
Database last updated on 14-Sep-2001 23:36:32 EDT.

Correlations: This address is reported 207 times from 19 July 2001 through 27 July 2001 on the Incidents.org database. In addition there are known exploits of rpc.statd (see references in "External RPC call" above.)

Defensive Recommendation: There is evidence that the destination hosts may be compromised so this should be investigated further. In addition, see the defensive recommendations discussed in the "External RPC call" alert analysis.

Queso fingerprint

Top 5 Source Hosts

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
199.183.24.194	86	86	3	3
130.207.193.70	12	12	5	5
209.10.41.242	11	11	3	3
64.192.140.5	5	5	4	4
158.252.44.223	4	4	3	3

Top 5 Destination Hosts

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.253.41	32	46	1	10
MY.NET.253.42	27	44	1	7
MY.NET.253.43	27	53	1	10
MY.NET.163.17	8	10	1	2
MY.NET.150.133	4	77	1	31

TCP packets with the SYN flag and reserved bits 1 and 2 (i.e. S12) trigger this alert. Queso is a reconnaissance tool used to determine which operating system is running on a particular host. This information, in turn, can be used to determine which exploits can be used to compromise this host.

Note that the majority of the traffic is from 199.183.24.194 sending to port 25 (SMTP) on the top 3 destination hosts listed above. Some of the destination ports include 1214 (KaZaA) and 6346 (Gnutella), which are peer-to-peer file sharing applications with known vulnerabilities.

Correlation: Detect # 2 in Paul Asadoorian's practical discusses Queso (http://www.sans.org/y2k/practical/Paul_Asadoorian_GIAC.doc).

Defensive Recommendation: Ensure that the latest OS and e-mail application patches are applied. Block access to hosts and ports in your firewall that should not be accessible from outside your network.

Connect to 515 from inside

All Source Hosts

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.1.2	76	76	2	2
MY.NET.5.119	1	1	1	1

All Destination Hosts

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.50.35	76	76	2	2
MY.NET.50.154	1	1	1	1

This alert indicates an internal host is attempting to connect to the Unix print spooler port (515) on an external host. However, all these alerts indicate that the source and destination hosts are all within the MY.NET.x.x network so these may be false positives.

Correlations: Detect from Chris Talianek at http://www.sans.org/y2k/012201.htm

Defensive Recommendation: Check on the hosts above to determine if printing from those destinations should be allowed. If not, ensure that LPR is configured correctly on the above 2 source hosts. Additionally, this would be a good time to double-check that you are blocking port 515 traffic from going outside your network.

Tiny Fragments – Possible Hostile Activity

Top 5 Source Hosts

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
62.32.160.39	13	13	10	10
62.32.204.42	8	8	7	7
194.65.57.182	5	5	4	4
194.65.28.26	5	5	4	4
212.45.227.212	4	4	3	3

Top 5 Destination Hosts

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.221.221	2	8	1	4
MY.NET.1.250	2	2	1	1
MY.NET.157.244	2	6	1	4
MY.NET.188.182	2	6	1	5
MY.NET.215.123	2	2	1	1

Network architectures limit the maximum size packet that they can process known as the Maximum Transmission Unit (MTU) or Maximum Segment Size (MSS). Ethernet limits the MTU to 1,500 bytes which means that if a host needs to send 2,400 bytes of data, TCP/IP will break it up (i.e. fragment it) into 2 packets. The IP header in the packet contains information (the ID and Fragment Offset) that is used by the receiving host to rebuild the original data.

Since fragmentation is used to break large chunks of data into MTU-sized chunks, we would expect to see large fragments except for possibly the last fragment in the chain. Attackers frequently use fragmentation to attempt to avoid detection by crafting packets so that data that would normally fit in one packet is broken into many small (i.e. Tiny) packets, sometimes only 1 byte per packet. Tiny fragments are likely due to some kind of malicious activity such as a denial of service and should be investigated further. Jason Anderson reviews fragmentation attacks at http://www.sans.org/infosecFAQ/threats/frag_attacks.htm

Correlation: David Hoelzer reported small fragmented packets at http://www.sans.org/y2k/121900.htm, which were further correlated by Laurie@edu at http://www.incidents.org/archives/y2k/122900.htm. Also see Detect #6 in E. A. Vazquez's practical at http://www.sans.org/y2k/practical/EAVazquezJr.html.

Defensive Recommendation: Some intrusion detection systems are not able to detect attack signatures broken up across multiple, fragmented packets so be sure your IDS will detect these types of attacks.

High port 65535 tcp – possible Red Worm – traffic

Top 5 Source Hosts

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.1.6	21	21	1	1
MY.NET.253.24	8	13	2	4
213.18.248.77	6	6	1	1
192.72.6.35	4	5	1	2
136.160.7.51	4	4	1	1

Top 5 Destination Hosts

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
62.208.91.225	21	21	1	1
MY.NET.253.23	8	51	2	12
213.18.248.77	6	6	1	1
194.90.1.1	4	4	1	1
136.160.7.51	4	4	1	1

The Adore Worm (aka Red Worm) attempts to gain access to Linux systems through vulnerabilities in LPRng, rpc.statd or BIND. If successful it installs a Trojan that listens on TCP port 65535. This alert triggers on TCP traffic to or from port 65535 which could indicate a potential Red Worm infection. However it should be noted that this could also be normal traffic since 65535 is also a valid ephemeral port.

Correlation: J. Anthony Dell describes the Adore Worm at http://www.sans.org/infosecFAQ/threats/mutation.htm. Also see Symantec's web page on the Adore worm at http://www.symantec.com/avcenter/venc/data/linux.adore.worm.html.

Defensive Recommendation: Ensure that all Unix and Linux systems have the latest patches installed. Run Adorefind (http://www.ists.dartmouth.edu/IRIA/knowledge_base/tools/adorefind.htm) on any of the servers listed above in MY.NET to see if they have been compromised. If so, they should be take offline until cleaned up and patched.

NMAP TCP ping!

Top 5 Source Hosts

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
64.152.70.68	21	21	2	2
63.211.17.228	4	4	1	1
146.145.28.226	2	2	2	2
208.35.152.13	2	2	1	1
205.244.118.245	1	1	1	1

Top 5 Destination Hosts

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.1.8	25	84	3	36
MY.NET.253.125	2	196	2	64
MY.NET.6.7	2	158	2	51
MY.NET.1.3	2	30	2	21
MY.NET.151.3	2	15	1	4

This alert is caused by a TCP packet that has the ACK flag set but the acknowledgement number is zero. It is characteristic of the reconnaissance tool Nmap that is discussed in a paper at http://www.sans.org/newlook/resources/IDFAQ/What_is_nmap.htm. Note that the bulk of the traffic was to MY.NET.1.8 on destination ports 80 and 53 which are typically open to outside access. In most of the scans the ports are reflexive (i.e. 80 -> 80 and 53->53).

Correlation: There are plenty of examples of Nmap reconnaissance scans at Incidents.org such as http://www.incidents.org/archives/y2k/032200-1700.htm.

Defensive Recommendation: Block access to hosts that should not be accessible from the outside via a firewall or packet-filtering device so that these scans are made ineffective. Also ensure all hosts have the latest security patches applied.

Port 55850 tcp – Possible myserver activity –

Top 5 Source Hosts

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
131.118.254.132	11	11	1	1
MY.NET.253.43	4	4	2	2
12.26.159.121	3	3	1	1
MY.NET.253.51	3	3	1	1
194.35.192.82	3	3	1	1

Top 5 Destination Hosts

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.5.74	11	15	1	4
MY.NET.5.29	3	16	1	8
MY.NET.253.51	3	14	1	9
12.26.159.121	3	3	1	1
193.63.84.10	3	3	1	1

This alert is triggered by activity on UDP port 55850, which is used by the MyServer DDOS Trojan. MyServer takes advantage of the rpc.statd vulnerabilities to replace the ls and ps commands on Linux systems then listens on port 55850.

Correlations: Mike Worman mentions MyServer in a posting at http://archives.neohapsis.com/archives/incidents/2000-10/0136.html. It is also mentioned at http://www.sans.org/082200.htm in postings by marchany and Scott Conti.

Defensive Recommendation: Ensure that all systems have the latest rpc.statd patches.

Null scan!

Top 5 Source Hosts

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
61.125.220.112	2	2	1	1
213.122.49.42	2	2	1	1
66.50.65.47	2	2	1	1
65.6.177.157	1	1	1	1

66.50.74.52	1	1	1	1

Top 5 Destination Hosts

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.150.133	2	77	2	31
MY.NET.70.195	2	10	1	6
MY.NET.53.54	2	22	2	10
MY.NET.217.62	2	61	2	17
MY.NET.21.228	2	3	1	2

TCP packets that have no flags set trigger this alert. Normally a TCP packet will have at least one flag set (e.g. ACK) so this traffic is highly unusual. Furthermore, some of the source and destination ports are 0 which is, again, highly unusual. This is typically indicative of reconnaissance, trying to evade intrusion detection systems or conduct OS fingerprinting and may be a prelude to a more focused attack.

Correlation: Joanne Treurniet demonstrates the use of hping reconnaissance tool, which generates these types of packets to port 0 (http://www.sans.org/y2k/practical/JoanneTreurniet.html - asst3).

Defensive Recommendation: Ensure that your perimeter firewall blocks denies all access by default except to specific hosts and ports that need to be available outside your network.

High port 65535 upd – possible Red Worm – traffic

Top 5 Source Hosts

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.157.244	5	5	2	2
MY.NET.218.14	4	4	3	3
MY.NET.101.89	3	7	1	2
195.42.215.219	2	2	1	1
217.4.142.109	2	2	1	1

Top 5 Destination Hosts

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
195.42.215.219	5	5	2	2
MY.NET.217.38	4	31	3	14
MY.NET.157.244	4	9	2	6
194.215.74.32	3	3	2	2
MY.NET.1.3	3	97	1	47

See explanation and defensive recommendations in "High port 65535 TCP – possible Red Worm – traffic" above.

SUNRPC highport access!

All Source Hosts

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
129.244.1.36	6	6	2	2
64.71.163.204	3	3	1	1
209.249.170.17	3	3	1	1

All Destination Hosts

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.218.66	5	8	1	3
MY.NET.98.128	3	7	1	4
MY.NET.139.40	3	6	1	3
MY.NET.218.14	1	384	1	116

These are similar to the "Attempted SUN RPC high port access" alerts in that they are triggered by attempts to port 32771 however the source ports were all different. The explanation and recommended defenses have already been outlined above.

ICMP SRC and DST outside network

All Source Hosts

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
172.156.129.239	1	1	1	1

All Destination Hosts

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
65.165.175.253	1	1	1	1

See UDP SRC and DST outside network.

Defensive Recommendation: In addition to the recommendation specified in "UDP SRC and DST outside network", consider blocking inbound ICMP requests.

Probably NMAP fingerprint attempt

All Source Hosts

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
195.132.112.174	1	1	1	1

All Destination Hosts

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.70.11	1	1296	1	49

As mentioned previously, Nmap (http://www.insecure.org) is a reconnaissance tool that can be used to do OS fingerprinting. This alert is triggered by TCP packets with the SYN, FIN, PUSH and URGENT flags set, which should not occur in normal circumstances. By analyzing the response to these crafted packets, Nmap can attempt to determine what operating system is running on a specific host that will point to specific exploits that can be tried to compromise the host. Note that there was only one packet with this signature and there were no other alerts from that source IP address so the attacker could have had some prior reconnaissance or, it could be due to an equipment malfunction.

Correlation: See http://www.incidents.org/archives/y2k/022800.htm for another example.

Defensive Recommendation: Since this technique depends on a response from the host, only listen on the specific ports required and only open specific ports in the firewall.

Samba client access

All Source Hosts

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
158.252.141.68	1	1	1	1

All Destination Hosts

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.138.33	1	1	1	1

Samba is software that allows Unix/Linux hosts to easily share files and printers with Windows hosts (see http://us1.samba.org/samba/samba.html for more information). This alert is triggered by an external host attempting to access the port (139) used for the file and printer sharing.

Correlation: The Samba Team has identified a vulnerability in SAMBA that could allow an attacker to gain root access on the Samba server. See http://us1.samba.org/samba/whatsnew/macroexploit.html (6/23/2001).

Defensive Recommendation: As mentioned in the defensive recommendations in "SMB Name Wildcard" alerts, file and print sharing outside your network should be blocked at your perimeter firewall. See SANS Top Ten #7 at http://www.sans.org/topten.htm.

SYN-FIN scan!

All Source Hosts

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
24.157.59.63	1	1	1	1

All Destination Hosts

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.97.228	1	7	1	4

A TCP packet with both the SYN and FIN flags set triggered this alarm. Since SYN and FIN do not occur in normal traffic, this packet is probably crafted to possibly circumvent an Intrusion Detection system while attempting reconnaissance. The following is the complete alert:

```
08/14-09:35:59.967077  [**] SYN-FIN scan! [**] 24.157.59.63:4 -> MY.NET.97.228:1214
```

Also note that there was an OOS packet just prior to this involving the same hosts:

```
08/14-09:34:18.096560 24.157.59.63:4 -> MY.NET.97.228:1214
TCP TTL:114 TOS:0x0 ID:62014  DF
**SF**** Seq: 0x4900584   Ack: 0x34FF0076   Win: 0x5010
TCP Options => EOL EOL
```

According to ARIN the source address belongs to a cable modem ISP in Canada:

```
Rogers@Home Bloor
1 Mount Pleasant Road
Toronto, ON M4Y 2Y5 CA
Netname: ON-ROG-7-3BLOOR-2
Netblock: 24.157.59.0 - 24.157.59.255
```

Destination port is 1214 is commonly used by KAZAA (www.kazaa.com) and Morpheus (www.musiccity.com) which are peer-to-peer file sharing products. SecurityTracker reports that KaZaA v1.3.1 and Morpheus v1.3 have a known vulnerability that could allow an attacker to have access to unauthorized files on the user's host (http://www.securitytracker.com/alerts/2001/Jul/1002113.html). In addition, Stan Sander outlines some strange behavior that Morpheus exhibited in his paper "Investigating One Incident of Anomalous Network Traffic" (http://www.sans.org/infosecFAQ/intrusion/net_traffic.htm). It is possible that an attacker is trying to find someone running one of these products and using crafted packets to circumvent detection.

Correlation: SANS contains plenty of examples of SYN-FIN scans. Including http://www.sans.org/y2k/032200-1700.htm and http://www.sans.org/y2k/051100.htm.

Defensive Recommendation: Check to see if the destination host is running a peer-to-peer product. If it is, you should consider finding a more secure alternative since there are known problems with many of the peer-to-peer products. You may want to consider blocking port 1214 as well as other known peer-to-peer ports at your perimeter firewall however this may be overkill unless you see significantly more traffic.

Scan Log Analysis

The following are the top 10 sources of scans from the portscan logs:

Count	Source IP	Registered to: (Source)	Destination Port(s)
1,070,130	MY.NET.70.69		111 (RPC)
73,028	MY.NET.134.14		Various from 0-32767
38,063	205.188.246.121	America Online, Inc USA (ARIN)	6970, 6972
29,674	MY.NET.160.114		Various 1024-65424
25,887	217.229.165.221	Deutsche Telekom AG Germany (RIPE)	1080 (WinGate), 110 (POP3), 111 (RPC), 119 (NNTP), 12345 (Netbus), 21 (FTP), 22 (SSH), 23 (Telnet), 25 (SMTP), 53 (DNS)
8,428	217.226.245.152	Deutsche Telekom AG Germany (RIPE)	21 (FTP)
7,728	4.33.128.228	BBN Planet USA, (ARIN)	111 (RPC)
7,671	212.187.67.33	Nijmegen Cablemodems 2 Netherlands (RIPE)	515 (LPR), 113(IDENT), 23(Telnet)
7,225	166.90.181.9	Level 3 Communications, Inc. USA (ARIN)	Mainly 111 (RPC), 3277x (also RPC)
6,774	MY.NET.217.38		Various from 21-65535

99.9% of the packets logged from MY.NET.70.69 were destined to various external hosts, all on port 111 (RPC). At an average rate of 25.3 packets per second (pps), this is obviously an automated scan indicating this host may have been compromised. Note however that this activity all occurred on 11 August 2001 and has since stopped, indicating someone may already be aware of the problem but this should be verified.

On 15 August 2001 at 16:08, MY.NET.134.14 sent 7,445 UDP packets (at 186 pps) from source port 1135 to 195.159.0.90 (PowerTech Information Systems, Oslo, Norway) on various ports. This was followed at 18:19 by 65,583 UDP packets (at 192.9 pps) from source ports 1190 and 1191 to various ports on 207.155.118.220 (PADS Development, California, USA) indicating that this host may also be compromised.

96.3% of the packets from America Online were UDP packets to port 6970 on sixteen MY.NET.x.x hosts. 6970 is a common port for the GateCrasher Trojan but is also used for streaming audio. Since there were no corresponding reports in the alerts log, this is probably streaming audio.

75.4% of the packets from MY.NET.160.114 were UDP packets to port 27005 on various destination networks including 24.x.x.x (@Home Network), 63.x.x.x (UUNET Technologies, Inc.), 64.x.x.x (Concentric Network Corporation), 65.x.x.x (@Home Network) and 66.x.x.x (ITC Deltacom). Port 27005 is used by the game Half-Life (see http://www.incidents.org/detect/gaming.php). However all the packets have a source port of 777, which is used by the AimSpy Trojan (see

http://www.sans.org/newlook/resources/IDFAQ/oddports.htm) so this host should be checked to see if it has been compromised. Note also that there were several SMB Name Wildcard alerts logged from some of those external network to MY.NET.160.114.

The largest grouping of traffic to and from MY.NET.217.38 appears to be on UDP ports in the 27000-29000 range, which is consistent with gaming. In addition there were several attempts on destination ports 6346 and 6347, which may indicate Gnutella traffic. The Snort alerts shows attempts from 212.179.26.6 (Bezeq International, Israel) to MY.NET.217.38 port 1214 (KaZaA) supporting the theory that this host may be or have participated in some peer-to-peer file sharing. Note also that the alerts show probes from 65.162.249.12 (RisingNet, Washington, USA) to this host for Trojans (port 27374) and open proxy service (port 1080) so this host is being actively targeted from external sources. It should be checked to see if it has been compromised.

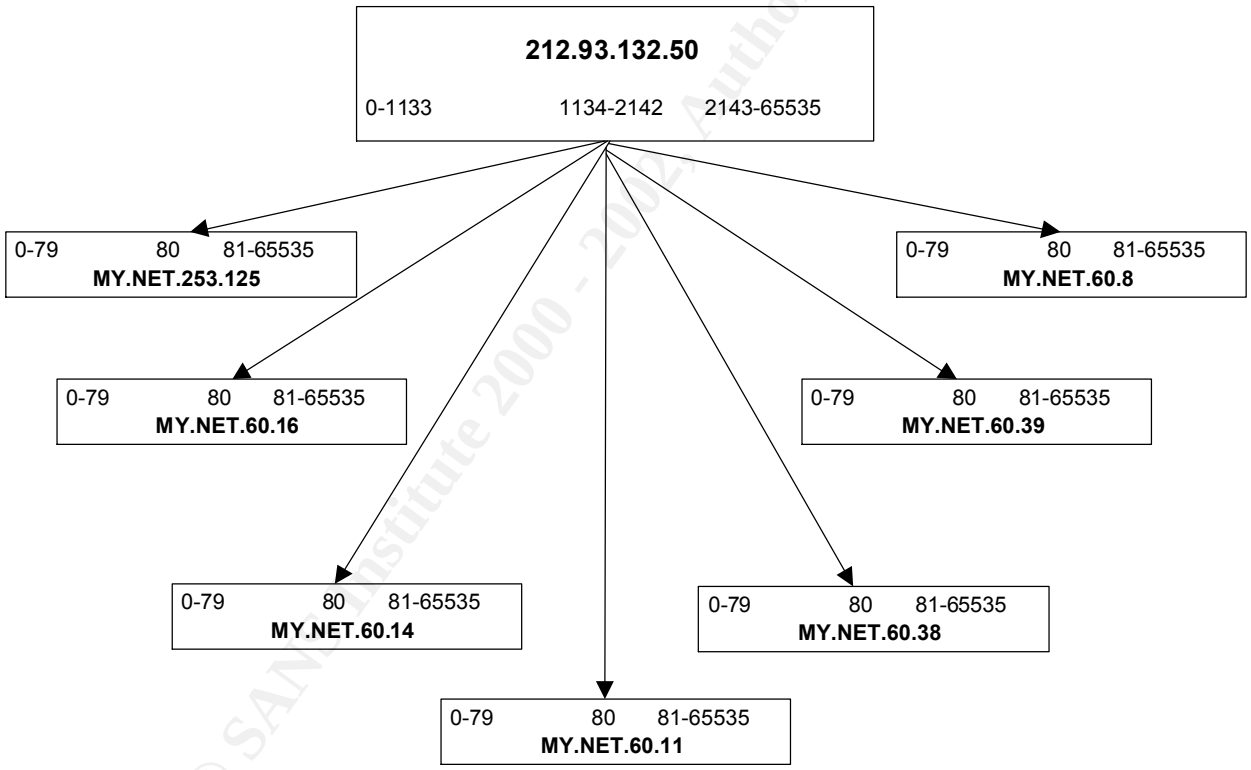
The remaining sources (217.229.165.221, 217.226.245.152, 4.33.128.228, 212.187.67.33 and 166.90.181.9) all appear to have been conducting network scans of MY.NET.x.x on ports with known vulnerabilities and Trojans. The targets of these scans should be checked to ensure they have the latest security patches installed.

Out Of Spec Analysis

The following were the top 5 sources of OOS packets:

Count	IP Address
147	212.93.132.50
132	199.183.24.194
52	128.46.156.155
29	139.117.1.8
18	198.110.76.242
17	130.207.193.70

From the link graph below, we can see that all the packets from the top source, 212.93.132.50, were destined to seven hosts in MY.NET.x.x. This source address is registered to Romania Data Systems, Bucharest, Romania.



Since all were to port 80, this appears to be a scan for web servers. All 147 packets had only the SYN flag and reserved bits set which could be normal if the source were attempting to negotiate Explicit Congestion Notification (ECN). However, in the following sequence of packets from this source notice how the IP identification jumps up and down within a couple of seconds.

08/12-**10:21:34.193679** 212.93.132.50:1145 -> MY.NET.60.39:80
TCP TTL:46 TOS:0x0 **ID:62672** DF
21S***** Seq: 0xD3EE28B9 Ack: 0x0 Win: 0x16D0

TCP Options => MSS: 1460 SackOK TS: 187497 0 EOL EOL EOL EOL

08/12-**10:21:34.348515** 212.93.132.50:1147 -> MY.NET.60.39:80
TCP TTL:46 TOS:0x0 **ID:3262** DF
21S***** Seq: 0xD438D0C7 Ack: 0x0 Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 187513 0 EOL EOL EOL EOL

08/12-**10:21:34.571059** 212.93.132.50:1149 -> MY.NET.60.16:80
TCP TTL:46 TOS:0x0 **ID:39694** DF
21S***** Seq: 0xD42F4781 Ack: 0x0 Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 187535 0 EOL EOL EOL EOL

08/12-**10:21:37.349752** 212.93.132.50:1147 -> MY.NET.60.39:80
TCP TTL:46 TOS:0x0 **ID:3263** DF
21S***** Seq: 0xD438D0C7 Ack: 0x0 Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 187813 0 EOL EOL EOL EOL

This does not seem normal since I would expect the IP ID to be fairly sequential in such a short timeframe. It is possible that this could be the result of a scan using Queso, which can be used to identify the host operating system, but it is interesting that this source did not show up in any of the Snort alerts for Queso fingerprinting. It is difficult to know for sure without the exact Snort rule that logged this packet and knowing if there was a response. Toby Miller's paper discusses the difficulties in differentiating Queso versus ECN at http://www.sans.org/y2k/ecn.htm.

In fact, of the 665 TCP packets in the OOS files, 659 had what appeared to be invalid TCP flags set and 580 of those were SYN packets with the reserved bits set on, possibly indicating that most were due to ECN. Five of the 580 packets were from 193.226.113.248 to MY.NET.150.133 (port 1214) and they all had the IP identification field set to zero, which indicates packet craft. This is correlated by the fact that this source also appeared as a scan and was flagged as Queso Fingerprinting in the alerts.

Additionally, one of the 580 also had an acknowledgement number set, again indicating packet craft. The remaining 79 of the 659 had tcp flags that were clearly crafted packets intended to circumvent intrusion detection systems and/or fingerprint operating systems.

The remaining 6 packets were flagged because they were malformed fragments. All had a fragment size of 34 bytes and offset of zero. Five of the packets were clearly crafted because they had both the Don't Fragment (DF) and More Fragments (MF) flags set. These were packets from the following sources:

63.207.128.121 -> MY.NET.150.133
64.160.13.55 -> MY.NET.70.11
63.205.8.9 -> MY.NET.217.62

Further information is needed on the last fragmented packet from 195.121.161.117 to MY.NET.227.215 since I could not tell why it was flagged.

In summary, it appears that most of the OOS packets may be related to ECN negotiation, which does not appear to be supported by the destination hosts in MY.NET.x.x. The remaining packets indicate reconnaissance and should be pursued further.

Summary

After completing analysis of the Snort data the following defensive steps are recommended:

- Only enable specific services on hosts that require them. All other services should be disabled.
- Ensure that all firewalls, routers, switches, host operating systems and applications (i.e. web servers, etc.,) have the latest security patches. Hosts that are accessible by external users should be addressed first however internal hosts including development servers, desktops and laptops should also be addressed.
- If your University's policy allows it, conduct periodic vulnerability scans against critical hosts.
- Set up egress filtering on your perimeter firewall and routers to prevent outbound traffic with source addresses that are not from your network space. This will help prevent an attacker from using your hosts to launch DDOS attacks.
- Monitor gaming, IRC and peer-to-peer file sharing to ensure it is consistent with the University's policy for students, faculty and employees.
- Ensure that all desktops, laptops and e-mail gateways have the latest anti-virus updates installed and activated.
- Since it is difficult sometimes to block traffic in University environments, consider using personal firewalls for desktops and laptops.

Finally the following hosts may have been compromised and should be checked and appropriate action taken:

MY.NET.70.69
MY.NET.134.14
MY.NET.160.114
MY.NET.217.38
MY.NET.98.164
MY.NET.97.214
MY.NET.97.207

Analysis Process

The analysis was done on a Windows NT machine with 256M RAM using SnortSnarf Version v010821.1 from Silicon Defense (http://www.silicondefense.com/software/snortsnarf/), Microsoft Access 2000, Microsoft Excel 2000 and Agrep for Win32 v3.37 from Tom Gries (http://www.tgries.de/agrep)

The process I followed is from Paul Asadoorian's practical (http://www.sans.org/y2k/practical/Paul_Asadoorian_GIAC.doc). I first combined all the alert files into one called "alert.total" using the NT copy command:

```
copy alert.010810 + alert.010811 + alert.010812 + alert.010813 + alert.010814 + alert.010815
alert.total
```

Since SnortSnarf requires numeric IP addresses, I wrote the a perl program called UniqueIP.pl (see Appendix B) to get a list of all the uniquely occurring IP addresses in that file so I could pick one that wasn't used to replace MY.NET with. This program outputs a sorted list with a count of each unique IP address that was imported into Access to generate the list of Top 10 Sources of Alerts". Note that I kept running out of memory running SnortSnarf so I ended up breaking it into two runs of three days each and manually combining the results.

For processing the Scan and OOS files I borrowed the snort_source.pl script from Mike Bell's practical (http://www.sans.org/y2k/practical/Mike_Bell_GCIA.doc). In addition to using it as-is, I also modified it to count destination IP and destination ports as needed.

Appendix A: References

American Registry for Internet Numbers. "WHOIS Database" URL: http://www.arin.net/whois/index.html (22 August 2001).

Anderson, Jason. "An Analysis of Fragmentation Attacks" 15 March 2001. URL: http://www.sans.org/infosecFAQ/threats/frag_attacks.htm (17 September 2001).

Bejtlich, Richard. "Network Intrusion Detection of Third Party Effects" 5 September 2000. URL: http://packetstormsecurity.org/papers/evaluation/nid_3pe_v101.pdf (26 August 2001).

Chmielarski, Tom. "Reconnaissance Techniques using Spoofed IP Addresses" 4 April 2001. URL: http://www.sans.org/newlook/resources/IDFAQ/spoofed_IP.htm (9 September 2001).

Cisco. "PIX Firewall Version 5.3" 9 August 2001. URL: http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_v53/syslog/index.htm (22 August 2001).

Cisco. "Configuring Access Control Lists" 15 August 2000. URL: http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_5_4/msfc/acc_list.htm (3 September 2001).

Dshield.org. "Search the DShield Database" 22 August 2001. URL: http://www.dshield.org/search.html (22 August 2001).

Incidents.org. "Top 10 Attackers" URL: http://www.incidents.org/cid/query/top_10ip_7.php (22 August 2001).

Internet Assigned Numbers Authority. "Well Known Port Numbers." 21 August 2001. URL: http://www.iana.org/assignments/port-numbers (22 August 2001).

Internet Assigned Numbers Authority. "Internet Protocol V4 Address Space" 19 June 2001. URL: http://www.iana.org/assignments/ipv4-address-space (28 August 2001).

Internet Security Systems. "X-Force" URL: http://xforce.iss.net/ (22 August 2001).

Miller, Toby. "ECN and it's impact on Intrusion Detection" 1999 URL: http://www.sans.org/y2k/ecn.htm (16 September 2001).

Mitre. "Common Vulnerabilities and Exposures" Version 20010507. URL: http://cve.mitre.org/cve/ (22 August 2001).

Northcutt, Stephen and Novak, Judy. Network Intrusion Detection: An Analyst's Handbook 2nd Edition. Indianapolis: New Riders Publishing. 2001.

Postel, J. "RFC792" September 1981. URL: http://www.rfc-editor.org/rfc/rfc792.txt (26 August 2001).

Sander, Stan. "Investigating One Incident of Anomalous Network Traffic" 15 June 2001. URL: http://www.sans.org/infosecFAQ/intrusion/net_traffic.htm. (12 September 2001).

SANS. "How to Eliminate the Ten Most Critical Internet Security Threats Version 1.33" 25 June 2001. URL: http://www.sans.org/topten.htm (5 September 2001)

SANS. "Windows Security Digest" Vol.4 No.3. 31 March 2001. URL: http://www.sans.org/newlook/digests/ntarchives/033101.htm (12 September 2001).

Shadow. URL: http://www.nswc.navy.mil/ISSEC/CID/index.html (18 September 2001).

Simovits. "Trojan List sorted on Trojan port" URL: http://www.simovits.com/trojans/trojans.html (24 August 2001).

Stevens, W. Richard. TCP/IP Illustrated, Volume 1. Reading: Addison-Wesley Longman, Inc. 1994.

University of Michigan's Computer Aided Engineering Network. "Welcome to the Internet Distance Map Service" URL: http://idmaps.eecs.umich.edu/ (22 August 2001).

Van Dixon, Kevin. "Spoof Bounce" 19 February 2001. URL: http://www.sans.org/infosecFAQ/intrusion/spoof.htm (10 September 2001)

Appendix B: UniqueIP.pl

```
#!/usr/bin/perl

# Finds all unique IP addresses in a file (8/13/01)

# Process parameters
#------------------
use Getopt::Std;
getopts("i:o:"); #valid command line switches

if (!defined($opt_i) or !defined($opt_o)) {
  print "\nSyntax: $0 -i inputfile -o outputfile\n\n";
  print "where   inputfile is the file to be searched\n";
  die   "        outputfile is where to store the results\n\n";
}

# First find all IP addresses and save in temp file
#--------------------------------------------------
open IN, $opt_i or die "Cannot open $opt_i for read :$!";
$out = "temp\$1.txt";
open OUT, ">$out" or die "Cannot open $out for write :$!";

print "\n\nSearching $in for IP addresses....\n\n";
$cnt = 0;
while (<IN>) {
   # The following looks for IP addresses like x.x.x.x
   # and the /g option looks for all occurrences.
   # The results are put in the array called "matches"
   @matches = /([0-9]+\.[0-9]+\.[0-9]+\.[0-9]+)/g;
   # For each one we found, write them as separate records
   foreach (@matches) {
      $cnt++;
      print OUT "$_\n";
   }
}
print "Total IP addresses found: $cnt";

# Sort the temp file
#-------------------
print "\n\nSorting......\n";
system("sort <$out >temp\$2.txt");

# Check the result of the system call, which is returned in $?
if (!$?) {
   print "Sorting complete.";
   # Now eliminate the duplicate entries
   $in="temp\$2.txt";
   open IN, $in or die "Cannot open $in for read :$!";
   open OUT, ">$opt_o" or die "Cannot open $opt_o for write :$!";
   print "\n\nEliminating duplicate IP addresses....\n\n";
   $cnt = 0;
   $dupcnt = 1;
   $oldip = <IN>;
```

```
        while (<IN>) {
          if ($oldip ne $_) {
            print OUT "$dupcnt,$oldip";  #Note that crlf is already part of record
            $oldip = $_;
            $cnt++;
            $dupcnt = 1;
          }
          else
          {
            $dupcnt++;
          }
        }
        print "Completed - Total unique IP addresses: $cnt\n\n";
      }
      else {
        print "Problem with sort - return code: $?\n\n";
}
```