Global Information Assurance Certification Paper

Dan Wangler

GIAC Intrusion Detection Practical
SANS Security New Orleans
January, 2001

Assignment 1- Network Detects

Defect #1

1438 20:24:28.25478 my.other.net.07 -> my.net.9.56 UDP D=45000 S=45000 LEN=52

```
 0: 4500 0048 55f6 4000 ff11 bb0a ac19 0939    E..HU.@........9
16: ac19 0938 afc8 afc8 0034 3551 0100 2710    ...8.....45Q..'.
32: 0000 0039 0000 0064 0100 2710 0000 0038    ...9...d..'....8
48: 0000 0064 0000 1178 002c 0101 9b00 0100    ...d...x.,......
64: 0000 0000 0000 0001                        ........
```

1439 20:24:28.25484 my.net.9.56 -> my.other.net.07 ICMP Destination unreachable (UDP port 45000 unreachable)

```
 0: 4500 0064 d00c 4000 ff01 40e8 ac19 0938    E..d..@...@....8
16: ac19 0939 0303 67e6 0000 0000 4500 0048    ...9..g.....E..H
32: 55f6 4000 ff11 bb0a ac19 0939 ac19 0938    U.@........9...8
48: afc8 afc8 0034 3551 0100 2710 0000 0039    .....45Q..'....9
64: 0000 0064 0100 2710 0000 0038 0000 0064    ...d..'....8...d
80: 0000 1178 002c 0101 9b00 0100 0000 0000    ...x.,..........
96: 0000 0001                                  ....
```

1443 20:24:33.25430 my.other.net.07 -> my.net.9.56 UDP D=45000 S=45000 LEN=52

```
 0: 4500 0048 55f7 4000 ff11 bb09 ac19 0939    E..HU.@........9
16: ac19 0938 afc8 afc8 0034 3551 0100 2710    ...8.....45Q..'.
32: 0000 0039 0000 0064 0100 2710 0000 0038    ...9...d..'....8
48: 0000 0064 0000 1178 002c 0101 9b00 0100    ...d...x.,......
64: 0000 0000 0000 0001                        ........
```

1444 20:24:33.25437 my.net.9.56 -> my.other.net.07 ICMP Destination unreachable (UDP port 45000 unreachable)

```
 0: 4500 0064 d00d 4000 ff01 40e7 ac19 0938    E..d..@...@....8
16: ac19 0939 0303 67e6 0000 0000 4500 0048    ...9..g.....E..H
32: 55f7 4000 ff11 bb09 ac19 0939 ac19 0938    U.@........9...8
48: afc8 afc8 0034 3551 0100 2710 0000 0039    .....45Q..'....9
64: 0000 0064 0100 2710 0000 0038 0000 0064    ...d..'....8...d
80: 0000 1178 002c 0101 9b00 0100 0000 0000    ...x.,..........
96: 0000 0001                                  ....
```

1448 20:24:38.25399 my.other.net.07 -> my.net.9.56 UDP D=45000 S=45000 LEN=52

```
 0: 4500 0048 55f8 4000 ff11 bb08 ac19 0939    E..HU.@........9
16: ac19 0938 afc8 afc8 0034 3551 0100 2710    ...8.....45Q..'.
32: 0000 0039 0000 0064 0100 2710 0000 0038    ...9...d..'....8
48: 0000 0064 0000 1178 002c 0101 9b00 0100    ...d...x.,......
64: 0000 0000 0000 0001                        ........
```

1449 20:24:38.25405 my.net.9.56 -> my.other.net.07 ICMP Destination unreachable (UDP port 45000 unreachable)

```
 0: 4500 0064 d00e 4000 ff01 40e6 ac19 0938    E..d..@...@....8
16: ac19 0939 0303 67e6 0000 0000 4500 0048    ...9..g.....E..H
32: 55f8 4000 ff11 bb08 ac19 0939 ac19 0938    U.@........9...8
48: afc8 afc8 0034 3551 0100 2710 0000 0039    .....45Q..'....9
64: 0000 0064 0100 2710 0000 0038 0000 0064    ...d..'....8...d
80: 0000 1178 002c 0101 9b00 0100 0000 0000    ...x.,..........
96: 0000 0001                                  ....
```

1. Source of Trace.

Company Internal network

2. Detect was generated by:

snoop running on a Sun Sparc 10 with Solaris 2.6

snoop –ta –x14 –s1500 -I infile host my.net.9.56

3. Probability the source address was spoofed:

It is unlikely that the source address was spoofed. The probes, although frequent, are not frequent to cause a Denial of Service. Therefore, the probe was probably

looking for something specific.

4. Description of attack:

This appears to be some kind of a probe on port 45000 on my.net.9.56. The probes were frequent, 5 seconds apart, and continued overnight.

5. Attack mechanism:

Constant probing for a listener on port 45000. The source address apparently is expecting something listening at this port, possible a Trojan.

6. Correlations:

This particular detect has never been reported before. A check on SANS, CVE, arachnid showed nothing.

7. Evidence of active targeting:

This was definitely targeting my.net.9.56. These were the only records during the period of this scan to involve my.other.07. my.net.9.56 was seen involved in communication between other addresses.

8. Severity:

Target Criticality – 3 (Machine is a group Unix Server.)

Lethality – 1 (The fact this probe ran over night at 5 second intervals indicated this was not a denial of service.)

System Countermeasures – 4(All recommended patches for this OS release are installed.)

Network Countermeasures – 1 (This probe originated internally. If this is an attack it is already inside our firewall

Severity - -1 (3+1) – (4+ 1) = -1

.

9. Defensive recommendation:

Owner of my.other.net.07 needs to be contacted to see what that machine is aiming at my.net.9.56. It may be an application that is trying to access resources on my.net.9.56. However, it may be that my.other.net.07 may be compromised and in some other trouble. For right now, ny.net.9.56 is surviving.

10. Multiple choice test question:

In the above trace, which could have signaled a spoofed source address?

a) Noisy probes from the same address and port
b) Noisy probes from multiple addresses and ports
c) Quiet probes from same address and port
d) Quiet probes from multiple addresses and ports
e) None of the above
f) All of the above

Answer: e all of the above. If this were a Denial of Service, any of the above four methods could have been used, all of which could have had spoofed source addresses.

Defect #2

494 20:09:25.11515 attack.net.223.103 -> my.net.19.155 TCP D=5858 S=3167 Syn Seq=3839253760 Len=0 Win=16384 Options=<mss 1460,nop,nop,sackOK>

```
 0: 4500 0030 9d7b 4000 8006 f2ac ac19 0935    E..0.{@.........5
16: ac19 0938 0c5f 16e2 e4d6 5d00 0000 0000    ...8._....].....
32: 7002 4000 7367 0000 0204 05b4 0101 0402    p.@.sg..........
```

495 20:09:25.11535 attack.net.223.103 -> my.net.19.155 TCP D=5000 S=3166 Syn Seq=3839202028 Len=0 Win=16384 Options=<mss 1460,nop,nop,sackOK>

```
 0: 4500 0030 9d7c 4000 8006 f2ab ac19 0935    E..0.|@.........5
16: ac19 0938 0c5e 1388 e4d5 92ec 0000 0000    ...8.^..........
32: 7002 4000 40d7 0000 0204 05b4 0101 0402    p.@.@...........
```

496 20:09:25.11555 attack.net.223.103 -> my.net.19.155 TCP D=8000 S=3175 Syn Seq=3839646883 Len=0 Win=16384 Options=<mss 1460,nop,nop,sackOK>

```
 0: 4500 0030 9d7d 4000 8006 f2aa ac19 0935    E..0.}@.........5
16: ac19 0938 0c67 1f40 e4dc 5ca3 0000 0000    ...8.g.@..\.....
32: 7002 4000 6b58 0000 0204 05b4 0101 0402    p.@.kX..........
```

497 20:09:25.11574 attack.net.223.103 -> my.net.19.155 TCP D=13326 S=3177 Syn Seq=3839743022 Len=0 Win=16384 Options=<mss 1460,nop,nop,sackOK>

```
 0: 4500 0030 9d7e 4000 8006 f2a9 ac19 0935    E..0.~@.....5
16: ac19 0938 0c69 340e e4dd d42e 0000 0000     ...8.i4.....
32: 7002 4000 defb 0000 0204 05b4 0101 0402    p.@..û.....
```

498 20:09:25.11594 attack.net.223.103 -> my.net.19.125 TCP D=4557 S=3165 Syn Seq=3839145853 Len=0 Win=16384 Options=<mss 1460,nop,nop,sackOK>

```
 0: 4500 0030 9d7f 4000 8006 f2a8 ac19 0935    E..0..@.....5
16: ac19 0938 0c5d 11cd e4d4 b77d 0000 0000     ...8.].....}....
32: 7002 4000 1e03 0000 0204 05b4 0101 0402    p.@.....
```

499 20:09:25.11614 attack.net.223.103 -> my.net.19.125 TCP D=6666 S=3169 Syn Seq=3839360776 Len=0 Win=16384 Options=<mss 1460,nop,nop,sackOK>

```
 0: 4500 0030 9d80 4000 8006 f2a7 ac19 0935    E..0..@.....5
16: ac19 0938 0c61 1a0a e4d7 ff08 0000 0000     ...8.a.....
32: 7002 4000 ce33 0000 0204 05b4 0101 0402    p.@..3.....
```

500 20:09:25.11639 attack.net.223.103 -> my.net.19.125 TCP D=7000 S=3171 Syn Seq=3839434661 Len=0 Win=16384 Options=<mss 1460,nop,nop,sackOK>

```
 0: 4500 0030 9d81 4000 8006 f2a6 ac19 0935    E..0..@.....5
16: ac19 0938 0c63 1b58 e4d9 1fa5 0000 0000     ...8.c.X.....
32: 7002 4000 ac45 0000 0204 05b4 0101 0402    p.@..E.....
```

501 20:09:25.11652 my.net.19.125 -> attack.net.223.103 TCP D=3172 S=7001 Rst Ack=3839492392 Win=0

```
 0: 4500 0028 54f1 4000 8006 3b3f ac19 0938    E..(T.@...;?...8
16: ac19 0935 1b59 0c64 0000 0000 e4da 0128     ...5.Y.d.......(
32: 5014 0000 3771 0000                        P...7q..
```

502 20:09:25.11661 my.net.19.125 -> attack.net.223.103 TCP D=3170 S=6667 Rst Ack=3839397810 Win=0

```
 0: 4500 0028 54f2 4000 8006 3b3e ac19 0938    E..(T.@...;>...8
16: ac19 0935 1a0b 0c62 0000 0000 e4d8 8fb2     ...5...b.....
32: 5014 0000 aa38 0000                        P....8..
```

503 20:09:25.11670 my.net.19.125 -> attack.net.223.103 TCP D=3173 S=7002 Rst Ack=3839549463 Win=0

```
 0: 4500 0028 54f3 4000 8006 3b3d ac19 0938    E..(T.@...;=...8
16: ac19 0935 1b5a 0c65 0000 0000 e4da e017     ...5.Z.e.....
32: 5014 0000 587f 0000                        P...X...
```

504 20:09:25.11678 my.net.19.155 -> attack.net.223.103 TCP D=3176 S=9000 Rst Ack=3839704023 Win=0

```
 0: 4500 0028 54f4 4000 8006 3b3c ac19 0938    E..(T.@...;<...8
16: ac19 0935 2328 0c68 0000 0000 e4dd 3bd7    ...5#(.h......;.
32: 5014 0000 f4eb 0000                        P.......
```

1. Source of Trace.

Company Network

2. Detect was generated by:

snoop running on a Sun Sparc 10 with Solaris 2.6

snoop –ta –x14 –s1500 -I infile host my.net.19.155

3. Probability the source address was spoofed:

Good probability. In the segment of the scan above, and in the complete scan, source address was probing my.net.19.155 and not waiting for a reply. The replies come in the form of a Ack-Rst. If three way handshaking would have been evident, then the probability would have been null.

4. Description of attack:

The scan shows a port probe launched against my.net.19.155. The could also have been a port scan where the source device was listening to see if it would receive a Syn-Ack.

5. Attack mechanism:

The attack mechanism is a Syn Flood of packets aimed at each port on my.net.19.155. Since this was extremely noisy, every .0001 seconds, this is probably a Denial of Service. This attack, though launched against multiple ports on the same machine, was launched with such intensity as it trying to tie up the machine and prevent its use. Also, with the number of Syn's directed at the machine, it could have been trying to fill the connection queue on the server with "half-open" connections so it would not respond to addition connection attempts.

6. Correlations:

The following describes the impact of Syn Floods and what action can be taken:
CERT® Advisory CA-1996-21 TCP SYN Flooding and IP Spoofing Attacks
Original issue date: September 19, 1996
Last revised: November 29, 2000**7. Evidence of active targeting:**

7. Evidence of active targeting:

This appears to be evidence of active targeting since the attacks were directed towards my.net.19.155. .

8. Severity:

Target Criticality – 3 (Machine is a group Unix Server.)
Lethality – 5 (Systems providing TCP-based services to the Internet community may be unable to provide those services while under attack and for some time after the attack ceases. The service itself is not harmed by the attack; usually only the ability to provide the service is impaired. In some cases, the system may exhaust memory, crash, or be rendered otherwise inoperative – Cert Advisory).
Network Countermeasures – 1 (This probe originated internally. If this is an attack it is already inside our firewall
Severity - 3(3+5) – (4+ 1) = 3 = Severe

9. Defensive recommendation:

There is, as yet, no generally accepted solution to this problem with the current IP protocol technology. However, proper router configuration and the addition of recommended patches can reduce the likelihood of a prolonged outage.

10. Multiple choice test question:

What response can be expected from a reply to a spoofed address?
A) No Response.
B) Reset
C) Ack Fin
D) HELO

Either/both A and B would be correct. If the spoofed address is that of a working machine, the proper response to the reply (RST ACK) would be a RST. If the machine were unknown or down, then no reply would be sent back.

Detect #3

```
1716 20:09:55.33174  scanner.net.18.106 -> my.net.25.68  ETHER Type=0800 (IP), size =
82 bytes
1716 20:09:55.33174  scanner.net.18.106 -> my.net.25.68  IP  D=my.net.25.68
S=scanner.net.18.106 LEN=68, ID=41110
1716 20:09:55.33174  scanner.net.18.106 -> my.net.25.68  UDP D=603 S=3225 LEN=48
1716 20:09:55.33174  scanner.net.18.106 -> my.net.25.68  RPC C XID=3735938274
PROG=0 (?) VERS=0 PROC=0

         0: 4500 0044 a096 0000 8011 2f73 ac19 0935    E..D....../s...5
        16: ac19 0938 0c99 025b 0030 c267 dead e4e2     ...8...[.0.g....
        32: 0000 0000 0000 0002 0000 0000 0000 0000     ................
        48: 0000 0000 0000 0000 0000 0000 0000 0000     ................
        64: 0000 0000                                   ....

_____
1717 20:09:55.33180  scanner.net.18.106 -> my.net.25.68  ETHER Type=0800 (IP), size =
82 bytes
1717 20:09:55.33180  scanner.net.18.106 -> my.net.25.68  IP  D=my.net.25.68
S=scanner.net.18.106 LEN=68, ID=41111
1717 20:09:55.33180  scanner.net.18.106 -> my.net.25.68  UDP D=604 S=3225 LEN=48
1717 20:09:55.33180  scanner.net.18.106 -> my.net.25.68  RPC C XID=3735938275
PROG=0 (?) VERS=0 PROC=0

         0: 4500 0044 a097 0000 8011 2f72 ac19 0935    E..D....../r...5
        16: ac19 0938 0c99 025c 0030 c265 dead e4e3     ...8...\.0.e....
        32: 0000 0000 0000 0002 0000 0000 0000 0000     ................
        48: 0000 0000 0000 0000 0000 0000 0000 0000     ................
        64: 0000 0000                                   ....

_____
1718 20:09:55.33190  scanner.net.18.106 -> my.net.25.68  ETHER Type=0800 (IP), size =
82 bytes
1718 20:09:55.33190  scanner.net.18.106 -> my.net.25.68  IP  D=my.net.25.68
S=scanner.net.18.106 LEN=68, ID=41112
1718 20:09:55.33190  scanner.net.18.106 -> my.net.25.68  UDP D=605 S=3225 LEN=48
1718 20:09:55.33190  scanner.net.18.106 -> my.net.25.68  RPC C XID=3735938276
PROG=0 (?) VERS=0 PROC=0

         0: 4500 0044 a098 0000 8011 2f71 ac19 0935    E..D....../q...5
        16: ac19 0938 0c99 025d 0030 c263 dead e4e4     ...8...].0.c....
        32: 0000 0000 0000 0002 0000 0000 0000 0000     ................
        48: 0000 0000 0000 0000 0000 0000 0000 0000     ................
        64: 0000 0000                                   ....

_____
1719 20:09:55.33197  scanner.net.18.106 -> my.net.25.68  ETHER Type=0800 (IP), size =
```

82 bytes
1719 20:09:55.33197 scanner.net.18.106 -> my.net.25.68 IP D=my.net.25.68 S=scanner.net.18.106 LEN=68, ID=41113
1719 20:09:55.33197 scanner.net.18.106 -> my.net.25.68 UDP D=606 S=3225 LEN=48
1719 20:09:55.33197 scanner.net.18.106 -> my.net.25.68 RPC C XID=3735938277 PROG=0 (?) VERS=0 PROC=0

```
 0: 4500 0044 a099 0000 8011 2f70 ac19 0935    E..D....../p...5
16: ac19 0938 0c99 025e 0030 c261 dead e4e5    ...8...^.0.a....
32: 0000 0000 0000 0002 0000 0000 0000 0000    ................
48: 0000 0000 0000 0000 0000 0000 0000 0000    ................
64: 0000 0000                                  ....
```

1720 20:09:55.33206 scanner.net.18.106 -> my.net.25.68 ETHER Type=0800 (IP), size = 82 bytes
1720 20:09:55.33206 scanner.net.18.106 -> my.net.25.68 IP D=my.net.25.68 S=scanner.net.18.106 LEN=68, ID=41114
1720 20:09:55.33206 scanner.net.18.106 -> my.net.25.68 UDP D=607 S=3225 LEN=48
1720 20:09:55.33206 scanner.net.18.106 -> my.net.25.68 RPC C XID=3735938278 PROG=0 (?) VERS=0 PROC=0

```
 0: 4500 0044 a09a 0000 8011 2f6f ac19 0935    E..D....../o...5
16: ac19 0938 0c99 025f 0030 c25f dead e4e6    ...8..._.0._....
32: 0000 0000 0000 0002 0000 0000 0000 0000    ................
48: 0000 0000 0000 0000 0000 0000 0000 0000    ................
64: 0000 0000                                  ....
```

1721 20:09:55.33209 my.net.25.68 -> scanner.net.18.106 ETHER Type=0800 (IP), size = 110 bytes
1721 20:09:55.33209 my.net.25.68 -> scanner.net.18.106 IP D=scanner.net.18.106 S=my.net.25.68 LEN=96, ID=22298
1721 20:09:55.33209 my.net.25.68 -> scanner.net.18.106 ICMP Destination unreachable (UDP port 600 unreachable)

```
 0: 4500 0060 571a 4000 ff01 b9e2 ac19 0938    E..`W.@........8
16: ac19 0935 0303 67de 0000 0000 4500 0044    ...5..g.....E..D
32: a093 0000 8011 2f76 ac19 0935 ac19 0938    ....../v...5...8
48: 0c99 0258 0030 c26d dead e4df 0000 0000    ...X.0.m........
64: 0000 0002 0000 0000 0000 0000 0000 0000    ................
80: 0000 0000 0000 0000 0000 0000 0000 0000    ................
```

1722 20:09:55.33215 scanner.net.18.106 -> my.net.25.68 ETHER Type=0800 (IP), size = 82 bytes

```
1722 20:09:55.33215  scanner.net.18.106 -> my.net.25.68  IP  D=my.net.25.68
S=scanner.net.18.106 LEN=68, ID=41115
1722 20:09:55.33215  scanner.net.18.106 -> my.net.25.68  UDP D=608 S=3225 LEN=48
1722 20:09:55.33215  scanner.net.18.106 -> my.net.25.68  RPC C XID=3735938279
PROG=0 (?) VERS=0 PROC=0

         0: 4500 0044 a09b 0000 8011 2f6e ac19 0935    E..D....../n...5
        16: ac19 0938 0c99 0260 0030 c25d dead e4e7    ...8...`.0.]....
        32: 0000 0000 0000 0002 0000 0000 0000 0000    ................
        48: 0000 0000 0000 0000 0000 0000 0000 0000    ................
        64: 0000 0000                                  ....

_____
1723 20:09:55.33224  scanner.net.18.106 -> my.net.25.68  ETHER Type=0800 (IP), size =
82 bytes
1723 20:09:55.33224  scanner.net.18.106 -> my.net.25.68  IP  D=my.net.25.68
S=scanner.net.18.106 LEN=68, ID=41116
1723 20:09:55.33224  scanner.net.18.106 -> my.net.25.68  UDP D=609 S=3225 LEN=48
1723 20:09:55.33224  scanner.net.18.106 -> my.net.25.68  RPC C XID=3735938280
PROG=0 (?) VERS=0 PROC=0

         0: 4500 0044 a09c 0000 8011 2f6d ac19 0935    E..D....../m...5
        16: ac19 0938 0c99 0261 0030 c25b dead e4e8    ...8...a.0.[....
        32: 0000 0000 0000 0002 0000 0000 0000 0000    ................
        48: 0000 0000 0000 0000 0000 0000 0000 0000    ................
        64: 0000 0000                                  ....

_____
1724 20:09:55.33233  scanner.net.18.106 -> my.net.25.68  ETHER Type=0800 (IP), size =
82 bytes
1724 20:09:55.33233  scanner.net.18.106 -> my.net.25.68  IP  D=my.net.25.68
S=scanner.net.18.106 LEN=68, ID=41117
1724 20:09:55.33233  scanner.net.18.106 -> my.net.25.68  UDP D=610 S=3225 LEN=48
1724 20:09:55.33233  scanner.net.18.106 -> my.net.25.68  RPC C XID=3735938281
PROG=0 (?) VERS=0 PROC=0

         0: 4500 0044 a09d 0000 8011 2f6c ac19 0935    E..D....../l...5
        16: ac19 0938 0c99 0262 0030 c259 dead e4e9    ...8...b.0.Y....
        32: 0000 0000 0000 0002 0000 0000 0000 0000    ................
        48: 0000 0000 0000 0000 0000 0000 0000 0000    ................
        64: 0000 0000                                  ....

_____
1725 20:09:55.33242  scanner.net.18.106 -> my.net.25.68  ETHER Type=0800 (IP), size =
82 bytes
1725 20:09:55.33242  scanner.net.18.106 -> my.net.25.68  IP  D=my.net.25.68
```

S=scanner.net.18.106 LEN=68, ID=41118
1725 20:09:55.33242 scanner.net.18.106 -> my.net.25.68 UDP D=611 S=3225 LEN=48
1725 20:09:55.33242 scanner.net.18.106 -> my.net.25.68 RPC C XID=3735938282 PROG=0 (?) VERS=0 PROC=0

```
 0: 4500 0044 a09e 0000 8011 2f6b ac19 0935    E..D....../k...5
16: ac19 0938 0c99 0263 0030 c257 dead e4ea    ...8...c.0.W....
32: 0000 0000 0000 0002 0000 0000 0000 0000    ................
48: 0000 0000 0000 0000 0000 0000 0000 0000    ................
64: 0000 0000                                  ....
```

1726 20:09:55.33250 scanner.net.18.106 -> my.net.25.68 ETHER Type=0800 (IP), size = 82 bytes
1726 20:09:55.33250 scanner.net.18.106 -> my.net.25.68 IP D=my.net.25.68 S=scanner.net.18.106 LEN=68, ID=41119
1726 20:09:55.33250 scanner.net.18.106 -> my.net.25.68 UDP D=612 S=3225 LEN=48
1726 20:09:55.33250 scanner.net.18.106 -> my.net.25.68 RPC C XID=3735938283 PROG=0 (?) VERS=0 PROC=0

```
 0: 4500 0044 a09f 0000 8011 2f6a ac19 0935    E..D....../j...5
16: ac19 0938 0c99 0264 0030 c255 dead e4eb    ...8...d.0.U....
32: 0000 0000 0000 0002 0000 0000 0000 0000    ................
48: 0000 0000 0000 0000 0000 0000 0000 0000    ................
```

1727 20:09:55.33259 scanner.net.18.106 -> my.net.25.68 ETHER Type=0800 (IP), size = 82 bytes
1727 20:09:55.33259 scanner.net.18.106 -> my.net.25.68 IP D=my.net.25.68 S=scanner.net.18.106 LEN=68, ID=41120
1727 20:09:55.33259 scanner.net.18.106 -> my.net.25.68 UDP D=613 S=3225 LEN=48
1727 20:09:55.33259 scanner.net.18.106 -> my.net.25.68 RPC C XID=3735938284 PROG=0 (?) VERS=0 PROC=0

```
 0: 4500 0044 a0a0 0000 8011 2f69 ac19 0935    E..D....../i...5
16: ac19 0938 0c99 0265 0030 c253 dead e4ec    ...8...e.0.S....
32: 0000 0000 0000 0002 0000 0000 0000 0000    ................
48: 0000 0000 0000 0000 0000 0000 0000 0000    ................
64: 0000 0000                                  ....
```

1728 20:09:55.33267 scanner.net.18.106 -> my.net.25.68 ETHER Type=0800 (IP), size = 82 bytes
1728 20:09:55.33267 scanner.net.18.106 -> my.net.25.68 IP D=my.net.25.68 S=scanner.net.18.106 LEN=68, ID=41121
1728 20:09:55.33267 scanner.net.18.106 -> my.net.25.68 UDP D=614 S=3225 LEN=48
1728 20:09:55.33267 scanner.net.18.106 -> my.net.25.68 RPC C XID=3735938285 PROG=0 (?) VERS=0 PROC=0

```
 0: 4500 0044 a0a1 0000 8011 2f68 ac19 0935    E..D....../h...5
16: ac19 0938 0c99 0266 0030 c251 dead e4ed    ...8...f.0.Q....
32: 0000 0000 0000 0002 0000 0000 0000 0000    ................
48: 0000 0000 0000 0000 0000 0000 0000 0000    ................
64: 0000 0000                                  ....
```

1729 20:09:55.33276 scanner.net.18.106 -> my.net.25.68 ETHER Type=0800 (IP), size = 82 bytes
1729 20:09:55.33276 scanner.net.18.106 -> my.net.25.68 IP D=my.net.25.68 S=scanner.net.18.106 LEN=68, ID=41122
1729 20:09:55.33276 scanner.net.18.106 -> my.net.25.68 UDP D=615 S=3225 LEN=48
1729 20:09:55.33276 scanner.net.18.106 -> my.net.25.68 RPC C XID=3735938286 PROG=0 (?) VERS=0 PROC=0

```
 0: 4500 0044 a0a2 0000 8011 2f67 ac19 0935    E..D....../g...5
16: ac19 0938 0c99 0267 0030 c24f dead e4ee    ...8...g.0.O....
32: 0000 0000 0000 0002 0000 0000 0000 0000    ................
48: 0000 0000 0000 0000 0000 0000 0000 0000    ................
64: 0000 0000                                  ....
```

1730 20:09:55.33285 scanner.net.18.106 -> my.net.25.68 ETHER Type=0800 (IP), size = 82 bytes
1730 20:09:55.33285 scanner.net.18.106 -> my.net.25.68 IP D=my.net.25.68 S=scanner.net.18.106 LEN=68, ID=41123
1730 20:09:55.33285 scanner.net.18.106 -> my.net.25.68 UDP D=616 S=3225 LEN=48
1730 20:09:55.33285 scanner.net.18.106 -> my.net.25.68 RPC C XID=3735938287 PROG=0 (?) VERS=0 PROC=0

```
 0: 4500 0044 a0a3 0000 8011 2f66 ac19 0935    E..D....../f...5
16: ac19 0938 0c99 0268 0030 c24d dead e4ef    ...8...h.0.M....
32: 0000 0000 0000 0002 0000 0000 0000 0000    ................
48: 0000 0000 0000 0000 0000 0000 0000 0000    ................
64: 0000 0000                                  ....
```

1731 20:09:55.33293 scanner.net.18.106 -> my.net.25.68 ETHER Type=0800 (IP), size = 82 bytes
1731 20:09:55.33293 scanner.net.18.106 -> my.net.25.68 IP D=my.net.25.68 S=scanner.net.18.106 LEN=68, ID=41124
1731 20:09:55.33293 scanner.net.18.106 -> my.net.25.68 UDP D=617 S=3225 LEN=48
1731 20:09:55.33293 scanner.net.18.106 -> my.net.25.68 RPC C XID=3735938288 PROG=0 (?) VERS=0 PROC=0

```
 0: 4500 0044 a0a4 0000 8011 2f65 ac19 0935    E..D...../e...5
16: ac19 0938 0c99 0269 0030 c24b dead e4f0    ...8...i.0.K....
32: 0000 0000 0000 0002 0000 0000 0000 0000    ................
48: 0000 0000 0000 0000 0000 0000 0000 0000    ................
64: 0000 0000                                  ....
```

1. Source of Trace.

snoop running on a Sun Sparc 10 with Solaris 2.6

snoop –x14 –s1500 -I infile host my.net.25.68

3. Probability the source address was spoofed:

This appears to be an UDP port scan. If it is, the source address cannot be spoofed or the attacker would not obtain any information.

4. Description of attack:

UPD scan for all udp ports on my.net.25.68

5. Attack mechanism:

The attacker sends udp probes to all devices on the attacked machine. If the attacked machine has a listener, it will not response. If it does not have a listener, then it will response with ICMP Destination unreachable (UDP port 600 unreachable)

6. Correlations:

I have not found an correlation for this scan. Certainly, the attacker is sending out the same package to each more on my.net.25.68. Also, a well formed RPC package would include the RPC type in bytes 41-44, ie, 00018xxx. However, it appears to be all zeros.

I did find a reference to UDP Port Scans in a publication by ISS on their attack signatures:

UDP Port Scan
Type Pre-attack probe
Console Name UDP_Port_Scan
Technical Description This check recognizes a portscan that is taking place on your network. A portscan is an attempt by an attacker to count the services running

on a machine by probing each port for a response. This vulnerability check detects a normal portscan as well as stealth scans (sometimes also referred to as Half Scans, SYN/ACK Scans, or FINscans). …

Why this is important This is an attempt by an intruder to determine how best to attack a system. By determining which services are running on a host, an intruder can direct an attack more effectively,
reducing the amount of time and effort required to gain unauthorized access.

False positives There are many legitimate applications that can appear to be a port scan. Therefore, you should investigate the initial events to determine whether they were legitimate or not.

Systems affected All hosts running UDP services.

What to do Identify the source of the port scan. Correlate this with the services that are running on the
target host. Is there a reasonable explanation? Identify the source of the scan as well as the intent behind the scan. You may want to take further precautions to protect the scanned devices. Check the access logs for indications of unauthorized access. If you do detect indications of unauthorized access, you should consider the system compromised and take appropriate action.

7. Evidence of active targeting:

Although many ports were referenced, only one machine was targeted, my.net.87.251.

8. Severity:

Target Criticality – 3 (Machine is a group Unix Server.)
Lethality – 2 (An scan like this can determine the active udp ports. With information like this, the attacker can customize an attack specific to what is running on this machine.)
System Countermeasures – 3, UDP traffic is generally accepted. Latest patches can minimize vulnerabilities.
Network Countermeasures – 1 (This probe originated internally. If this is an attack it is already inside our firewall
Severity - 3 (3+2) – (3+ 1) = 1 = Moderate

9. Defensive recommendation:

Per ISS: Identify the source of the port scan. Correlate this with the services that are running on the
target host. Is there a reasonable explanation? Identify the source of the scan as well as the intent behind the scan. You may want to take further precautions to protect

the scanned devices. Check the access logs for indications of unauthorized access. If you do detect indications of unauthorized access, you should consider the system compromised and take appropriate action.

10. Multiple choice test question:

What are legitimate RPC program numbers?

A) 1 to 1024
B) 100000 to 536870937
C) 1 to 65535
E) Any number less than 1000

Answer is B. RPC program numbers are located in bytes 12 –15 of the RPC Header that immediately follows the UDP headers. The currently defined values range from 10000 to 536870937. In the above scans, the program numbers were 0

Detect #4

```
4985 20:11:15.72408  my.net.42.25 -> intruder.net.ftp.62  FTP R port=3255
4986 20:11:15.72439  my.net.42.25 -> intruder.net.ftp.62  FTP R port=3235 530 Login
incorrect.
4987 20:11:15.72529  intruder.net.ftp.62 -> my.net.42.25  FTP C port=3235 USER
remote\r\n
4988 20:11:15.72769  my.net.42.25 -> intruder.net.ftp.62  FTP R port=3263 331 Password
require
4989 20:11:15.72873  intruder.net.ftp.62 -> my.net.42.25  FTP C port=3263 PASS
service\r\n
4990 20:11:15.73421  my.net.42.25 -> intruder.net.ftp.62  FTP R port=3263 530 Login
incorrect.
4991 20:11:15.73510  intruder.net.ftp.62 -> my.net.42.25  FTP C port=3263 USER rje\r\n
4992 20:11:15.74407  my.net.42.25 -> intruder.net.ftp.62  FTP R port=3238
4993 20:11:15.74746  my.net.42.25 -> intruder.net.ftp.62  FTP R port=3249 331 Password
require
4994 20:11:15.74853  intruder.net.ftp.62 -> my.net.42.25  FTP C port=3249 PASS
auditor\r\n
4995 20:11:15.75408  my.net.42.25 -> intruder.net.ftp.62  FTP R port=3253
4996 20:11:15.75484  my.net.42.25 -> intruder.net.ftp.62  FTP R port=3263 331 Password
require
4997 20:11:15.75522  my.net.42.25 -> intruder.net.ftp.62  FTP R port=3249 530 Login
incorrect.
4998 20:11:15.75591  intruder.net.ftp.62 -> my.net.42.25  FTP C port=3263 PASS rje\r\n
4999 20:11:15.75648  intruder.net.ftp.62 -> my.net.42.25  FTP C port=3249 USER root\r\n
```

5000 20:11:15.75992 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3244 331 Password require
5001 20:11:15.76095 intruder.net.ftp.62 -> my.net.42.25 FTP C port=3244 PASS admin\r\n
5002 20:11:15.76429 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3244 530 Login incorrect.
5003 20:11:15.76469 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3263 530 Login incorrect.
5004 20:11:15.76526 intruder.net.ftp.62 -> my.net.42.25 FTP C port=3244 USER root\r\n
5005 20:11:15.76578 intruder.net.ftp.62 -> my.net.42.25 FTP C port=3263 USER sam_exec\r\n
5006 20:11:15.77089 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3244 331 Password require
5007 20:11:15.77200 intruder.net.ftp.62 -> my.net.42.25 FTP C port=3244 PASS root\r\n
5008 20:11:15.77412 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3239
5009 20:11:15.77418 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3265
5010 20:11:15.78406 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3256
5011 20:11:15.79777 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3236 331 Password require
5012 20:11:15.79904 intruder.net.ftp.62 -> my.net.42.25 FTP C port=3236 PASS nonpriv\r\n
5013 20:11:15.80068 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3236 530 Login incorrect.
5014 20:11:15.80163 intruder.net.ftp.62 -> my.net.42.25 FTP C port=3236 USER school\r\n
5015 20:11:15.80409 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3250
5016 20:11:15.81056 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3244 230 User root logged
5017 20:11:15.81271 intruder.net.ftp.62 -> my.net.42.25 FTP C port=3244 PASV\r\n
5018 20:11:15.81575 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3259 331 Password require
5019 20:11:15.82055 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3260 331 Password require
5020 20:11:15.82259 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3245 331 Password require
5021 20:11:15.82414 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3235
5022 20:11:15.83301 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3241 530 Login incorrect.
5023 20:11:15.83343 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3247 530 Login incorrect.
5024 20:11:15.84385 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3251 530 Login incorrect.
5025 20:11:15.85091 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3261 331 Password require
5026 20:11:15.85177 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3257 530 Login incorrect.

5027 20:11:15.85409 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3249
5028 20:11:15.85945 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3238 331 Password require
5029 20:11:15.86062 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3256 530 Login incorrect.
5030 20:11:15.86219 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3242 530 Login incorrect.
5031 20:11:15.86409 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3263
5032 20:11:15.86470 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3264 530 Login incorrect.
5033 20:11:15.87356 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3253
5034 20:11:15.87399 intruder.net.ftp.62 -> my.net.42.25 FTP C port=3253
5035 20:11:15.88252 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3246 331 Password require
5036 20:11:15.88952 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3258 331 Password require
5037 20:11:15.89411 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3236
5038 20:11:15.90419 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3244
5039 20:11:15.93441 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3266 220 machname FTP ser
5040 20:11:15.93872 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3243 331 Password require
5041 20:11:15.94065 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3255 331 Password require
5042 20:11:15.94181 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3265 530 Login incorrect.
5043 20:11:15.95133 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3250 331 Password require
5046 20:11:15.95363 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3240 530 Login incorrect.
5047 20:11:15.97642 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3236
5048 20:11:15.97682 intruder.net.ftp.62 -> my.net.42.25 FTP C port=3236
5049 20:11:15.99607 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3235 331 Password require
5050 20:11:15.99730 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3249 331 Password require
5051 20:11:16.00380 my.net.42.25 -> intruder.net.ftp.62 FTP R port=3239 230 User oracle logg
5052 20:11:16.01518 intruder.net.ftp.62 -> my.net.42.25 FTP C port=3256

1. Source of Trace.

Company Network, created by ISS Internet Scanner

2. Detect was generated by:

snoop running on a Sun Sparc 10 with Solaris 2.6

snoop –ta –x14 –s1500 -I infile host my.net.42.25

3. Probability the source address was spoofed:

Source address was not spoofed. It appears that there is an attempt to crack passwords. This would need responses from the attached machine. If the source address were spoofed, then the response would not get back to the attacker.

4. Description of attack:

This is an attempt to guess passwords.

5. Attack mechanism:

The attacker is ftp'ing to various ports on the attacked machine. When prompted, the attacker tries a default password. If the password has not changed, he is in.

6. Correlations:

This detect was generated by ISS Internet Scanner.

7. Evidence of active targeting:

Only one machine was targeted in the scan above. An attempt was made to crack passwords on this machine.

8. Severity: (If this were an actual attack)

Target Criticality – 3 (Machine is a group Unix Server.)
Lethality – 5 (According to the scan, password or root and several ID's were successfully cracked.)
System Countermeasures – 3, Should be removed and randomized character generation should be used.
Network Countermeasures – 1 (This probe originated internally. If this is an attack it is already inside our firewall
Severity - 3 (3+5) – (3+ 1) = 4 = Severe

9. Defensive recommendation:

Anytime a system in installed, care must be taken to change default

passwords. Passwords should also be set to expire every 60-90 days and new ones generated. In this case, we ran Internet Scanner to determine login vulnerabilities. This scan found several.

10. Multiple choice test question:

Whats is the number one cause of unauthorized access to a server?

a) People leaving passwords taped to the side of their displays
b) Passwords that involved words or birthdates or other easily remembered character string are used.
c) Defaults passwords are retained after an application is installed.
d) Passwords are removed for easy access to the system during checkout and never put back on.

Answer: All of the above have contributed to unauthorized access at one time or another. It would be difficult to determine which is the most abused. Probably, writing the password down and leaving it posted on the display or lay next to the machine.

Detect #5

```
18727 20:19:10.88499  udp.scan.net.254 -> my.net.102.68  UDP D=278 S=4002 LEN=29
18728 20:19:10.88506  udp.scan.net.254 -> my.net.102.68  UDP D=279 S=4002 LEN=29
18729 20:19:10.88513  udp.scan.net.254 -> my.net.102.68  UDP D=280 S=4002 LEN=29
18730 20:19:10.88520  udp.scan.net.254 -> my.net.102.68  UDP D=281 S=4002 LEN=29
18731 20:19:10.88528  udp.scan.net.254 -> my.net.102.68  UDP D=282 S=4002 LEN=29
18732 20:19:10.88535  udp.scan.net.254 -> my.net.102.68  UDP D=283 S=4002 LEN=29
18733 20:19:10.88542  udp.scan.net.254 -> my.net.102.68  UDP D=284 S=4002 LEN=29
18734 20:19:10.88550  udp.scan.net.254 -> my.net.102.68  UDP D=285 S=4002 LEN=29
18735 20:19:10.88557  udp.scan.net.254 -> my.net.102.68  UDP D=286 S=4002 LEN=29
18736 20:19:10.88564  udp.scan.net.254 -> my.net.102.68  UDP D=287 S=4002 LEN=29
18737 20:19:10.88571  udp.scan.net.254 -> my.net.102.68  UDP D=288 S=4002 LEN=29
18738 20:19:10.88578  udp.scan.net.254 -> my.net.102.68  UDP D=289 S=4002 LEN=29
18739 20:19:10.88586  udp.scan.net.254 -> my.net.102.68  UDP D=290 S=4002 LEN=29
18740 20:19:10.88593  udp.scan.net.254 -> my.net.102.68  UDP D=291 S=4002 LEN=29
18741 20:19:10.88600  udp.scan.net.254 -> my.net.102.68  UDP D=292 S=4002 LEN=29
18742 20:19:10.88608  udp.scan.net.254 -> my.net.102.68  UDP D=293 S=4002 LEN=29
18743 20:19:10.88615  udp.scan.net.254 -> my.net.102.68  UDP D=294 S=4002 LEN=29
18744 20:19:10.88622  udp.scan.net.254 -> my.net.102.68  UDP D=295 S=4002 LEN=29
18745 20:19:10.88629  udp.scan.net.254 -> my.net.102.68  UDP D=296 S=4002 LEN=29
18746 20:19:10.88636  udp.scan.net.254 -> my.net.102.68  UDP D=297 S=4002 LEN=29
```

18747 20:19:10.88644 udp.scan.net.254 -> my.net.102.68 UDP D=298 S=4002 LEN=29
18748 20:19:10.88651 udp.scan.net.254 -> my.net.102.68 UDP D=299 S=4002 LEN=29
18749 20:19:10.88658 udp.scan.net.254 -> my.net.102.68 UDP D=300 S=4002 LEN=29
18750 20:19:11.08626 udp.scan.net.254 -> my.net.102.68 UDP D=301 S=4002 LEN=29
18751 20:19:11.08633 udp.scan.net.254 -> my.net.102.68 UDP D=302 S=4002 LEN=29
18752 20:19:11.08638 udp.scan.net.254 -> my.net.102.68 UDP D=303 S=4002 LEN=29
18753 20:19:11.08643 udp.scan.net.254 -> my.net.102.68 UDP D=304 S=4002 LEN=29
18754 20:19:11.08648 udp.scan.net.254 -> my.net.102.68 UDP D=305 S=4002 LEN=29
18755 20:19:11.08653 udp.scan.net.254 -> my.net.102.68 UDP D=306 S=4002 LEN=29
18756 20:19:11.08657 udp.scan.net.254 -> my.net.102.68 UDP D=307 S=4002 LEN=29
18757 20:19:11.08666 udp.scan.net.254 -> my.net.102.68 UDP D=308 S=4002 LEN=29
18758 20:19:11.08672 udp.scan.net.254 -> my.net.102.68 UDP D=309 S=4002 LEN=29
18759 20:19:11.08680 udp.scan.net.254 -> my.net.102.68 UDP D=310 S=4002 LEN=29
18760 20:19:11.08687 udp.scan.net.254 -> my.net.102.68 UDP D=311 S=4002 LEN=29
18761 20:19:11.08694 udp.scan.net.254 -> my.net.102.68 UDP D=312 S=4002 LEN=29
18762 20:19:11.08704 udp.scan.net.254 -> my.net.102.68 UDP D=313 S=4002 LEN=29
18763 20:19:11.08709 udp.scan.net.254 -> my.net.102.68 UDP D=314 S=4002 LEN=29
18764 20:19:11.08718 udp.scan.net.254 -> my.net.102.68 UDP D=315 S=4002 LEN=29
18765 20:19:11.08722 udp.scan.net.254 -> my.net.102.68 UDP D=316 S=4002 LEN=29
18766 20:19:11.08730 udp.scan.net.254 -> my.net.102.68 UDP D=317 S=4002 LEN=29
18767 20:19:11.08732 my.net.102.68 -> udp.scan.net.254 ICMP Destination unreachable (UDP port 301 unreachable)
18768 20:19:11.08739 udp.scan.net.254 -> my.net.102.68 UDP D=318 S=4002 LEN=29
18769 20:19:11.08745 udp.scan.net.254 -> my.net.102.68 UDP D=319 S=4002 LEN=29
18770 20:19:11.08752 udp.scan.net.254 -> my.net.102.68 UDP D=320 S=4002 LEN=29
18771 20:19:11.08759 udp.scan.net.254 -> my.net.102.68 UDP D=321 S=4002 LEN=29
18772 20:19:11.08767 udp.scan.net.254 -> my.net.102.68 UDP D=322 S=4002 LEN=29
18773 20:19:11.08774 udp.scan.net.254 -> my.net.102.68 UDP D=323 S=4002 LEN=29
18774 20:19:11.08781 udp.scan.net.254 -> my.net.102.68 UDP D=324 S=4002 LEN=29
18775 20:19:11.08788 udp.scan.net.254 -> my.net.102.68 UDP D=325 S=4002 LEN=29

1. Source of Trace.

Company Network, created by ISS Internet Scanner

2. Detect was generated by:

snoop running on a Sun Sparc 10 with Solaris 2.6

snoop –ta –x14 –s1500 -I infile host my.net.102.68

3. Probability the source address was spoofed:

This was another type of UDP port scanning. This scan was captured by a

snoop trace during a scan of a server for vulnerabilities. Since we know who the source is, this is not a spoofed address. Also, if an attacker wants to know available ports, he would not spoof the address.

4. Description of attack:

This was a scan for udp port scan by Internet Security Systems' Internet Scanner. This probe was launched against all ports on my.net.102.68. The main difference is that the source port numbers remain the same (4002)

5. Attack mechanism:

This was a scan for udp port scan by Internet Security Systems' Internet Scanner. This probe was launched against all ports on my.net.102.68. The main difference is that the source port numbers remain the same (4002). Ports that are have listeners do not respond. Ports that do not have listeners respond with ICMP Destination unreachable (UDP port 301 unreachable).

6. Correlations:

I have not found an correlation for this scan. Certainly, the attacker is sending out the same package to each more on my.net.102.68. Also, a well formed RPC package would include the RPC type in bytes 41-44, ie, 00018xxx. However, it appears to be all zeros.

I did find a reference to UDP Port Scans in a publication by ISS on their attack signatures:

UDP Port Scan
Type Pre-attack probe
Console Name UDP_Port_Scan
Technical Description This check recognizes a portscan that is taking place on your network. A portscan is an attempt by an attacker to count the services running on a machine by probing each port for a response. This vulnerability check detects a normal portscan as well as stealth scans (sometimes also referred to as Half Scans, SYN/ACK Scans, or FINscans). …

Why this is important This is an attempt by an intruder to determine how best to attack a system. By determining which services are running on a host, an intruder can direct an attack more effectively,
reducing the amount of time and effort required to gain unauthorized access.

False positives There are many legitimate applications that can appear to be a port scan. Therefore, you should investigate the initial events to determine whether they were legitimate or not.

Systems affected All hosts running UDP services.

What to do Identify the source of the port scan. Correlate this with the services that are running on the
target host. Is there a reasonable explanation? Identify the source of the scan as well as the intent behind the scan. You may want to take further precautions to protect the scanned devices. Check the access logs for indications of unauthorized access. If you do detect indications of unauthorized access, you should consider the system compromised and take appropriate action.

7. Evidence of active targeting:

Although many ports were referenced, only one machine was targeted, my.net.102.68

8. Severity:

Target Criticality – 3 (Machine is a group Unix Server.)
Lethality – 2 (An scan like this can determine the active udp ports. With information like this, the attacker can customize an attack specific to what is running on this machine.)
System Countermeasures – 3, UDP traffic is generally accepted. Latest patches can minimize vulnerabilities.
Network Countermeasures – 1 (This probe originated internally. If this is an attack it is already inside our firewall
Severity - 3 (3+2) – (3+ 1) = 1 = Moderate

9. Defensive recommendation:

Per ISS: Identify the source of the port scan. Correlate this with the services that are running on the
target host. Is there a reasonable explanation? Identify the source of the scan as well as the intent behind the scan. You may want to take further precautions to protect the scanned devices. Check the access logs for indications of unauthorized access. If you do detect indications of unauthorized access, you should consider the system compromised and take appropriate action.

10. Multiple choice test question:

What are legitimate RPC program numbers?
D) 1 to 1024
E) 100000 to 536870937
F) 1 to 65535
F) Any number less than 1000

Answer is B. RPC program numbers are located in bytes 12 –15 of the RPC Header that immediately follows the UDP headers. The currently defined values range from

10000 to 536870937. In the above scans, the program numbers were 0

Assignment 2 - Describe the State of Intrusion Detection

Write a white paper on any single intrusion detection technology or challenge. You may choose any IDS, IDS technology or approach, or network pattern; or you may choose any attack, reconnaissance technique, denial of service, or exploit that operates across a network or within a host system.

The State of Intrusion Detection Systems for Corporate America

What is "IDS"? This is a question that I had 120 days ago as I was approached to lead a team to analyze existing intrusion detection systems and recommend one for installation at our company. Prior to this, I had work for many years as a mainframe systems programmer and several other years as a Unix administrator. The last few years I was responsible to maintaining both the server and application running on a Sun Enterprise 3500 server. During this time, the biggest thing I had to worry about was my own administration staff making unauthorized modification to the application and slowing performance for our customers. I had never heard of "Denial of Service" or that we have been responsible for it in our own system until my manager started asking questions about "yesterday's" slowdown.

However, all of that started changing when our CIO attended a seminar and was asked what our company was doing about intrusion detection. The CIO with more of a financial background than a technical one, had never heard of intrusion detection systems and hardly knew that, like most companies today, our internal networks and our business zones were protected solely by firewalls. As soon as he got back, he asked the IT Security manager to discuss intrusion detection with him and wanted to know what we were doing about it. The application I supported was in sustain mode; I was determined to be the one who could bring IDS into our network. I suddenly became the intrusion detection expert at the company. The problem was, I knew nothing about it. And the IT Security Manager didn't know enough about it to be able to discuss even what we wanted to do with it once we had it installed.

Since our company did not have an IDS, I made the assumption that this was an infant concept and that there would be only a few systems to consider. I thought I would be on loan to the Security Manager for a few months and then I would be back doing what I was doing before. So where did I go to find what information I could on intrusion detection? The Internet, of course. A quick MSN scan (www.msn.com) of intrusion detection systems, however, produced a list of over 22,000 entries for both physical and data intrusion detection references. I almost went into a panic. The concept, at least, was

not the infant idea that I thought it was. During my search, I saw a reference on the first page of hits to Michael Sobirey's Intrusion Detection Systems page (http://www-rnks.informatik.tu-cottbus.de/~sobirey/ids.html) containing a list of 92 host and network based Intrusion Detection (& Response) Systems. That reduced the 22,000 down to under a hundred and I figured that this was a good start.

The list was, indeed, a good start. It introduced me to both intrusion detection projects and systems (both non-commercial and commercial). Other documents convinced me that both host and network IDS's were essential in protecting our network.

Having been in Mainframe support for many years and perhaps still under the influence of IBM's proprietary system programs, I quickly decided that the best products for corporate use are commercial products. This, in part, was based on the fact that my company does not want to have to generate code or even maintain programs that is uses. This is a philosophy generally accepted in the commercial world today: It is cheaper to buy a program that contains 80% of needed functionality and have the commercial vendor maintain the program than to hire and maintain a programming staff to provide that functionality. This philosophy immediately reduced the articles to about 20. Then came the revelation. Commercial Intrusion Detection Systems **are** in its infancy.

Drawing on my knowledge of the company after 20+ years of employment, I generated a list of requirements to be used in evaluating IDS. My company is a global manufacturing company with assembly plants all over the world. Each manufacturing entity has its own LAN with a network support staff to monitoring it. Many have their own Internet access points, with DMZ's and Business Zones. And while the local network techs maintain support over their own LAN, network design and second and third level of support for the subnets are centered at the corporate headquarters. The idea of centralized IT has been a way of life at this company for years. Even distributed mainframes that were in use in the 70's and 80's were maintained by a corporate level staff housed at the corporate headquarters.

So based on my 20+ years of experience, I determined the following requirements for any IDS we would acquire for installation into our network:

- Centralized Management of all detectors, located at the corporate headquarters
- Distributed Monitoring, with consoles located in each of the distributed subnets with a collector at or near the management console at the corporate headquarters.
- Network Monitoring capable of handling 100 Mbps line speeds with plans to pursue 1Gbps
- Minimal impact on network reporting suspected intrusions back to monitors
- Host detectors with small footprint and taking little CPU to perform its monitoring duties

- Management and Monitoring or Network and Host sensors would have to be integrated, i.e., look very closely like one another (or the same product) so that a very small staff (of 3-5) could provide Level 2 and Level 3 support and all policy management of the detectors.
- Our company employs widespread usage of VLANs and we must be able to monitor network traffic in the VLANS.
- Our company mostly employs Sun Enterprise Servers, Solaris releases 2.6, 2.7 and migrating to Solaris 8 in our Business Zones and DMZs but we also have Compac servers running Windows NT 4.0 and migrating to Windows 2000. We must be able to harden and protect these servers with a host detector.
- Since many IDS are signature based, we must partner with one that has regular signature updates and has the ability to centrally update the detectors with the new signature list. And we must be able to write our own signatures to protect our network after an attack has been identified and before our vendor can give us an signature update
- Since network sensors will be deployed in areas where network and server expertise is limited, an appliance with the sensor already installed and the OS hardened would be critical.
- The product must scale to a enterprise the size of my company.
- The product must have the capabilities to reduce false positives.

The last item was of particular importance. As indicated above, my company is a global corporation, employing 50,000 worldwide, with manufacturing interests in Europe, Israel, India, Southeast Asia, China, Japan and Korea. We are expecting to deploy 4-5 thousand network/host sensors worldwide to protect our intranet, our Business Zones, our DMZ's, our VPN gateways, and even internal server farms that contain critical corporate information. One console for every 50 sensors would not work. We would need 5,000 sensors per one management console. We would need a centralized monitoring console that could at least catch the most critical of the intrusions, while we could employ multiple monitoring consoles for local support teams. company to set up policies to reduce the noise level for alerts to a handful per hour or even per day so that they would not overwhelm the network around the centralized monitor and so they would overwhelm the small support staff looking at the alerts.

Having decided on our requirements, I once again referred to Michael Sobirey's Intrusion Detection Systems paper. I eliminated the open systems and the research projects. I started contacting several vendors and invited them in. What I found was that IDS really is in its infancy. Except for companies like ISS, AXENT, CISCO and NFR, most companies were either not yet profitable or their product has not been on the market for six months. These products I eliminated for the reason, with the number of sensors we plan to deploy, I could not afford to go with a company or product that had not proven itself in the market. In addition, most of these companies only had a few people devoted to identifying new attacked and supporting the product. My company expects 7X24 support if we spend the money we expect to with the selected.

Talking to the vendors, most of the more reliable companies office full time staff devoted to identifying net attacks and producing updated signature files to accommodate them. Unfortunately, some of the companies seem to think that updates coinciding with product upgrades is sufficient time to update signatures. However, we are expecting monthly updates, similar to our virus protection provider, and more frequently if the conditions call for it. Some of the products we evaluated had easy wizards available to assist in creating customized signatures. This is a very valuable feature that could allow us to create our own signature for new attacks if the selected vendor cannot provide us with an update soon enough. Sadly, we found a number of products that do not allow customized signatures. Some companies appear to have compiled their signature files for quicker execution. While speed is fine, without this feature, their product was considered incomplete. While some products use Windows based wizards to assist in creating new signatures, other provide a programming language. These products allow for the creation of very specific and very efficient signature checks, it does require the support learning a new programming language. These products we considered more difficult to support and maintain and were given lower marks.

Hooks and shims! While we evaluated some products, we asked how the identify and prevent mechanism works. Some products uses hooks in the kernel to intercept system calls and can, therefore, prevent commands from executing. However, a hooked kernel implies dependency on the IDS vendor for upgrades to our servers. This was a feature that was not well received by our Windows Change Management group. But what about shims? Somehow shims are different from hooks. At least the word sounds better. The different I was given is that shims, rather than intercepting system calls in the kernel, looks at the IP stack and responds to generated processes after they have been initialized but before they can do any harm. While the hooks may be more efficient, shims sound like a better way to go. In fact even these products were given lesser ratings that products that only monitored system logs for intrusion or abuse detection. However, even some of these products used OS built-in monitoring to assist with detection. One product requires Solaris' Basic Security Module (BSM) to be enabled so it can monitor the BSM audit logs. This is fine, but we have no installation in house that has BSM enable. And to enable it, requires a reboot of the server, a event we don't talk much about. Most of our metrics are based on 99.97% uptime and to schedule a down just to turn on a feature will not win us friends.

As the product evaluations continue, scaling becomes the big issue. One of our requirements is for centralized management and monitoring of policies and events. This means that alerts from 5000+ sensors must filter back to a single management console. While most vendors say their products can easily accommodate such numbers, industry publications repeated state that there is a practical limit of 50 sensors per management consoles. We have installed several products and each one requires pushing each policy manually to each sensor affected. This feature alone will support the 50:1 statement and implies that the 2 men support staff we currently have in place to manage the IDS will be immensely understaffed.

Another critical requirement is the network appliance. Some of our deployments will be into areas that lack expertise in network technology and server administration. It became very clear that the vendor of our NIDS must have an appliance. Vendors like Cisco and ISS offer appliances which only need an ip address to connect into the network. Remote administration can be performed from our central support area. While NFR does not offer an appliance, they do offer a bootable CD with the network sensor and hardened OS already configured so all you have to do is reboot. Other vendors like Symantec/AXENT are working on an appliance for a future release.

And that brings me to the most dreaded words I have heard from each vendor: "We will have that next release." It is amazing how many of these vendors who had never thought about certain features suddenly state "We will have them in the next release". However, since no product on the market meets 100% of our requirements, we must depend on those words to be true. Our initial IDS deployment will only consist of 2 network sensors and two host sensors. We will have six weeks to fine-tune these products to reduce false positives to a level supportable by our two man support staff. By then, the "next release" will be out. We will probably reevaluate our decision at that time to see if the select products remains our IDS of choice or whether we will have to struggle until "next release" to get a product that meets our requirements.

In conclusion, IDS is the current industry keyword for network/date protection. Many major corporations are frantically evaluating and installing IDS. Many of these companies can and do divide their network into manageable subnets. However, from my perspective, few products meet our needs. And if we cannot find a product that can scale to the level that we think we need, we may have to go away from the commercial software. Anyone know anything about snort?

References:

Michael Sobirey's Intrusion Detection Systems page (http://www-rnks.informatik.tu-cottbus.de/~sobirey/ids.html)

List of IDS evaluated and under consideration:
Internet Security Systems – RealSecure IDS
Symantec/AXENT – NetProwler/Intruder Alert IDS
Clicknet – Entercept IDS
Cisco – CiscoSecure IDS
Network Flight Recorder
Intrusion.com – SecureNet Pro
Network Associates – CyberCop

ISS - Evaluating an Intrusion Detection Solution - A Strategy for a Successful IDS Evaluation (http://documents.iss.net/literature/RealSecure/ids_eval.pdf)

Assignment 3 - "Analyze This"

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 rulebase. From time to time, the power has failed or the disk was full so you do not have data for all days.

I pulled the information provided by GIAC Enterprises and evaluated them using grep and awk on my Unix machine. If found snort trapped on 26 different rules:

1. DNS udp DoS attack described on unisog
2. STATDX UDP attack
3. connect to 515 from inside
4. connect to 515 from outside
5. spp_portscan: portscan status from
6. Watchlist 000220 IL-ISDNNET-990517
7. Watchlist 000222 NET-NCFC
8. SITE EXEC - Possible wu-ftpd exploit - GIAC000623
9. TCP SMTP Source Port traffic
10. Happy 99 Virus
11. Tiny Fragments - Possible Hostile Activity
12. Back Orifice
13. External RPC call
14. Attempted Sun RPC high port access
15. NMAP TCP ping!
16. site exec - Possible wu-ftpd exploit - GIAC000623
17. SMB Name Wildcard
18. Russia Dynamo - SANS Flash 28-jul-00
19. Broadcast Ping to subnet 70
20. Queso fingerprint
21. WinGate 1080 Attempt
22. Probable NMAP fingerprint attempt
23. Null scan!
24. SUNRPC highport access!
25. SNMP public access
26. SYN-FIN scan! .

I pulled the latest rule set from www.snort.org. It had some of the rules. However, I had to guess at others. I have tried to give a brief explanation of what each rule was looking for, citing references from the SANS Institute,

www.snort.org, articles on intrusion detection and selected exploit references found the the web. I have evaluated each alert detected and have give examples of the detected alert plus my comments. I have tried to correlate the alerts to the scans and the OOS Checks as much as possible. My findings are listed below.

spp_portscan

Throughout the approximately 2 months of scans that were available, there were number port scans (approximately 38269). Most of these were either from a single external ip address making a single scan or from inside my.net. my.net.217.158 was very active on Jan 12 and Jan 13. A scan of the snort alerts indicated it was the source of port scans almost 5000 times. A look at the snort scans also indicated my.net.217.158 was communication to several external addresses:

```
MY.NET.217.158:2496 -> 208.245.58.6:1788 SYN **S*****
MY.NET.217.158:2497 -> 208.245.58.6:1789 SYN **S*****
MY.NET.217.158:2501 -> 208.245.58.6:1789 SYN **S*****
MY.NET.217.158:2500 -> 208.245.58.6:1788 SYN **S*****
MY.NET.217.158:2340 -> 24.190.76.73:1375 NOACK *1**RP*U RESERVEDBITS
MY.NET.217.158:2503 -> 208.245.58.6:1789 SYN **S*****
MY.NET.217.158:2500 -> 208.245.58.6:1788 SYN **S*****
MY.NET.217.158:1651 -> 142.103.53.239:8874 NOACK *1*FRP*U RESERVEDBITS
MY.NET.217.158:2509 -> 208.245.58.6:1789 SYN **S*****
MY.NET.217.158:2508 -> 208.245.58.6:1788 SYN **S*****
MY.NET.217.158:2340 -> 24.115.202.230:1098 FULLXMAS 21SFRPAU
RESERVEDBITS
MY.NET.217.158:2521 -> 208.245.58.6:1789 SYN **S*****
MY.NET.217.158:2340 -> 216.86.197.123:3187 NOACK 2*S*R**U RESERVEDBITS
MY.NET.217.158:1651 -> 142.103.53.239:8874 NOACK *1*FRP*U RESERVEDBITS
MY.NET.217.158:2340 -> 128.54.192.46:4910 UNKNOWN 21****A* RESERVEDBITS
MY.NET.217.158:0 -> 24.190.76.73:2340 NOACK **S*RP*U
MY.NET.217.158:2340 -> 216.86.197.123:3187 NOACK 2*S*R**U RESERVEDBITS
MY.NET.217.158:2340 -> 24.115.202.230:1098 NOACK *1*FR**U RESERVEDBITS
MY.NET.217.158:2340 -> 24.190.76.73:1332 FIN ***F****
MY.NET.217.158:0 -> 142.103.53.239:1651 NOACK *1*FRP*U RESERVEDBITS
MY.NET.217.158:2528 -> 208.245.58.6:1789 SYN **S*****
MY.NET.217.158:2340 -> 193.253.187.173:1051 NOACK 21SF*P** RESERVEDBITS
MY.NET.217.158:2340 -> 24.190.76.73:1332 NOACK **S*RP*U
MY.NET.217.158:2527 -> 208.245.58.6:1788 SYN **S*****
MY.NET.217.158:1651 -> 142.103.53.239:8874 NULL ********
MY.NET.217.158:16 -> 193.253.187.173:2340 NOACK 21SF*P** RESERVEDBITS
MY.NET.217.158:2340 -> 24.190.76.73:1375 NOACK *1**RP*U RESERVEDBITS
MY.NET.217.158:2536 -> 208.245.58.6:1789 SYN **S*****
MY.NET.217.158:2535 -> 208.245.58.6:1788 SYN **S*****
MY.NET.217.158:2340 -> 24.43.190.17:1497 NOACK *1S**P*U RESERVEDBITS
MY.NET.217.158:2340 -> 24.115.202.230:1098 NOACK *1*FR**U RESERVEDBITS
MY.NET.217.158:2548 -> 208.245.58.6:1789 SYN **S*****
MY.NET.217.158:2547 -> 208.245.58.6:1788 SYN **S*****
MY.NET.217.158:2340 -> 24.115.202.230:1098 INVALIDACK *1S***AU
RESERVEDBIT
```

Whereas the SYN's could be indicative of a start of a three way handshake, it appears that it was definitely trying to contact 24.9.25.103 (cc483908-a.gscrk1.sc.home.com) a cable modem address. .

DNS udp DoS attack described on unisog

On Jan 06, we saw an increased number of attacks on DSN.

01/06-19:09:55.821832 [**] DNS udp DoS attack described on unisog [**] 209.67.50.203:16208 -> MY.NET.1.4:53
01/06-19:09:56.025468 [**] DNS udp DoS attack described on unisog [**] 209.67.50.203:18146 -> MY.NET.1.5:53
01/06-19:09:56.071497 [**] DNS udp DoS attack described on unisog [**] 209.67.50.203:17306 -> MY.NET.1.3:53
01/06-19:09:56.681048 [**] DNS udp DoS attack described on unisog [**] 209.67.50.203:14928 -> MY.NET.1.3:53
01/06-19:09:56.839929 [**] DNS udp DoS attack described on unisog [**] 209.67.50.203:11441 -> MY.NET.1.3:53
01/06-19:09:56.839986 [**] DNS udp DoS attack described on unisog [**] 209.67.50.203:29792 -> MY.NET.1.3:53
01/06-19:09:57.540083 [**] DNS udp DoS attack described on unisog [**] 209.67.50.203:1146 -> MY.NET.1.5:53
01/06-19:09:57.663543 [**] DNS udp DoS attack described on unisog [**] 209.67.50.203:24184 -> MY.NET.1.4:53
01/06-19:09:57.777517 [**] DNS udp DoS attack described on unisog [**] 209.67.50.203:28719 -> MY.NET.1.3:53
01/06-19:09:58.159759 [**] DNS udp DoS attack described on unisog [**] 209.67.50.203:29894 -> MY.NET.1.4:53
01/06-19:09:59.307287 [**] DNS udp DoS attack described on unisog [**] 209.67.50.203:23991 -> MY.NET.1.4:53
01/06-19:09:59.928731 [**] DNS udp DoS attack described on unisog [**] 209.67.50.203:12149 -> MY.NET.1.4:53
01/06-19:10:00.021295 [**] DNS udp DoS attack described on unisog [**] 209.67.50.203:29626 -> MY.NET.1.4:53
01/06-19:10:00.358111 [**] DNS udp DoS attack described on unisog [**] 209.67.50.203:5083 -> MY.NET.1.5:53
01/06-19:10:01.022567 [**] DNS udp DoS attack described on unisog [**] 209.67.50.203:22107 -> MY.NET.1.4:53
01/06-19:10:01.924105 [**] DNS udp DoS attack described on unisog [**] 209.67.50.203:1615 -> MY.NET.1.3:53
01/06-19:10:02.143881 [**] DNS udp DoS attack described on unisog [**] 209.67.50.203:12760 -> MY.NET.1.4:53

This appears to be a DoS attack against domain servers my.net.1.3, my.net.1.4, my.net.1.5. There were over 36000 of these packets hitting the three dns servers in two hours. This attack was described in the unisog University Security mailing list. There was no identification as to the origin of this attack. However, address 209.67.50.203 belongs to futuresite.register.com. register.com is a web based domain name registration service. Per unisog email from Glenn Forbes Fleming Larratt <glratt@rice.edu> , he asked “Have others seen a steady stream of DNS requests (~220/min/DNS server) from 209.67.50.203? On Thursday I

talked to Mathew Zito, the admin for this address, who said they are not the source but the target of a DoS > attack. He claimed they are seeing 60 - 90 Mb data stream > toward 209.67.50.203. On Thursday I blocked 209.67.50.0 at our boarder > routers. After ~ 48 hours these DNS requests are still coming.

---

connect to 515 from outside; connect to 515 from inside

12/15-00:24:40.072989 [**] connect to 515 from outside [**]
141.211.176.99:2767 -> MY.NET.1.207:515
12/15-00:24:40.074590 [**] connect to 515 from outside [**]
141.211.176.99:2785 -> MY.NET.1.225:515
12/15-00:24:40.076065 [**] connect to 515 from outside [**]
141.211.176.99:2801 -> MY.NET.1.241:515
12/15-00:24:40.076397 [**] connect to 515 from outside [**]
141.211.176.99:2808 -> MY.NET.1.248:515
12/15-00:24:40.076452 [**] connect to 515 from outside [**]
141.211.176.99:2813 -> MY.NET.1.253:515
12/15-00:24:43.066015 [**] connect to 515 from outside [**]
141.211.176.99:2707 -> MY.NET.1.147:515
12/15-00:24:43.066063 [**] connect to 515 from outside [**]
141.211.176.99:2711 -> MY.NET.1.151:515
12/15-00:24:43.068670 [**] connect to 515 from outside [**]
141.211.176.99:2743 -> MY.NET.1.183:515
12/15-00:24:43.068831 [**] connect to 515 from outside [**]
141.211.176.99:2745 -> MY.NET.1.185:515
12/15-00:24:43.068883 [**] connect to 515 from outside [**]
141.211.176.99:2746 -> MY.NET.1.186:515
12/15-00:24:43.068935 [**] connect to 515 from outside [**]
141.211.176.99:2748 -> MY.NET.1.188:515
12/15-00:24:43.068986 [**] connect to 515 from outside [**]
141.211.176.99:2749 -> MY.NET.1.189:515
12/15-00:24:43.069206 [**] connect to 515 from outside [**]
141.211.176.99:2754 -> MY.NET.1.194:515
12/15-00:24:43.069258 [**] connect to 515 from outside [**]
141.211.176.99:2755 -> MY.NET.1.195:515
12/15-00:24:43.069308 [**] connect to 515 from outside [**]
141.211.176.99:2756 -> MY.NET.1.196:515
12/15-00:24:43.073752 [**] connect to 515 from outside [**]
141.211.176.99:2807 -> MY.NET.1.247:515
12/15-00:24:43.073806 [**] connect to 515 from outside [**]
141.211.176.99:2808 -> MY.NET.1.248:515
12/15-00:24:43.075098 [**] connect to 515 from outside [**]
141.211.176.99:2813 -> MY.NET.1.253:515
12/15-00:24:48.069194 [**] connect to 515 from outside [**]
141.211.176.99:2952 -> MY.NET.2.137:515

12/15-00:24:48.071755 [**] connect to 515 from outside [**]
141.211.176.99:2987 -> MY.NET.2.172:515
12/15-00:24:48.073346 [**] connect to 515 from outside [**]
141.211.176.99:3010 -> MY.NET.2.195:515
12/15-00:24:48.073403 [**] connect to 515 from outside [**]
141.211.176.99:3011 -> MY.NET.2.196:515
12/15-00:24:48.074945 [**] connect to 515 from outside [**]
141.211.176.99:3025 -> MY.NET.2.210:515
12/15-00:25:00.068534 [**] connect to 515 from outside [**]
141.211.176.99:3353 -> MY.NET.4.19:515
12/15-00:25:00.068586 [**] connect to 515 from outside [**]
141.211.176.99:3354 -> MY.NET.4.20:515
12/15-00:25:00.074941 [**] connect to 515 from outside [**]
141.211.176.99:3428 -> MY.NET.4.94:515
12/15-00:25:00.074997 [**] connect to 515 from outside [**]
141.211.176.99:3429 -> MY.NET.4.95:515
12/15-00:25:00.075050 [**] connect to 515 from outside [**]
141.211.176.99:3430 -> MY.NET.4.96:515
12/15-00:25:00.075106 [**] connect to 515 from outside [**]
141.211.176.99:3432 -> MY.NET.4.98:515
12/15-00:25:07.065387 [**] connect to 515 from outside [**]
141.211.176.99:3465 -> MY.NET.4.131:515
12/15-00:25:07.065447 [**] connect to 515 from outside [**]
141.211.176.99:3468 -> MY.NET.4.134:515
12/15-00:25:08.857950 [**] connect to 515 from outside [**]
141.211.176.99:3741 -> MY.NET.5.147:515
12/15-00:25:08.857999 [**] connect to 515 from outside [**]
141.211.176.99:3746 -> MY.NET.5.152:515
12/15-00:25:12.070840 [**] connect to 515 from outside [**]
141.211.176.99:3810 -> MY.NET.5.214:515
12/15-00:25:12.070895 [**] connect to 515 from outside [**]
141.211.176.99:3811 -> MY.NET.5.215:515
12/15-00:25:12.070948 [**] connect to 515 from outside [**]
141.211.176.99:3813 -> MY.NET.5.217:515
12/15-00:25:12.071003 [**] connect to 515 from outside [**]
141.211.176.99:3814 -> MY.NET.5.218:515
12/15-00:25:12.071057 [**] connect to 515 from outside [**]
141.211.176.99:3815 -> MY.NET.5.219:515
12/15-00:25:12.071111 [**] connect to 515 from outside [**]
141.211.176.99:3816 -> MY.NET.5.220:515
12/15-00:25:12.071188 [**] connect to 515 from outside [**]
141.211.176.99:3817 -> MY.NET.5.221:515
12/15-00:25:12.072637 [**] connect to 515 from outside [**]
141.211.176.99:3833 -> MY.NET.5.237:515
12/15-00:25:12.072978 [**] connect to 515 from outside [**]
141.211.176.99:3840 -> MY.NET.5.244:515
12/15-00:25:12.073034 [**] connect to 515 from outside [**]
141.211.176.99:3841 -> MY.NET.5.245:515

Next, I found a large number of traffic trying to access port 515, generally used for print spooling. While it would not be wise to allow outsiders to use your printers, if that was happening, then I would have expected to see only one a

handful of destinations in my.net. However, there were attempts to access over 2800 different addresses in my.net between December 15 and December 20. This appears to be a scan to see which servers are listening on port 515 possibly for a future attack. Of the 4200 hits, the top three offenders were 141.211.176.99 (vishuman28.us.itd.umich.edu) over 2200 times, 216.119.15.88 (name not resolved) over 1200 times, and 209.217.166.69 (name not resolved) just under 1000. The inside connects probably are valid since most of these were from my.net to my.net.

STATDX UDP

The STATDX UDP exploit is described by George Bakos in the practical for GIAC from SANS Security DC 2000: “The rpc.statd is the NFS file lock status reporter. Its function is to track NFS connections with requests to the rpc.lockd. In the event of a server going down, the rpc.statd will attempt to reestablish those locks by communicating the server's status to the NFS client's lock manager. There is a process of the rpc.statd which passes logging information using the syslog() function. The format string passed is user supplied data, with a UID:GID of 0:0, without any proper bounds checking. It is possible, and proven, that this buffer could be overflowed, placing executable code into the process address space and overwriting the process return address, forcing the execution of that code. This is commonly known as "smashing the stack". An excellent discussion on this theory and practice by Aleph One was published in Phrack #49

There was on hit by the STATDX UPD rule.
01/06-06:39:35.583605 [**] STATDX UDP attack [**] 206.210.80.6:1074 -> MY.NET.6.15:32776
And though this could be an indication of an attack, there was not additional information to support it.

SITE EXEC - Possible wu-ftpd exploit - GIAC000623

From CIAC000623, Wuarchive-ftpd, more affectionately known as wu-ftpd, is a replacement ftp daemon for Unix systems developed at Washington University (*.wustl.edu) by Bryan D. O'Connor. (*who is no longer working on it or supporting it!*) Wu-ftpd is the most popular ftp daemon on the Internet, used on many anonymous ftp sites all around the world. (Michael Sparks, November 21, 2000). Here he describes the possibility of improper use of code used by the ftp daemon can allow a malicious remote ftp client to subvert an FTP server. When this improper use is performed, remote system access is possible.

There were three hits by the SITE EXEC rule.
11/26-17:30:50.939661 [**] site exec - Possible wu-ftpd exploit - GIAC000623 [**] 24.23.255.246:4507 -> MY.NET.130.98:21
12/16-12:21:46.219962 [**] SITE EXEC - Possible wu-ftpd exploit - GIAC000623 [**] 209.1.62.94.11:4584 -> MY.NET.156.127:21

12/21-15:26:29.595664 [**] site exec - Possible wu-ftpd exploit - GIAC000623 [**] 64.217.116.106:1684 -> MY.NET.97.162:21

However, again, there was no additional documentation that would suggest that something was not right

---

SNMP public access

The Simple Network Management Protocol, SNMP, is a commonly used service that provides network management and monitoring capabilities. SNMP is vulnerable because it is often automatically installed on many network devices with "public" as the read string and "private" as the write string.

In the scans, there are entries similar to the following:

01/11-18:12:07.868642 [**] SNMP public access [**] 128.183.38.30:1032 -> MY.NET.154.26:161^M
01/11-18:13:04.645805 [**] SNMP public access [**] MY.NET.111.156:3887 -> MY.NET.50.154:161^M
01/11-18:13:04.732204 [**] SNMP public access [**] MY.NET.111.156:3887 -> MY.NET.50.154:161^M
01/11-18:13:04.843710 [**] SNMP public access [**] MY.NET.111.156:3887 -> MY.NET.50.154:161^M
01/11-18:13:05.849834 [**] SNMP public access [**] MY.NET.111.156:3887 -> MY.NET.50.154:161^M
01/12-00:03:47.201533 [**] SNMP public access [**] MY.NET.111.156:2401 -> MY.NET.50.154:161^M
01/12-01:42:52.548628 [**] SNMP public access [**] MY.NET.162.201:1819 -> MY.NET.50.154:161^M
01/12-01:42:52.645350 [**] SNMP public access [**] MY.NET.162.201:1819 -> MY.NET.50.154:161^M
01/12-01:42:53.845474 [**] SNMP public access [**] MY.NET.162.201:1819 -> MY.NET.50.154:161^M

and

01/12-09:31:41.697088 [**] SNMP public access [**] 128.46.156.231:1030 -> MY.NET.100.206:161^M
01/12-09:32:02.380437 [**] SNMP public access [**] 128.46.156.231:1092 -> MY.NET.100.143:161^M
01/12-09:32:04.048761 [**] SNMP public access [**] 128.46.156.231:1093 -> MY.NET.100.143:161^M
01/12-09:32:04.082561 [**] SNMP public access [**] 128.46.156.231:1093 -> MY.NET.100.143:161^M
01/12-09:32:04.134998 [**] SNMP public access [**] 128.46.156.231:1094 -> MY.NET.100.143:161^M
01/12-09:32:10.408144 [**] SNMP public access [**] 128.46.156.231:1096 -> MY.NET.100.99:161^M

01/12-09:32:10.649334 [**] SNMP public access [**] 128.46.156.231:1097 -> MY.NET.100.99:161^M
01/12-09:32:16.978157 [**] SNMP public access [**] 128.46.156.231:1100 -> MY.NET.100.99:161^M
01/12-09:32:17.639995 [**] SNMP public access [**] 128.46.156.231:1102 -> MY.NET.100.99:161^M

The events that originate from my.net probably is probably not significant. Typically, h/w and s/w monitoring programs will communicate back to a server using SNMP. As long as this is originating within my.net as the first series indicates it is probably OK. However, in the second series, an outsider, 128.46.156.231 (ece156-dhcp-2.ecn.purdue.edu) appears to be searching for a machine that listens on port 161. This could be the preliminary reconnaissance for an attack.

SYN-FIN scan!

Snort Alerts
11/28-20:03:02.051978 [**] SYN-FIN scan! [**] 139.130.61.206:109 -> MY.NET.4.11:109
11/28-20:03:02.262991 [**] SYN-FIN scan! [**] 139.130.61.206:109 -> MY.NET.4.22:109
11/28-20:03:02.676619 [**] SYN-FIN scan! [**] 139.130.61.206:109 -> MY.NET.4.43:109
11/28-20:03:03.365556 [**] SYN-FIN scan! [**] 139.130.61.206:109 -> MY.NET.4.76:109
11/28-20:03:03.427207 [**] SYN-FIN scan! [**] 139.130.61.206:109 -> MY.NET.4.80:109
11/28-20:03:03.603252 [**] SYN-FIN scan! [**] 139.130.61.206:109 -> MY.NET.4.89:109
11/28-20:03:04.129266 [**] SYN-FIN scan! [**] 139.130.61.206:109 -> MY.NET.4.115:109

Snort OOS Scans
```
01/10-12:05:59.008697 195.56.182.206:21 -> MY.NET.60.4:21
TCP TTL:28 TOS:0x0 ID:39426
**SF**** Seq: 0x73CA09A4   Ack: 0x2BFBF785   Win: 0x404
00 00 00 00 00 00                                ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/10-12:05:59.029151 195.56.182.206:21 -> MY.NET.60.5:21
TCP TTL:28 TOS:0x0 ID:39426
**SF**** Seq: 0x73CA09A4   Ack: 0x2BFBF785   Win: 0x404
00 00 00 00 00 00                                ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/10-12:05:59.047457 195.56.182.206:21 -> MY.NET.60.6:21
TCP TTL:28 TOS:0x0 ID:39426
```

```
**SF**** Seq: 0x73CA09A4   Ack: 0x2BFBF785   Win: 0x404
00 00 00 00 00 00                               ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/10-12:05:59.068691 195.56.182.206:21 -> MY.NET.60.7:21
TCP TTL:28 TOS:0x0 ID:39426
**SF**** Seq: 0x73CA09A4   Ack: 0x2BFBF785   Win: 0x404
00 00 00 00 00 00                               ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/10-12:05:59.081619 195.56.182.206:21 -> MY.NET.60.8:21
TCP TTL:28 TOS:0x0 ID:39426
**SF**** Seq: 0x73CA09A4   Ack: 0x2BFBF785   Win: 0x404
00 00 00 00 00 00                               ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/10-12:05:59.107369 195.56.182.206:21 -> MY.NET.60.9:21
TCP TTL:28 TOS:0x0 ID:39426
**SF**** Seq: 0x73CA09A4   Ack: 0x2BFBF785   Win: 0x404
00 00 00 00 00 00                               ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/10-12:05:59.125941 195.56.182.206:21 -> MY.NET.60.10:21
TCP TTL:28 TOS:0x0 ID:39426
**SF**** Seq: 0x73CA09A4   Ack: 0x2BFBF785   Win: 0x404
00 00 00 00 00 00                               ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/10-12:05:59.300238 195.56.182.206:21 -> MY.NET.60.19:21
TCP TTL:28 TOS:0x0 ID:39426
**SF**** Seq: 0x73CA09A4   Ack: 0x2BFBF785   Win: 0x404
00 00 00 00 00 00                               ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/10-12:05:59.328588 195.56.182.206:21 -> MY.NET.60.20:21
TCP TTL:28 TOS:0x0 ID:39426
**SF**** Seq: 0x73CA09A4   Ack: 0x2BFBF785   Win: 0x404
00 00 00 00 00 00                               ......

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

Syn-Fin scans are typically scans for types of operating systems. Different operating systems respond differently to the Syn-Fin. It can also be used to tell which machines have listeners on the ports that the Syn-Fin packet was addressed to. Syn-Fins can also be used to get around intrusion detections, which may only be looking for valid flags. In the snort scans, I saw SynFin probes from a xx sites. They were directed to a variety of ports; with the heaviest hit ports being port 109, 53, 21 and 9055. The heaviest hitters are listed:
630 SYN-FIN packets were detected.from address 132.68.37.141 on port 21
1242 SYN-FIN packets were detected.rom address 63.204.152.253 on port 53
3052 SYN-FIN packets were detected.rom address 194.204.224.131 on port 109
17604 SYN-FIN packets were detected.om address 211.34.40.1 on port 53
1951 SYN-FIN packets were detected.rom address 139.130.61.206 on port 109

8565 SYN-FIN packets were detected.rom address 194.234.48.26 on port 21
1580 SYN-FIN packets were detected.rom address 194.197.170.7 on port 9055
706 SYN-FIN packets were detected.from address 193.253.202.9 on port 21
4096 SYN-FIN packets were detected.rom address 147.8.182.157 on port 109
9878 SYN-FIN packets were detected.rom address 195.56.182.206 on port 21
1790 SYN-FIN packets were detected from address 200.194.102.99 on port 21

Interestingly enough, six of the above addresses (194.204.224.131, 211.34.40.1, 194.234.48.26. 147.8.182.157, 195.56.182.206 and 200.194.102.99 timeout when issuing nslookup for them. As for the OOS scan, it showed nothing was unusual except for the Syn-Fin flags.

TCP SMTP Source Port traffic

Not having access to the snort rule that generated this alert, I can only guess what the admin was trying to catch. First, there were 100 such incidents, where the source port is 25 and the destination port is 25. This would be normal for intercommunications between sendmail servers.

12/29-19:44:34.554608 [**] TCP SMTP Source Port traffic [**] 63.11.25.117:25 -> MY.NET.10.55:25
12/29-19:44:34.555339 [**] TCP SMTP Source Port traffic [**] 63.11.25.117:25 -> MY.NET.154.25:25
12/29-19:44:34.897578 [**] TCP SMTP Source Port traffic [**] 63.11.25.117:25 -> MY.NET.151.66:25
12/29-19:44:34.901062 [**] TCP SMTP Source Port traffic [**] 63.11.25.117:25 -> MY.NET.68.33:25
12/29-19:44:35.107165 [**] TCP SMTP Source Port traffic [**] 63.11.25.117:25 -> MY.NET.105.59:25
12/29-19:44:35.108907 [**] TCP SMTP Source Port traffic [**] 63.11.25.117:25 -> MY.NET.71.34:25
12/29-19:44:35.137350 [**] TCP SMTP Source Port traffic [**] 63.11.25.117:25 -> MY.NET.10.101:25
12/29-19:44:38.233079 [**] TCP SMTP Source Port traffic [**] 63.11.25.117:25 -> MY.NET.68.29:25
12/29-19:44:38.245935 [**] TCP SMTP Source Port traffic [**] 63.11.25.117:25 -> MY.NET.2.206:25
12/29-19:44:38.248866 [**] TCP SMTP Source Port traffic [**] 63.11.25.117:25 -> MY.NET.162.43:25
12/29-19:44:38.363036 [**] TCP SMTP Source Port traffic [**] 63.11.25.117:25 -> MY.NET.141.67:25
12/29-19:44:38.403936 [**] TCP SMTP Source Port traffic [**] 63.11.25.117:25 -> MY.NET.143.147:25
12/29-19:44:38.404798 [**] TCP SMTP Source Port traffic [**] 63.11.25.117:25 -> MY.NET.2.103:25

12/29-19:44:38.404933 [**] TCP SMTP Source Port traffic [**] 63.11.25.117:25 -> MY.NET.60.26:25

As can be seen, the vast majority of the hits originated from 63.11.25.117 to specific servers in my.net, maybe a scan for other programs listening on port 25. This could have been reconnaissance for a future attack after locations of mail servers were found.

Happy 99 Virus

A single alert for the Happy 99 virus was noticed by snort:
12/22-20:25:10.840208 [**] Happy 99 Virus [**] 63.216.198.158:2239 -> MY.NET.6.47:25
This sounds like a hit on a virus by an alert specifically looking for that signature. The mail server should be examined for the virus and removed as necessary.

According to Herschel Gelman in his SANS DC 2000 Practical Assignment, this virus contains the character string "X-Spanska:Yes"). The additional snort scans provided for analysis does not reveal this character string so I cannot positively identify it as the Happy 99. However, precautions should be exercised just in case.

Tiny Fragments - Possible Hostile Activity

Tiny fragments are a penetration technique by splitting the header into multiple parts (SANS New Orleans, Jan 2001, Intrusion Detection Track 3). There were several tiny fragments packets observed by snort:
01/08-19:31:40.978665 [**] Tiny Fragments - Possible Hostile Activity [**] 202.101.43.22 -> MY.NET.1.8
01/08-19:31:40.978762 [**] Tiny Fragments - Possible Hostile Activity [**] 202.101.43.22 -> MY.NET.1.8
01/08-20:20:14.649930 [**] Tiny Fragments - Possible Hostile Activity [**] 202.108.43.51 -> MY.NET.1.8
01/08-20:25:29.556073 [**] Tiny Fragments - Possible Hostile Activity [**] 202.96.96.3-> MY.NET.1.8
01/08-20:25:29.556164 [**] Tiny Fragments - Possible Hostile Activity [**] 202.96.96.3-> MY.NET.1.8
01/08-20:44:55.829858 [**] Tiny Fragments - Possible Hostile Activity [**] 61.140.75.5-> MY.NET.1.8
01/08-20:51:18.240574 [**] Tiny Fragments - Possible Hostile Activity [**] 61.140.75.5-> MY.NET.1.10
01/08-20:51:18.240666 [**] Tiny Fragments - Possible Hostile Activity [**] 61.140.75.5-> MY.NET.1.10
01/08-21:02:22.142298 [**] Tiny Fragments - Possible Hostile Activity [**] 61.134.9.13 -> MY.NET.1.8
01/08-21:06:10.168953 [**] Tiny Fragments - Possible Hostile Activity [**] 61.140.75.5-> MY.NET.1.8

01/08-21:29:22.216641 [**] Tiny Fragments - Possible Hostile Activity [**] 210.12.160.30 -> MY.NET.1.8
01/08-21:42:59.944866 [**] Tiny Fragments - Possible Hostile Activity [**] 61.134.9.13 -> MY.NET.1.8
01/08-21:44:35.714323 [**] Tiny Fragments - Possible Hostile Activity [**] 210.12.160.30 -> MY.NET.1.8
/08-21:44:35.714412 [**] Tiny Fragments - Possible Hostile Activity [**] 210.12.160.30 -> MY.NET.1.8
01/08-21:48:25.081338 [**] Tiny Fragments - Possible Hostile Activity [**] 61.155.13.3-> MY.NET.1.8
01/08-21:48:25.081438 [**] Tiny Fragments - Possible Hostile Activity [**] 61.155.13.3-> MY.NET.1.8
01/08-21:51:26.878989 [**] Tiny Fragments - Possible Hostile Activity [**] 61.134.9.13 -> MY.NET.1.8
01/08-22:01:50.196346 [**] Tiny Fragments - Possible Hostile Activity [**] 210.12.160.30 -> MY.NET.1.8
01/08-22:01:50.196493 [**] Tiny Fragments - Possible Hostile Activity [**] 210.12.160.30 -> MY.NET.1.8
01/08-22:11:53.711091 [**] Tiny Fragments - Possible Hostile Activity [**] 61.134.9.13 -> MY.NET.1.8
01/08-22:26:20.868690 [**] Tiny Fragments - Possible Hostile Activity [**] 202.205.5.1 -> MY.NET.1.8
01/08-22:29:28.574636 [**] Tiny Fragments - Possible Hostile Activity [**] 61.155.13.3-> MY.NET.1.8
01/08-22:29:28.574726 [**] Tiny Fragments - Possible Hostile Activity [**] 61.155.13.3-> MY.NET.1.8
01/08-22:56:07.569098 [**] Tiny Fragments - Possible Hostile Activity [**] 202.108.43.52 -> MY.NET.1.8

Tthe vast majority were directed to my.net.1.8, my.net.1.9 or my.net.1.10. While these three address were involved with other exchanges, there is nothing to suggest that they were compromised in any way.

Back Orifice

Back Orifice is remote takeover program whereby a server is loaded onto a compromised computer and clients on the attacker's local machine can communicate with the host to launch attacks. In our snort scans, I found 77 packets that could be associated with this exploit. Here are a few found in the snort alerts files:

12/09-22:23:21.911339 [**] Back Orifice [**] 209.94.199.202:31338 -> MY.NET.60.174:31337
12/09-22:23:22.210320 [**] Back Orifice [**] 209.94.199.202:31338 -> MY.NET.60.196:31337
12/09-22:23:22.233913 [**] Back Orifice [**] 209.94.199.202:31338 -> MY.NET.60.197:31337

12/09-22:23:22.248025 [**] Back Orifice [**] 209.94.199.202:31338 -> MY.NET.60.200:31337
12/09-22:23:22.254885 [**] Back Orifice [**] 209.94.199.202:31338 -> MY.NET.60.201:31337
12/09-22:23:22.439725 [**] Back Orifice [**] 209.94.199.202:31338 -> MY.NET.60.221:31337
12/09-22:23:22.500230 [**] Back Orifice [**] 209.94.199.202:31338 -> MY.NET.60.228:31337
12/09-22:23:22.668469 [**] Back Orifice [**] 209.94.199.202:31338 -> MY.NET.60.235:31337
12/09-22:23:22.691936 [**] Back Orifice [**] 209.94.199.202:31338 -> MY.NET.60.239:31337
12/09-22:23:22.692095 [**] Back Orifice [**] 209.94.199.202:31338 -> MY.NET.60.243:31337
12/09-22:23:22.700555 [**] Back Orifice [**] 209.94.199.202:31338 -> MY.NET.60.247:31337
12/09-22:23:22.707559 [**] Back Orifice [**] 209.94.199.202:31338 -> MY.NET.60.248:31337
12/09-22:25:07.668148 [**] Back Orifice [**] 209.94.199.202:31338 -> MY.NET.60.2:31337
12/09-22:25:07.839921 [**] Back Orifice [**] 209.94.199.202:31338 -> MY.NET.60.15:31337
12/09-22:25:07.839978 [**] Back Orifice [**] 209.94.199.202:31338 -> MY.NET.60.16:31337
12/09-22:25:07.917873 [**] Back Orifice [**] 209.94.199.202:31338 -> MY.NET.60.22:31337
12/09-22:25:08.039128 [**] Back Orifice [**] 209.94.199.202:31338 -> MY.NET.60.36:31337
12/09-22:25:08.379764 [**] Back Orifice [**] 209.94.199.202:31338 -> MY.NET.60.152:31337
12/09-22:25:08.829984 [**] Back Orifice [**] 209.94.199.202:31338 -> MY.NET.60.185:31337
12/09-22:25:09.119468 [**] Back Orifice [**] 209.94.199.202:31338 -> MY.NET.60.212:31337
12/09-22:25:09.130663 [**] Back Orifice [**] 209.94.199.202:31338 -> MY.NET.60.216:31337
12/09-22:25:09.218353 [**] Back Orifice [**] 209.94.199.202:31338 -> MY.NET.60.224:31337
12/26-11:56:14.718987 [**] Back Orifice [**] 212.217.124.157:4484 -> MY.NET.97.242:31337
12/30-21:53:41.829492 [**] Back Orifice [**] 216.99.200.242:50012 -> MY.NET.202.94:31337
12/31-00:27:58.511333 [**] Back Orifice [**] 216.99.200.242:50012 -> MY.NET.202.94:31337
12/31-00:28:02.547558 [**] Back Orifice [**] 216.99.200.242:50013 -> MY.NET.202.94:31337

01/05-21:51:04.702446 [**] Back Orifice [**] 24.112.86.56:3166 -> MY.NET.98.115:31337
01/12-06:55:36.674631 [**] Back Orifice [**] 207.253.109.40:3574 -> MY.NET.60.8:31337
01/12-07:09:51.971904 [**] Back Orifice [**] 207.253.109.40:2497 -> MY.NET.60.8:31337
01/16-16:16:00.885539 [**] Back Orifice [**] 64.229.42.221:1144 -> MY.NET.217.150:31337
Of the list, the most noted one is source address 209.94.199.202 (cuscon1096.tstt.net.tt) possible probing for a BO listener on the my.net.60 subnet.

Other scans indicated that there was a lot of interest in my.net.60 subnet from a variety of origins, including a Syn-Fin probe from 195.56.182.206.

Watchlist 000222 NET-NCFC

```
01/11-03:33:24.434649  [**] Watchlist 000222 NET-NCFC [**]
159.226.45.3:3331 -> MY.NET.253.42:25
01/11-03:33:36.229691  [**] Watchlist 000222 NET-NCFC [**]
159.226.45.3:3331 -> MY.NET.253.42:25
01/11-03:33:37.150143  [**] Watchlist 000222 NET-NCFC [**]
159.226.45.3:3331 -> MY.NET.253.42:25
01/11-03:33:41.182980  [**] Watchlist 000222 NET-NCFC [**]
159.226.45.3:3331 -> MY.NET.253.42:25
01/11-03:33:49.680633  [**] Watchlist 000222 NET-NCFC [**]
159.226.45.3:3331 -> MY.NET.253.42:25
01/11-03:33:57.208186  [**] Watchlist 000222 NET-NCFC [**]
159.226.45.3:3335 -> MY.NET.6.7:113
01/11-08:59:39.649577  [**] Watchlist 000222 NET-NCFC [**]
159.226.45.3:4555 -> MY.NET.253.42:25
01/11-08:59:40.128817  [**] Watchlist 000222 NET-NCFC [**]
159.226.45.3:4555 -> MY.NET.253.42:25
01/11-09:00:35.981375 [**] Watchlist 000222 NET-NCFC [**]
159.226.45.3:113 -> MY.NET.253.42:55072
01/11-09:00:40.468897  [**] Watchlist 000222 NET-NCFC [**]
159.226.45.3:4555 -> MY.NET.253.42:25
01/11-09:00:41.117514  [**] Watchlist 000222 NET-NCFC [**]
159.226.45.3:4555 -> MY.NET.253.42:25
01/11-09:00:54.801043  [**] Watchlist 000222 NET-NCFC [**]
159.226.45.3:4555 -> MY.NET.253.42:25
01/11-09:00:54.802672  [**] Watchlist 000222 NET-NCFC [**]
159.226.45.3:4555 -> MY.NET.253.42:25
01/11-09:00:57.552250  [**] Watchlist 000222 NET-NCFC [**]
159.226.45.3:4555 -> MY.NET.253.42:25
01/11-09:01:27.295281  [**] Watchlist 000222 NET-NCFC [**]
159.226.45.3:4555 -> MY.NET.253.42:25

and

01/12-02:14:40.128308  [**] Watchlist 000222 NET-NCFC [**]
159.226.121.37:1246 -> MY.NET.6.7:143
```

```
01/12-02:14:40.459726  [**] Watchlist 000222 NET-NCFC [**]
159.226.121.37:1246 -> MY.NET.6.7:143
01/12-02:14:41.312220  [**] Watchlist 000222 NET-NCFC [**]
159.226.121.37:1246 -> MY.NET.6.7:143
01/12-02:14:57.909557  [**] Watchlist 000222 NET-NCFC [**]
159.226.121.37:1245 -> MY.NET.6.7:143
01/12-02:14:57.911021  [**] Watchlist 000222 NET-NCFC [**]
159.226.121.37:1245 -> MY.NET.6.7:143
01/12-02:14:58.873655  [**] Watchlist 000222 NET-NCFC [**]
159.226.121.37:1246 -> MY.NET.6.7:143
01/12-02:14:58.888362  [**] Watchlist 000222 NET-NCFC [**]
159.226.121.37:1246 -> MY.NET.6.7:143
```

Although, there were 2400 packets from this particular network, most were directed at the sendmail server (port 250 and the IMAP server (port 143). I do not know why they were placed on the watchlist, I do not see any vicious activity from these scans.

Watchlist 000220 IL-ISDNNET-990517

```
01/16-08:22:53.054820  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.45.241:2728 -> MY.NET.202.94:7000
01/16-08:22:54.108004  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.45.241:2728 -> MY.NET.202.94:7000
01/16-08:22:55.995194  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.45.241:2728 -> MY.NET.202.94:7000
01/16-08:22:56.944266  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.45.241:2728 -> MY.NET.202.94:7000
01/16-08:23:04.028035  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.45.241:2728 -> MY.NET.202.94:7000
01/16-08:23:10.750681  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.45.241:2728 -> MY.NET.202.94:7000
01/16-08:23:11.652853  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.45.241:2728 -> MY.NET.202.94:7000
01/16-08:23:15.582188  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.45.241:2728 -> MY.NET.202.94:7000
01/16-08:23:20.914416  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.45.241:2728 -> MY.NET.202.94:7000
01/16-08:23:21.857269  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.45.241:2728 -> MY.NET.202.94:7000
01/16-08:23:30.213284  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.45.241:2728 -> MY.NET.202.94:7000
01/16-08:23:31.366787  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.45.241:2728 -> MY.NET.202.94:7000
01/16-08:23:31.675471  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.45.241:2728 -> MY.NET.202.94:7000
01/16-08:23:31.883821  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.45.241:2728 -> MY.NET.202.94:7000
01/16-08:23:33.033259  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.45.241:2728 -> MY.NET.202.94:7000
01/16-08:23:33.150003  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.45.241:2728 -> MY.NET.202.94:7000
01/16-08:23:33.636859  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.45.241:2728 -> MY.NET.202.94:7000
```

```
01/16-08:23:33.872960  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.45.241:2728 -> MY.NET.202.94:7000
```

This network is already on the watchlist and it probably needs to be. There were 221176 packets from this network, mostly hitting ports 6699 and 7000. Port 6699 could be used by Napster. Port 7000 could be used by Subseven. Without additional information, it would be hard to say what was going on.

One thing observed is that some sites were more interested in my.net than others. Further analysis of the alerts found the following:
Watchlist 000220 IL-ISDNNET-990517 activity reported from 212.179.77.20 2288 time(s).
Watchlist 000220 IL-ISDNNET-990517 activity reported from 212.179.79.2 25181 time(s).
Watchlist 000220 IL-ISDNNET-990517 activity reported from 212.179.44.105 1517 time(s).
Watchlist 000220 IL-ISDNNET-990517 activity reported from 212.179.27.111 1062 time(s).
Watchlist 000220 IL-ISDNNET-990517 activity reported from 212.179.79.2 9309 time(s).
Watchlist 000220 IL-ISDNNET-990517 activity reported from 212.179.38.135 1221 time(s).
Watchlist 000220 IL-ISDNNET-990517 activity reported from 212.179.27.111 37604 time(s).
Watchlist 000220 IL-ISDNNET-990517 activity reported from 212.179.79.2 9525 time(s).
Watchlist 000220 IL-ISDNNET-990517 activity reported from 212.179.42.102 1014 time(s).

Since all of the top 10 were from the 212.179 net, a quick nslookup on 212.179.1.1 resolved the name to fr-c27001.arel.co.it, indicating they originated from an Israeli network. (Another look indicated they all came from 212.179). And that there was two way conversation:
Watchlist 000220 IL-ISDNNET-990517 activity directed to MY.NET.221.10 by 212.179.79.2 386 time(s).
Watchlist 000220 IL-ISDNNET-990517 activity directed to MY.NET.217.246 by 212.179.79.2 1 time(s).
Watchlist 000220 IL-ISDNNET-990517 activity directed to MY.NET.220.126 by 212.179.79.2 25182 time(s).
Watchlist 000220 IL-ISDNNET-990517 activity directed to MY.NET.202.234 by 212.179.79.2 8 time(s).
Watchlist 000220 IL-ISDNNET-990517 activity directed to MY.NET.221.158 by 212.179.79.2 1 time(s).
Watchlist 000220 IL-ISDNNET-990517 activity directed to MY.NET.201.142 by 212.179.45.241 6 time(s).
Watchlist 000220 IL-ISDNNET-990517 activity directed to MY.NET.202.30 by 212.179.77.20 2288 time(s).

Watchlist 000220 IL-ISDNNET-990517 activity directed to MY.NET.204.166 by 212.179.44.119 24 time(s).
Watchlist 000220 IL-ISDNNET-990517 activity directed to MY.NET.219.18 by 212.179.79.2 8 time(s).
Watchlist 000220 IL-ISDNNET-990517 activity directed to MY.NET.218.22 by 212.179.37.92 3 time(s).
Watchlist 000220 IL-ISDNNET-990517 activity directed to MY.NET.253.43 by 212.179.7.36 32 time(s).
Watchlist 000220 IL-ISDNNET-990517 activity directed to MY.NET.229.114 by 212.179.79.2 5080 time(s).
Watchlist 000220 IL-ISDNNET-990517 activity directed to MY.NET.202.234 by 212.179.7.161 1 time(s).
More data should be collected to find out what the interest is.

External RPC call
On Solaris 2.x operating systems, rpcbind listens on TCP/UDP port 111. Rpcbind permits a remote attacker to insert and delete entries without "super user" status by spoofing a source address. Ironically, it inserts the entries as being owned by "super user." If any of the IP address listed below were able to connect to port 111, the systems may have been compromised (per Andrew G. Siske Jr. from his GIAC Intrusion Detection Practical Assignment for SANS Security DC 2000 July 5 – 10, 2000)

External RPC call

```
2/20-15:05:30.479083  [**] External RPC call [**]
148.228.125.215:1754 -> MY.NET.133.16:111
2/20-15:05:30.492407  [**] External RPC call [**]
148.228.125.215:1826 -> MY.NET.133.87:111
2/20-15:05:33.432083  [**] External RPC call [**]
148.228.125.215:1995 -> MY.NET.133.252:111
2/20-15:05:33.433899  [**] External RPC call [**]
148.228.125.215:1997 -> MY.NET.133.254:111
2/20-15:05:33.434043  [**] External RPC call [**]
148.228.125.215:1740 -> MY.NET.133.2:111
2/20-15:05:33.434096  [**] External RPC call [**]
148.228.125.215:1742 -> MY.NET.133.4:111
2/20-15:05:33.478589  [**] External RPC call [**]
148.228.125.215:1942 -> MY.NET.133.199:111
2/20-15:05:33.478686  [**] External RPC call [**]
148.228.125.215:1813 -> MY.NET.133.74:111
2/20-15:05:33.478738  [**] External RPC call [**]
148.228.125.215:1814 -> MY.NET.133.75:111
2/20-15:05:33.485015  [**] External RPC call [**]
148.228.125.215:1842 -> MY.NET.133.103:111
```

```
2/20-15:05:33.485064  [**] External RPC call [**]
148.228.125.215:1843 -> MY.NET.133.104:111
2/20-15:05:33.485983  [**] External RPC call [**]
148.228.125.215:1850 -> MY.NET.133.111:111
2/20-15:05:33.495392  [**] External RPC call [**]
148.228.125.215:1884 -> MY.NET.133.141:111
2/22-09:33:22.421500  [**] External RPC call [**]
195.57.62.153:2567 -> MY.NET.15.127:111
2/24-23:09:31.264010  [**] External RPC call [**]
208.185.235.100:1605 -> MY.NET.6.15:111
2/24-23:09:31.264509  [**] External RPC call [**]
208.185.235.100:1605 -> MY.NET.6.15:111
2/24-23:09:31.439030  [**] External RPC call [**]
208.185.235.100:1605 -> MY.NET.6.15:111
2/24-23:29:40.993129  [**] External RPC call [**]
208.185.235.100:4065 -> MY.NET.94.75:111
2/24-23:30:55.515786  [**] External RPC call [**]
208.185.235.100:4213 -> MY.NET.100.130:111
2/29-19:44:58.915910  [**] External RPC call [**]
63.11.25.117:1661 -> MY.NET.6.15:111
2/29-19:44:59.267296  [**] External RPC call [**]
63.11.25.117:1661 -> MY.NET.6.15:111
2/29-19:44:59.283486  [**] External RPC call [**]
63.11.25.117:1661 -> MY.NET.6.15:111
2/29-19:44:59.574997  [**] External RPC call [**]
63.11.25.117:2 -> MY.NET.6.15:111
2/29-19:44:59.672590  [**] External RPC call [**]
63.11.25.117:5 -> MY.NET.6.15:111
2/29-19:44:59.914419  [**] External RPC call [**]
63.11.25.117:4 -> MY.NET.6.15:111
2/29-19:45:05.937334  [**] External RPC call [**]
63.11.25.117:1009 -> MY.NET.6.15:111
2/30-14:26:56.780877  [**] External RPC call [**]
130.212.20.72:3810 -> MY.NET.6.15:111
2/30-14:26:56.917902  [**] External RPC call [**]
130.212.20.72:969 -> MY.NET.6.15:111
2/30-14:26:57.014288  [**] External RPC call [**]
130.212.20.72:969 -> MY.NET.6.15:111
2/30-14:26:57.014350  [**] External RPC call [**]
130.212.20.72:969 -> MY.NET.6.15:111
2/30-14:28:00.689070  [**] External RPC call [**]
130.212.20.72:2254 -> MY.NET.15.127:111
1/01-11:00:03.077635  [**] External RPC call [**]
211.48.210.193:1251 -> MY.NET.15.127:111
1/02-15:00:55.007003  [**] External RPC call [**]
192.71.148.152:4847 -> MY.NET.15.127:111
```

```
1/06-05:04:21.793408  [**] External RPC call [**]
206.210.80.6:1414 -> MY.NET.6.15:111
1/06-05:04:21.829933  [**] External RPC call [**]
206.210.80.6:1414 -> MY.NET.6.15:111
1/06-05:04:21.830004  [**] External RPC call [**]
206.210.80.6:1414 -> MY.NET.6.15:111
1/06-05:04:21.888825  [**] External RPC call [**]
206.210.80.6:1414 -> MY.NET.6.15:111
1/06-05:04:21.888876  [**] External RPC call [**]
206.210.80.6:1414 -> MY.NET.6.15:111
1/06-05:04:21.919235  [**] External RPC call [**]
206.210.80.6:1414 -> MY.NET.6.15:111
1/06-05:04:45.761356  [**] External RPC call [**]
206.210.80.6:3832 -> MY.NET.15.127:111
1/06-05:08:19.304357  [**] External RPC call [**]
206.210.80.6:1751 -> MY.NET.100.130:111
1/18-20:12:23.068148  [**] External RPC call [**]
202.84.134.141:748 -> MY.NET.6.15:111
1/18-20:12:23.672941  [**] External RPC call [**]
202.84.134.141:748 -> MY.NET.6.15:111
1/18-20:12:46.806033  [**] External RPC call [**]
202.84.134.141:615 -> MY.NET.15.127:111
1/18-20:16:20.752084  [**] External RPC call [**]
202.84.134.141:718 -> MY.NET.100.130:111
```

Some of these packets appear to be scanning the my.net.133 subnet for a listener. However, there were many pointed to my.net.6.15. This could be a compromised server and deserves to be looked at.

Attempted Sun RPC high port access

```
01/18-15:22:22.115054  [**] Attempted Sun RPC high port access [**]
205.188.153.102:4000 -> MY.NET.105.115:32771
01/18-15:28:21.871061  [**] Attempted Sun RPC high port access [**]
205.188.153.102:4000 -> MY.NET.105.115:32771
01/18-15:29:22.051633  [**] Attempted Sun RPC high port access [**]
205.188.153.102:4000 -> MY.NET.105.115:32771
01/18-15:32:21.905052  [**] Attempted Sun RPC high port access [**]
205.188.153.102:4000 -> MY.NET.105.115:32771
01/18-15:34:22.019894  [**] Attempted Sun RPC high port access [**]
205.188.153.102:4000 -> MY.NET.105.115:32771
01/18-15:36:21.922805  [**] Attempted Sun RPC high port access [**]
205.188.153.102:4000 -> MY.NET.105.115:32771
01/18-15:46:21.778928  [**] Attempted Sun RPC high port access [**]
205.188.153.102:4000 -> MY.NET.105.115:32771
01/18-15:47:21.729610  [**] Attempted Sun RPC high port access [**]
205.188.153.102:4000 -> MY.NET.105.115:32771
```

01/18-15:50:21.698215 [**] Attempted Sun RPC high port access [**]
205.188.153.102:4000 -> MY.NET.105.115:32771
01/18-15:57:21.298708 [**] Attempted Sun RPC high port access [**]
205.188.153.102:4000 -> MY.NET.105.115:32771
01/18-15:57:36.545067 [**] Attempted Sun RPC high port access [**]
205.188.153.102:4000 -> MY.NET.105.115:32771
01/18-16:01:04.557428 [**] Attempted Sun RPC high port access [**]
205.188.153.102:4000 -> MY.NET.105.115:32771
01/18-16:02:31.145278 [**] Attempted Sun RPC high port access [**]
205.188.153.102:4000 -> MY.NET.105.115:32771
01/18-16:02:31.145352 [**] Attempted Sun RPC high port access [**]
205.188.153.102:4000 -> MY.NET.105.115:32771
01/18-16:04:20.894643 [**] Attempted Sun RPC high port access [**]
205.188.153.102:4000 -> MY.NET.105.115:32771
01/18-16:07:20.694301 [**] Attempted Sun RPC high port access [**]
205.188.153.102:4000 -> MY.NET.105.115:32771
01/18-16:13:20.629641 [**] Attempted Sun RPC high port access [**]
205.188.153.102:4000 -> MY.NET.105.115:32771
01/18-16:14:20.851226 [**] Attempted Sun RPC high port access [**]
205.188.153.102:4000 -> MY.NET.105.115:32771
01/18-16:18:21.395469 [**] Attempted Sun RPC high port access [**]
205.188.153.102:4000 -> MY.NET.105.115:32771
01/18-16:26:21.299216 [**] Attempted Sun RPC high port access [**]
205.188.153.102:4000 -> MY.NET.105.115:32771
01/18-16:26:36.828690 [**] Attempted Sun RPC high port access [**]
205.188.153.102:4000 -> MY.NET.105.115:32771

Two thousand hits, all from the 205.188 network, all with source port 4000 and destination port 32771. Name resolution of 205.188.153.102 was not found but 205.188.153.100 was resolved to fes-d004.icq.aol.com. This is probably an ICQ connection with AOL. Only 2000 in a two-month span probably is not excessive. However, there were a few other addresses seen in the alerts: 216.99.200.242, 216.34.243.246 and 216.13.244.241. None of these addresses resolved to names:

Attempted Sun RPC high port access activity from 216.34.243.246 to MY.NET.104.52 1 time(s).
Attempted Sun RPC high port access activity from 216.13.244.241 to MY.NET.221.130 45 time(s).
Attempted Sun RPC high port access activity from 216.99.200.242 to MY.NET.202.94 4 time(s).

In the noise of the ICQ traffic, the above my.net. machines could have been compromised.

NMAP TCP ping!/ Probable NMAP fingerprint attempt

As found in Steven Northcutt's Network Intrusion Detection – An Analyst's Handbook, "nmpa is the most versatile scanner available. This software can create a large number of

traces, and in early 1999 was being called the most potent denial-of-service engine available." Its purpose is "to quickly determine what services the internal system has available." And it can even be used for OS fingerprinting.

```
1/06-00:09:52.682882  [**] NMAP TCP ping! [**] 204.155.48.3:80 ->
MY.NET.179.77:80^M
1/06-22:39:02.441347  [**] NMAP TCP ping! [**] 192.102.197.234:80 ->
MY.NET.1.10:53^M
1/06-22:39:02.457644  [**] NMAP TCP ping! [**] 192.102.197.234:53 ->
MY.NET.1.10:53^M
1/07-10:48:33.629114  [**] NMAP TCP ping! [**] 192.102.197.234:80 ->
MY.NET.1.8:53^M
1/07-10:48:33.633724  [**] NMAP TCP ping! [**] 192.102.197.234:53 ->
MY.NET.1.8:53^M
1/07-10:48:38.631203  [**] NMAP TCP ping! [**] 192.102.197.234:80 ->
MY.NET.1.8:53^M
1/07-10:48:38.635631  [**] NMAP TCP ping! [**] 192.102.197.234:53 ->
MY.NET.1.8:53^M
1/07-11:57:25.475707  [**] NMAP TCP ping! [**] 199.197.130.21:80 ->
MY.NET.253.114:80^M
1/07-16:48:02.083059  [**] NMAP TCP ping! [**] 194.186.36.190:53 ->
MY.NET.1.8:53^M
1/07-16:48:02.088099  [**] NMAP TCP ping! [**] 194.186.36.190:80 ->
MY.NET.1.8:53^M
1/08-03:55:04.080517  [**] NMAP TCP ping! [**] 194.133.58.129:80 ->
MY.NET.1.4:53^M
1/08-03:55:06.576846  [**] NMAP TCP ping! [**] 194.133.58.129:80 ->
MY.NET.100.165:80^M
1/08-03:55:06.699552  [**] NMAP TCP ping! [**] 212.208.74.129:80 ->
MY.NET.100.165:80^M
1/08-09:18:26.175613  [**] NMAP TCP ping! [**] 12.21.190.9:80 ->
MY.NET.60.14:80^M
1/08-09:18:26.251514  [**] NMAP TCP ping! [**] 208.205.199.9:80 ->
MY.NET.60.14:80^M
```

```
01/18-14:27:56.251483  [**] NMAP TCP ping! [**] MY.NET.70.38:52342 ->
MY.NET.0.0:37558^M
01/18-14:28:04.964171  [**] NMAP TCP ping! [**] MY.NET.70.38:52342 ->
MY.NET.0.0:40997^M
01/18-14:28:38.544677  [**] NMAP TCP ping! [**] MY.NET.70.38:52342 ->
MY.NET.0.1:41638^M
01/18-14:29:03.573081  [**] NMAP TCP ping! [**] MY.NET.70.38:52342 ->
MY.NET.0.2:41014^M
```

01/18-14:29:12.724763 [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.2:32979^M
01/18-14:29:27.512340 [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.3:38829^M
01/18-14:29:59.222497 [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.3:33778^M
01/18-14:32:34.871510 [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.5:37656^M
01/18-14:32:48.310398 [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.5:38614^M
01/18-14:33:23.310908 [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.5:38209^M
01/18-14:34:13.010505 [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.6:33821^M
01/18-14:34:23.399289 [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.6:33821^M
01/18-14:36:35.629674 [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.8:36703^M
01/18-14:40:05.678410 [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.10:41932^M
01/18-14:41:13.577136 [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.11:42943^M
01/18-14:42:07.417302 [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.12:40679^M
01/18-14:42:23.717414 [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.12:43028^M
01/18-14:43:06.206568 [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.12:44552^M
01/18-14:46:01.775501 [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.13:33220^M
01/18-14:47:26.696215 [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.14:44494^M
01/18-14:49:22.584183 [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.15:42300^M
01/18-14:51:22.543037 [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.16:38538^M
01/18-14:53:24.732660 [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.17:38630^M
01/18-14:54:15.866369 [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.18:37725^M
01/18-14:56:25.259471 [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.19:37676^M
01/18-14:56:39.982836 [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.19:37676^M
01/18-14:56:54.771575 [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.19:38195^M

01/18-15:00:05.720170 [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.20:35107^M
01/18-15:01:09.759717 [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.21:36527^M
01/18-15:02:14.198907 [**] NMAP TCP ping! [**] MY.NET.70.38:52342 -> MY.NET.0.21:36540^M

In the snort alerts, there were 558 hits on this rule. Interesting enough, my.net.70.38 originated 262 of them. This machine, using port 52342 appears to be interested in what is in my.net. This could be a sign that my.net.70.38 is already compromised. Another note, most of the remaining hits were single hits from many servers. Some of these are probably probes trying to see what they can find. But of note is 192.102.197.234 scanning my.net.1.8. my.net.1.8 has been seen in earlier scans. It was also the destination on a connection to port 515 alert. This may be another machine worth looking at. It, too, may be been compromised. Of note are the traps observed on December 8:

Dec 8 14:16:52 MY.NET.1.8:33503 -> 192.232.16.64:53 UDP
Dec 8 14:16:52 MY.NET.1.8:33503 -> 206.197.81.11:53 UDP
Dec 8 14:16:53 MY.NET.1.8:33503 -> 131.158.15.198:53 UDP
Dec 8 14:16:53 MY.NET.1.8:33503 -> 209.50.252.54:53 UDP
Dec 8 14:16:53 MY.NET.1.8:33503 -> 202.54.1.30:53 UDP
Dec 8 14:16:53 MY.NET.1.8:33503 -> 198.6.1.82:53 UDP
Dec 8 14:16:53 MY.NET.1.8:33503 -> 164.109.10.23:53 UDP
Dec 8 14:16:54 MY.NET.1.8:33503 -> 206.228.179.10:53 UDP
Dec 8 14:16:54 MY.NET.1.8:33503 -> 209.185.190.86:53 UDP
Dec 8 14:16:54 MY.NET.1.8:33503 -> 198.41.3.101:53 UDP
Dec 8 14:16:55 MY.NET.1.8:33503 -> 204.71.154.5:53 UDP
Dec 8 14:16:56 MY.NET.1.8:33503 -> 207.126.105.146:53 UDP
Dec 8 14:16:56 MY.NET.1.8:33503 -> 24.2.0.27:53 UDP
Dec 8 14:16:56 MY.NET.1.8:33503 -> 64.22.130.243:53 UDP
Dec 8 14:17:00 MY.NET.1.8:33503 -> 192.245.243.251:53 UDP
Dec 8 14:17:02 MY.NET.1.8:33503 -> 204.152.184.64:53 UDP
Dec 8 14:17:03 MY.NET.1.8:33503 -> 202.54.1.30:53 UDP
Dec 8 14:17:03 MY.NET.1.8:33503 -> 202.54.1.18:53 UDP
Dec 8 14:17:03 MY.NET.1.8:33503 -> 192.156.136.148:53 UDP
Dec 8 14:17:06 MY.NET.1.8:33503 -> 206.132.75.195:53 UDP
Dec 8 14:17:07 MY.NET.1.8:33503 -> 192.245.243.251:53 UDP
Dec 8 14:17:10 MY.NET.1.8:33503 -> 209.50.252.53:53 UDP
Dec 8 14:17:10 MY.NET.1.8:33503 -> 204.59.64.222:53 UDP
Dec 8 14:17:10 MY.NET.1.8:33503 -> 194.97.109.1:53 UDP
Dec 8 14:17:11 MY.NET.1.8:33503 -> 205.188.185.18:53 UDP
Dec 8 14:17:51 MY.NET.1.8:33503 -> 192.245.243.250:53 UDP
Dec 8 14:17:53 MY.NET.1.8:33503 -> 202.54.1.18:53 UDP
Dec 8 14:17:51 MY.NET.1.8:33503 -> 208.184.216.239:53 UDP
Dec 8 14:17:52 MY.NET.1.8:33503 -> 209.50.252.53:53 UDP
Dec 8 14:17:52 MY.NET.1.8:33503 -> 64.14.197.185:53 UDP
Dec 8 14:17:52 MY.NET.1.8:33503 -> 204.253.104.11:53 UDP

Dec 8 14:17:53 MY.NET.1.8:33503 -> 195.110.96.67:53 UDP
Dec 8 14:17:53 MY.NET.1.8:33503 -> 193.205.245.8:53 UDP
Dec 8 14:17:56 MY.NET.1.8:33503 -> 209.167.79.5:53 UDP
Dec 8 14:17:56 MY.NET.1.8:33503 -> 216.219.254.10:53 UDP
Dec 8 14:17:57 MY.NET.1.8:33503 -> 198.41.3.101:53 UDP
Dec 8 14:17:58 MY.NET.1.8:33503 -> 204.178.107.226:53 UDP
Dec 8 14:17:59 MY.NET.1.8:33503 -> 208.178.148.39:53 UDP
Dec 8 14:18:00 MY.NET.1.8:33503 -> 198.41.3.101:53 UDP
Dec 8 14:18:02 MY.NET.1.8:33503 -> 198.41.3.38:53 UDP
Dec 8 14:18:02 MY.NET.1.8:33503 -> 198.92.128.130:53 UDP
Dec 8 14:18:07 MY.NET.1.8:33503 -> 202.54.1.30:53 UDP
Dec 8 14:18:10 MY.NET.1.8:33503 -> 202.41.110.66:53 UDP
Dec 8 14:18:11 MY.NET.1.8:33503 -> 206.245.245.10:53 UDP
Dec 8 14:18:11 MY.NET.1.8:33503 -> 137.192.2.1:53 UDP
Dec 8 14:18:12 MY.NET.1.8:33503 -> 35.8.2.41:53 UDP
Dec 8 14:18:16 MY.NET.1.8:33503 -> 206.245.188.4:53 UDP
These are udp probes against DNS across many different networks.

SMB Name Wildcard

12/21-00:45:15.451549 [**] SMB Name Wildcard [**] 209.180.158.162:137 -> MY.NET.133.82:137^M
12/21-01:39:45.327927 [**] SMB Name Wildcard [**] 207.245.208.135:137 -> MY.NET.133.197:137^M
12/21-09:13:13.527372 [**] SMB Name Wildcard [**] 24.64.183.166:1064 -> MY.NET.133.253:137^M
12/21-09:13:16.528628 [**] SMB Name Wildcard [**] 24.64.183.166:1064 -> MY.NET.133.253:137^M
12/22-18:56:37.934882 [**] SMB Name Wildcard [**] MY.NET.101.160:137 -> MY.NET.101.192:137^M
12/22-18:56:39.424676 [**] SMB Name Wildcard [**] MY.NET.101.160:137 -> MY.NET.101.192:137^M
12/22-18:57:45.478510 [**] SMB Name Wildcard [**] MY.NET.101.160:137 -> MY.NET.101.192:137^M
12/22-18:57:46.966046 [**] SMB Name Wildcard [**] MY.NET.101.160:137 -> MY.NET.101.192:137^M
12/22-19:00:05.735938 [**] SMB Name Wildcard [**] MY.NET.101.160:137 -> MY.NET.101.192:137^M
12/22-19:01:14.142539 [**] SMB Name Wildcard [**] MY.NET.101.160:137 -> MY.NET.101.192:137^M
12/22-19:11:50.369869 [**] SMB Name Wildcard [**] MY.NET.101.160:137 -> MY.NET.101.192:137^M
12/28-01:15:04.478612 [**] SMB Name Wildcard [**] 141.157.104.204:137 -> MY.NET.6.15:137^M

12/28-01:15:05.968578 [**] SMB Name Wildcard [**] 141.157.104.204:137 -> MY.NET.6.15:137^M
12/28-01:15:06.162487 [**] SMB Name Wildcard [**] 141.157.104.204:137 -> MY.NET.6.15:137^M
12/28-01:15:07.471875 [**] SMB Name Wildcard [**] 141.157.104.204:137 -> MY.NET.6.15:137^M
12/28-01:15:07.641522 [**] SMB Name Wildcard [**] 141.157.104.204:137 -> MY.NET.6.15:137^M
12/28-01:15:14.994688 [**] SMB Name Wildcard [**] 141.157.104.204:137 -> MY.NET.6.15:137^M
12/28-01:15:15.186736 [**] SMB Name Wildcard [**] 141.157.104.204:137 -> MY.NET.6.15:137^M
12/28-01:15:15.889467 [**] SMB Name Wildcard [**] 141.157.104.204:137 -> MY.NET.6.15:137^M
12/28-01:15:16.455235 [**] SMB Name Wildcard [**] 141.157.104.204:137 -> MY.NET.6.15:137^M
12/28-01:15:16.660256 [**] SMB Name Wildcard [**] 141.157.104.204:137 -> MY.NET.6.15:137^M
12/28-01:15:16.671322 [**] SMB Name Wildcard [**] 141.157.104.204:137 -> MY.NET.6.15:137^M
12/28-01:15:17.392894 [**] SMB Name Wildcard [**] 141.157.104.204:137 -> MY.NET.6.15:137^M

This is probably an attempt to connect to share network devices. Although there appears to be suspicious activity of outside machines trying to connect to my.net shares, this does not look that it succeeded. It is concerning that these machines know enough about my.net to query for a netbios shared device. But, again quoting Steven Northcutt in his Network Intrusion Detection – An Analyst's Handbook "one of the characteristics of NetBIOS is that traffice to destination port UDP 137 is often caused by something a site initiates."

Russia Dynamo - SANS Flash 28-jul-00

12/08-15:36:30.735338 [**] Russia Dynamo - SANS Flash 28-jul-00 [**] MY.NET.205.138:6699 -> 194.87.6.38:2478^M
12/08-15:36:36.529133 [**] Russia Dynamo - SANS Flash 28-jul-00 [**] MY.NET.205.138:6699 -> 194.87.6.38:2478^M
12/08-15:36:54.688783 [**] Russia Dynamo - SANS Flash 28-jul-00 [**] MY.NET.205.138:6699 -> 194.87.6.38:2478^M
12/08-15:37:02.727540 [**] Russia Dynamo - SANS Flash 28-jul-00 [**] MY.NET.205.138:6699 -> 194.87.6.38:2478^M
12/08-15:37:04.280071 [**] Russia Dynamo - SANS Flash 28-jul-00 [**] MY.NET.205.138:6699 -> 194.87.6.38:2478^M
12/08-15:37:11.982432 [**] Russia Dynamo - SANS Flash 28-jul-00 [**] MY.NET.205.138:6699 -> 194.87.6.38:2478^M
12/08-15:37:12.356256 [**] Russia Dynamo - SANS Flash 28-jul-00 [**] 194.87.6.38:2478 -> MY.NET.205.138:6699^M

12/08-15:37:14.520429 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
MY.NET.205.138:6699 -> 194.87.6.38:2478^M
12/08-15:37:15.025826 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
MY.NET.205.138:6699 -> 194.87.6.38:2478^M
12/08-15:37:15.026159 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
MY.NET.205.138:6699 -> 194.87.6.38:2478^M
12/08-15:37:19.627507 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
MY.NET.205.138:6699 -> 194.87.6.38:2478^M
12/08-15:37:20.135241 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
MY.NET.205.138:6699 -> 194.87.6.38:2478^M
12/08-15:37:22.507959 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
MY.NET.205.138:6699 -> 194.87.6.38:2478^M
12/08-15:37:25.576289 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
MY.NET.205.138:6699 -> 194.87.6.38:2478^M
12/08-15:37:27.210635 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
MY.NET.205.138:6699 -> 194.87.6.38:2478^M
12/08-15:37:29.869205 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
MY.NET.205.138:6699 -> 194.87.6.38:2478^M
12/08-15:37:31.064003 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
194.87.6.38:2478 -> MY.NET.205.138:6699^M
12/08-15:37:31.504070 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
MY.NET.205.138:6699 -> 194.87.6.38:2478^M
12/08-15:37:32.283505 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
MY.NET.205.138:6699 -> 194.87.6.38:2478^M
12/08-15:37:33.116016 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
194.87.6.38:2478 -> MY.NET.205.138:6699^M
12/08-15:37:33.378470 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
MY.NET.205.138:6699 -> 194.87.6.38:2478^M
12/08-15:37:35.310908 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
MY.NET.205.138:6699 -> 194.87.6.38:2478^M
12/08-15:37:35.759820 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
MY.NET.205.138:6699 -> 194.87.6.38:2478^M
12/08-15:37:35.993009 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
MY.NET.205.138:6699 -> 194.87.6.38:2478^M
12/08-15:37:35.993578 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
MY.NET.205.138:6699 -> 194.87.6.38:2478^M
12/08-15:37:35.994085 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
MY.NET.205.138:6699 -> 194.87.6.38:2478^M
12/08-15:37:36.273115 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
MY.NET.205.138:6699 -> 194.87.6.38:2478^M
12/08-15:37:44.155665 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
MY.NET.205.138:6699 -> 194.87.6.38:2478^M
12/08-15:37:46.598110 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
194.87.6.38:2478 -> MY.NET.205.138:6699^M
12/08-15:37:47.474360 [**] Russia Dynamo - SANS Flash 28-jul-00 [**]
MY.NET.205.138:6699 -> 194.87.6.38:2478^M

12/08-15:37:49.317923 [**] Russia Dynamo - SANS Flash 28-jul-00 [**] 194.87.6.38:2478 -> MY.NET.205.138:6699^M
12/08-15:37:49.318515 [**] Russia Dynamo - SANS Flash 28-jul-00 [**] MY.NET.205.138:6699 -> 194.87.6.38:2478^M
12/08-15:37:50.408488 [**] Russia Dynamo - SANS Flash 28-jul-00 [**] 194.87.6.38:2478 -> MY.NET.205.138:6699^M
12/08-15:37:50.410346 [**] Russia Dynamo - SANS Flash 28-jul-00 [**] MY.NET.205.138:6699 -> 194.87.6.38:2478^M
12/08-15:37:50.410627 [**] Russia Dynamo - SANS Flash 28-jul-00 [**] MY.NET.205.138:6699 -> 194.87.6.38:2478^M

These are confusing alerts. I do not have access to the snort rule that generated this alert so I went to SANS to look at the July 28 advisory. In the advisory, it says “SANS Flash Report: Trojans Sending More Data To Russia July 28, 2000, 6:20 pm, EDT

After your announcement, I took a look at our router logs for traffic to/from that netblock for the last couple of days. Although the number of probes is relatively small, it looks much more like RingZero than the traces you forwarded in your alert -- note the probes of ports 80, 8080, and 3128. The strange thing is that port 7778...7777 is a default napster port. Could this be an attempt to find proxies/servers to store MP3s for the imminent shutdown of napster? That would explain the large volume of data. I'm attaching the pertinent router netflow data to add to your investigation. Jane DelFavero” And even though port 6699 is not mentioned in the advisory, it is used by Napster and could explain the alert. This may have been the admin checking to see how much of this activity is taking place.

Broadcast Ping to subnet 70

Although there were 157 broadcast pings to subnet 70 in my.net, there seemed to be more activity from a couple of addresses: 194.102.93.101 a, 193.231.220.137, 213.154.131.131:

12/01-19:29:13.430607 [**] Broadcast Ping to subnet 70 [**] 213.154.131.131 -> MY.NET.70.255
12/01-19:29:45.831685 [**] Broadcast Ping to subnet 70 [**] 213.154.131.131 -> MY.NET.70.255
12/01-19:30:37.858481 [**] Broadcast Ping to subnet 70 [**] 213.154.131.131 -> MY.NET.70.255
12/01-19:33:45.862162 [**] Broadcast Ping to subnet 70 [**] 213.154.131.131 -> MY.NET.70.255
12/01-19:36:01.982198 [**] Broadcast Ping to subnet 70 [**] 213.154.131.131 -> MY.NET.70.255

12/01-20:19:57.473529 [**] Broadcast Ping to subnet 70 [**] 213.154.131.131 -> MY.NET.70.255
12/01-20:21:20.371790 [**] Broadcast Ping to subnet 70 [**] 213.154.131.131 -> MY.NET.70.255
12/01-20:21:28.235084 [**] Broadcast Ping to subnet 70 [**] 213.154.131.131 -> MY.NET.70.255
12/01-20:21:34.721425 [**] Broadcast Ping to subnet 70 [**] 213.154.131.131 -> MY.NET.70.255
Broadcast pings generate many packets with very little effort. It appears that someone was trying to cause a denial of service on subnet 70. I would guess that the source addresses are spoofed.

Queso Fingerprint

Queso is scanning tool that fingerprints a network. According to Toby Miller in Global Incident Analysis Center - Detects Analyzed 7/25/00 – it is characterized by the use of high source ports; by sending two (2) SYN packets with the reserved bits set; by sending out a variety of packets including SYNs, SYN | ACKs, PUSH, SYN | FIN, FIN, and FIN | ACK and by allowing the user to scan what ports are desired.

11/28-12:02:21.000796 [**] Queso fingerprint [**] 206.65.191.129:43747 -> MY.NET.219.114:568^M
11/28-12:02:21.000845 [**] Queso fingerprint [**] 206.65.191.129:43748 -> MY.NET.219.114:969^M
11/28-12:02:25.679423 [**] Queso fingerprint [**] 206.65.191.129:43981 -> MY.NET.219.114:147^M
11/28-12:02:28.438916 [**] Queso fingerprint [**] 206.65.191.129:44141 -> MY.NET.219.114:775^M
11/28-12:02:28.738746 [**] Queso fingerprint [**] 206.65.191.129:44154 -> MY.NET.219.114:8^M
11/28-12:02:32.798485 [**] Queso fingerprint [**] 206.65.191.129:44334 -> MY.NET.219.114:524^M
11/28-12:02:32.798587 [**] Queso fingerprint [**] 206.65.191.129:44335 -> MY.NET.219.114:567^M
11/28-12:02:33.359326 [**] Queso fingerprint [**] 206.65.191.129:44365 -> MY.NET.219.114:1512^M
11/28-12:02:33.359381 [**] Queso fingerprint [**] 206.65.191.129:44366 -> MY.NET.219.114:222^M
11/28-12:02:33.359486 [**] Queso fingerprint [**] 206.65.191.129:44367 -> MY.NET.219.114:1485^M
11/28-12:02:33.359542 [**] Queso fingerprint [**] 206.65.191.129:44368 -> MY.NET.219.114:370^M

11/28-12:02:37.537746 [**] Queso fingerprint [**] 206.65.191.129:44538 -> MY.NET.219.114:625^M
11/28-12:02:37.537803 [**] Queso fingerprint [**] 206.65.191.129:44539 -> MY.NET.219.114:1421^M
11/28-12:02:37.537947 [**] Queso fingerprint [**] 206.65.191.129:44540 -> MY.NET.219.114:1510^M
11/28-12:02:37.538002 [**] Queso fingerprint [**] 206.65.191.129:44541 -> MY.NET.219.114:627^M
11/28-12:02:37.538056 [**] Queso fingerprint [**] 206.65.191.129:44543 -> MY.NET.219.114:3456^M
11/28-12:02:39.057027 [**] Queso fingerprint [**] 206.65.191.129:44623 -> MY.NET.219.114:157^M
11/28-12:02:39.057082 [**] Queso fingerprint [**] 206.65.191.129:44624 -> MY.NET.219.114:1997^M
11/28-12:02:39.057261 [**] Queso fingerprint [**] 206.65.191.129:44625 -> MY.NET.219.114:3900^M
11/28-12:02:39.057313 [**] Queso fingerprint [**] 206.65.191.129:44626 -> MY.NET.219.114:1484^M
11/28-12:02:39.057389 [**] Queso fingerprint [**] 206.65.191.129:44627 -> MY.NET.219.114:26^M
11/28-12:02:39.057472 [**] Queso fingerprint [**] 206.65.191.129:44628 -> MY.NET.219.114:680^M
11/28-12:02:39.378814 [**] Queso fingerprint [**] 206.65.191.129:44646 -> MY.NET.219.114:670^M
11/28-12:02:39.378870 [**] Queso fingerprint [**] 206.65.191.129:44647 -> MY.NET.219.114:977^M
11/28-12:02:40.318167 [**] Queso fingerprint [**] 206.65.191.129:44693 -> MY.NET.219.114:1652^M
11/28-12:02:40.318218 [**] Queso fingerprint [**] 206.65.191.129:44695 -> MY.NET.219.114:5303^M
11/28-12:02:40.318270 [**] Queso fingerprint [**] 206.65.191.129:44696 -> MY.NET.219.114:212^M
11/28-12:02:40.318321 [**] Queso fingerprint [**] 206.65.191.129:44698 -> MY.NET.219.114:172^M
You need to be concerned anytime fingerprinting is happening. It may be a precursor to a real attack.

WinGate 1080 Attempt

01/07-22:22:09.813300 [**] WinGate 1080 Attempt [**] 4.41.126.148:4908 -> MY.NET.60.38:1080^M
01/07-23:06:28.500003 [**] WinGate 1080 Attempt [**] 203.134.52.163:2152 -> MY.NET.60.38:1080^M

01/07-23:06:29.600898 [**] WinGate 1080 Attempt [**] 203.134.52.163:2152 -> MY.NET.60.38:1080^M
01/07-23:13:09.782001 [**] WinGate 1080 Attempt [**] 209.212.128.47:1103 -> MY.NET.98.194:1080^M
01/07-23:13:58.988698 [**] WinGate 1080 Attempt [**] 205.136.57.121:3805 -> MY.NET.98.194:1080^M
01/07-23:27:07.098718 [**] WinGate 1080 Attempt [**] 204.117.70.5:1880 -> MY.NET.60.38:1080^M
01/07-23:31:41.608603 [**] WinGate 1080 Attempt [**] 203.134.52.163:2473 -> MY.NET.60.8:1080^M
01/07-23:31:42.717590 [**] WinGate 1080 Attempt [**] 203.134.52.163:2473 -> MY.NET.60.8:1080^M
01/07-23:31:43.809601 [**] WinGate 1080 Attempt [**] 203.134.52.163:2473 -> MY.NET.60.8:1080^M
01/08-00:42:09.432282 [**] WinGate 1080 Attempt [**] 208.185.24.9:3007 -> MY.NET.219.154:1080^M
01/08-01:01:21.497074 [**] WinGate 1080 Attempt [**] 198.63.2.192:3620 -> MY.NET.98.221:1080^M
01/08-01:54:50.178182 [**] WinGate 1080 Attempt [**] 206.105.43.16:1470 -> MY.NET.97.41:1080^M
01/08-01:54:50.940925 [**] WinGate 1080 Attempt [**] 206.105.43.16:1470 -> MY.NET.97.41:1080^M
01/08-01:54:51.640812 [**] WinGate 1080 Attempt [**] 206.105.43.16:1470 -> MY.NET.97.41:1080^M
01/08-01:54:51.829801 [**] WinGate 1080 Attempt [**] 213.61.112.10:2418 -> MY.NET.97.41:1080^M
01/08-01:54:51.891630 [**] WinGate 1080 Attempt [**] 213.61.112.10:2419 -> MY.NET.97.41:1080^M
01/08-01:54:51.906043 [**] WinGate 1080 Attempt [**] 213.61.112.10:2420 -> MY.NET.97.41:1080^M
01/08-01:54:51.906186 [**] WinGate 1080 Attempt [**] 213.61.112.10:2421 -> MY.NET.97.41:1080^M

A Wingate or Socks proxy server generally operate on ports 8080 and 1080. There are several exploits of Wingate proxies, including undetected access through the proxy. However, even though there are approx 200 of these, they are not arriving at a big frequency. This could be an example of attempts to locate a wingate proxy or it could mean that an IRC server is merely checking for a mis-configured Wingate or SOCKS proxy.

SUNRPC highport access

01/15-16:14:02.922115 [**] SUNRPC highport access! [**] 64.4.13.74:1863 -> MY.NET.98.199:32771^M
01/15-16:14:22.843477 [**] SUNRPC highport access! [**] 64.4.13.74:1863 -> MY.NET.98.199:32771^M
01/15-16:14:57.840949 [**] SUNRPC highport access! [**] 64.4.13.74:1863 -> MY.NET.98.199:32771^M
01/15-16:15:02.856726 [**] SUNRPC highport access! [**] 64.4.13.74:1863 -> MY.NET.98.199:32771^M
01/15-16:15:27.282224 [**] SUNRPC highport access! [**] 64.4.13.74:1863 -> MY.NET.98.199:32771^M
01/15-16:15:57.952991 [**] SUNRPC highport access! [**] 64.4.13.74:1863 -> MY.NET.98.199:32771^M
01/15-16:16:12.999754 [**] SUNRPC highport access! [**] 64.4.13.74:1863 -> MY.NET.98.199:32771^M
01/15-16:16:52.113058 [**] SUNRPC highport access! [**] 64.4.13.74:1863 -> MY.NET.98.199:32771^M
01/15-16:16:52.123816 [**] SUNRPC highport access! [**] 64.4.13.74:1863 -> MY.NET.98.199:32771^M
01/15-16:17:03.073912 [**] SUNRPC highport access! [**] 64.4.13.74:1863 -> MY.NET.98.199:32771^M
01/15-16:17:38.034816 [**] SUNRPC highport access! [**] 64.4.13.74:1863 -> MY.NET.98.199:32771^M
01/16-01:59:15.223453 [**] SUNRPC highport access! [**] 205.188.4.6:5190 -> MY.NET.218.238:32771^M
01/16-01:59:15.231934 [**] SUNRPC highport access! [**] 205.188.4.6:5190 -> MY.NET.218.238:32771^M
01/16-01:59:15.448343 [**] SUNRPC highport access! [**] 205.188.4.6:5190 -> MY.NET.218.238:32771^M
01/16-01:59:15.457213 [**] SUNRPC highport access! [**] 205.188.4.6:5190 -> MY.NET.218.238:32771^M

Is this more of the ICR connection with AOL? This is very similar to the Sun Highport Attempt, except that the source port is different. The early scans consistently used port 4000, characteristic with ICQ. Even though these do not used port 4000, the most frequent alert:

SUNRPC highport access! activity from 205.188.153.139 to MY.NET.213.158 91 time(s).

Does refer to an AOL resource. There are not many of these, 204 in two months. But you may want to keep the alert in case the activity increases.

Null Scans

12/01-08:22:56.033419 [**] Null scan! [**] 213.56.48.243:1981 -> MY.NET.212.38:4742

12/01-11:18:13.974388 [**] Null scan! [**] 62.27.226.104:1246 -> MY.NET.209.118:3
12/01-14:14:53.299732 [**] Null scan! [**] 24.69.140.11:6699 -> MY.NET.226.34:4061
12/02-19:45:03.676468 [**] Null scan! [**] 24.3.197.41:15740 -> MY.NET.211.58:6346
12/03-05:09:31.301470 [**] Null scan! [**] 212.83.81.80:21045 -> MY.NET.222.210:4368
12/03-07:54:40.688704 [**] Null scan! [**] 24.6.36.118:1801 -> MY.NET.225.50:4020
12/03-12:17:26.257524 [**] Null scan! [**] 24.202.65.165:1932 -> MY.NET.212.214:2854
12/03-15:09:44.490386 [**] Null scan! [**] 62.137.125.36:12288 -> MY.NET.208.94:0
12/03-19:28:59.158312 [**] Null scan! [**] 62.137.162.139:18245 -> MY.NET.204.82:21504
12/04-05:42:37.941259 [**] Null scan! [**] 143.233.243.171:0 -> MY.NET.207.122:1064
12/05-00:52:30.484006 [**] Null scan! [**] 24.169.52.151:1164 -> MY.NET.201.78:1071
12/05-08:10:47.306236 [**] Null scan! [**] 24.217.79.34:1189 -> MY.NET.206.122:6346
12/05-12:44:12.083980 [**] Null scan! [**] 200.52.185.253:2719 -> MY.NET.221.26:41033
12/05-14:12:25.029881 [**] Null scan! [**] 24.112.182.54:2265 -> MY.NET.205.66:6688
12/05-19:10:22.075613 [**] Null scan! [**] 24.232.90.152:1620 -> MY.NET.202.62:6688
12/05-20:01:58.484602 [**] Null scan! [**] 141.213.201.76:1 -> MY.NET.204.206:2927
12/05-20:50:24.972078 [**] Null scan! [**] 24.150.65.89:231 -> MY.NET.179.209:2224

A NULL scan is utilized to map out a network. This is probably a reconnaissance of MY.NET and is probaly the prelude to a more directed attack

----

SNORT Out Of Spec scans

Looking the Out of Spec files, I see most probes for listeners on specific ports, i.e, Ports 21, 53, 109, etc.

01/18-20:15:48.329009 MY.NET.217.150:0 -> 216.232.142.129:2340
TCP TTL:126 TOS:0x0 ID:54060 DF
21*F*P*U Seq: 0x52210592 Ack: 0x4B070093 Win: 0x5010
TCP Options => EOL EOL

=+=
+=+=+

```
01/18-20:15:49.519906 193.253.202.9:21 -> MY.NET.104.125:21
TCP TTL:27 TOS:0x0 ID:39426
**SF**** Seq: 0x4ECC4D11   Ack: 0x4BC10674   Win: 0x404
00 00 00 00 00 00                                ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+=+
01/18-20:15:51.637201 193.253.202.9:21 -> MY.NET.104.231:21
TCP TTL:27 TOS:0x0 ID:39426
**SF**** Seq: 0x2A721919   Ack: 0x354BAE8F   Win: 0x404
00 00 00 00 00 00                                ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+=+
01/18-20:16:01.986307 193.253.202.9:21 -> MY.NET.106.233:21
TCP TTL:27 TOS:0x0 ID:39426
**SF**** Seq: 0x63DE9740   Ack: 0x3BEF3E38   Win: 0x404
00 00 00 00 00 00                                ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+=+
01/18-20:16:07.377069 193.253.202.9:21 -> MY.NET.107.247:21
TCP TTL:27 TOS:0x0 ID:39426
**SF**** Seq: 0x48A1892D   Ack: 0x43C33623   Win: 0x404
00 00 00 00 00 00                                ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+=+
01/18-20:16:29.601475 MY.NET.217.150:134 -> 216.232.142.129:2340
TCP TTL:126 TOS:0x0 ID:39789  DF
21*F*P*U Seq: 0x522105D0   Ack: 0x58870093   Win: 0x5010
TCP Options => EOL EOL
```

However, in the noise of all the probing, there were several servers that deserved closer looks. It seems that my.nt.217.182, my.net.217.126, my.net.217.150 and my.net.217.158 were generating OOS packets were were terribly screwed up. The datagrams from a couple of the devices are listed below.

```
01/14-15:14:25.827711 MY.NET.217.158:2340 -> 24.189.31.228:4251
TCP TTL:126 TOS:0x0 ID:6918  DF
2*SFRP** Seq: 0x7EFAFB0   Ack: 0x242   Win: 0x5010
TCP Options => EOL EOL NOP Opt 248 (6): 2050 9805 CC 1979735040 CC
1979735040 CC 1979735040 CC 1979735040 CC 1979735040 CC 197973504
```

0 CC 1979735040 CC 1979735040 CC 1979735040 CC 1979735040 CC
1979735040 CC 1979735040 CC 1979735040 CC 1979735040 CC 1979735040
CC 1
979735040 CC 1979735040 CC 1979735040 CC 1979735040 CC 1979735040
CC 1979735040 CC 1979735040 CC 1979735040 CC 1979735040 CC 1979735
040 CC 1979735040 CC 1979735040 CC 1979735040 CC 1979735040 CC
1979735040 CC 1979735040 CC 1979735040 CC 1979735040 CC 1979735040
CC
 1979735040 CC 1979735040

=+=
+=+=+
01/14-15:14:59.076934 MY.NET.217.158:2340 -> 213.89.116.21:1862
TCP TTL:126 TOS:0x0 ID:55328 DF
2*SFRP** Seq: 0x3D07BB   Ack: 0x8A5BBB00   Win: 0x5010
FD EA 04 9B 00 00 15 B0 FC 84 2E 81 04 F3 C8 79 ....y
06 92   ..

=+=
+=+=+
01/14-15:15:04.256910 MY.NET.217.158:1 -> 213.89.116.21:2340
TCP TTL:126 TOS:0x0 ID:40997 DF
2*SFRP** Seq: 0x74607BD   Ack: 0x2AAFBB00   Win: 0x5018
FD EA 61 0F 00 00 EE 59 F0 B1 F0 6B F2 4A F5 A4 ..a....Y...k.J..
F4 EE   ..

=+=
+=+=+
01/14-15:15:27.448904 MY.NET.217.158:2340 -> 24.189.31.228:4251
TCP TTL:126 TOS:0x0 ID:52534 DF
2*SFRP** Seq: 0x8B84708   Ack: 0x640242   Win: 0x5010
TCP Options => EOL EOL

=+=
+=+=+
01/14-15:15:41.845930 MY.NET.217.158:2340 -> 24.189.31.228:4251
TCP TTL:126 TOS:0x0 ID:55619 DF
2*SFRP** Seq: 0x8DF   Ack: 0xAC640242   Win: 0x5010
TCP Options => EOL EOL Opt 21 Opt 21 Opt 21 Opt 21 Opt 21 Opt 21 Opt 21
Opt 21 Opt 21 Opt 21 Opt 21 Opt 21 Opt 21 Opt 21 Opt 21 Opt
21 Opt 21 Opt 21 Opt 21 Opt 21 Opt 21 Opt 21 Opt 21 Opt 21 Opt 21 Opt 21 Opt
21 Opt 21 Opt 21 Opt 21 Opt 21 Opt 21 Opt 21 Opt 21 Opt
 21 Opt 21 Opt 21 Opt 21

=+=
+=+=+

01/15-08:54:22.563537 MY.NET.217.150:140 -> 65.6.108.84:2340
TCP TTL:126 TOS:0x0 ID:61068 DF
SFAU Seq: 0x1304042B Ack: 0x9A3C029D Win: 0x5018
TCP Options => EOL EOL Opt 92 Opt 92 Opt 92 Opt 92 Opt 92 Opt 92 Opt 92
Opt 92 Opt 92 Opt 92 Opt 92 Opt 92 Opt 92 Opt 92 Opt 92 Opt
92 Opt 92 Opt 92 Opt 92 Opt 92 Opt 92 Opt 92 Opt 92 Opt 92 Opt 92 Opt 92 Opt
92 Opt 92 Opt 92 Opt 92 Opt 92 Opt 92 Opt 92 Opt 92 Op
92 Opt 92 Opt 92 Opt 92

=+=
+=+=+
01/15-08:55:43.009243 MY.NET.217.150:2340 -> 65.6.108.84:4868
TCP TTL:126 TOS:0x0 ID:17045 DF
SFAU Seq: 0xA0433 Ack: 0x6654029D Win: 0x5010
TCP Options => EOL EOL

=+=
+=+=+
01/15-08:57:11.538261 MY.NET.217.150:2340 -> 206.167.141.48:3436
TCP TTL:126 TOS:0x0 ID:48800 DF
21SFRPAU Seq: 0x3759494 Ack: 0x7A8F5455 Win: 0x5018
09 24 0D 6C 03 75 94 94 7A 8F 54 55 00 FF 50 18 .$.l.u..z.TU..P.
FD E5 DD 4D 00 00 5C 78 09 01 F6 79 51 47 C1 AC ...M..\x...yQG..
07 4D .M

=+=
+=+=+
01/15-09:00:22.369947 MY.NET.217.150:202 -> 212.187.104.145:2340
TCP TTL:126 TOS:0x0 ID:38073 DF
2*SF*P** Seq: 0x10D7034F Ack: 0xD9B6DA55 Win: 0x5010
00 CA 09 24 10 D7 03 4F D9 B6 DA 55 0E 4B 50 10 ...$...O...U.KP.
FD E8 8A FA 00 00 FD 70 EE C8 FD DB EE 18 FE 50 ....p.......P
EF 6E .n

=+=
+=+=+

Even though there could be hardware solutions for these devices, they do seem to be talking to multiple devices across multiple ports. And they are on the same subnet. I would seem terribly coincidental for all this to be just happening. It appears that the packet may have been crafted to get past firewalls and IDS systems and deliver a questionable payload. These servers seem to be communicating with other devices outside my.net with varying sizes of payload. You may want to shut these servers down, or at least look at them to see what is going on.

Summary

The collection of snort scans produced the following traps:

Rule	Number of hits
DNS udp DoS attack described on unisog	16146
STATDX UDP attack	1
connect to 515 from inside	159
connect to 515 from outside	4238
spp_portscan: portscan status from	221176
Watchlist 000220 IL-ISDNNET-990517	105918
Watchlist 000222 NET-NCFC	2401
SITE EXEC - Possible wu-ftpd exploit - GIAC000623	3
TCP SMTP Source Port traffic	100
Happy 99 Virus	1
Tiny Fragments - Possible Hostile Activity	5340
Back Orifice	77
External RPC call	59
Attempted Sun RPC high port access	2053
NMAP TCP ping!	558
site exec - Possible wu-ftpd exploit - GIAC000623	2
SMB Name Wildcard	515
Russia Dynamo - SANS Flash 28-jul-00	546
Broadcast Ping to subnet 70	154
Queso fingerprint	710
WinGate 1080 Attempt	2239
Probable NMAP fingerprint attempt	8
Null scan!	826
SUNRPC highport access!	204
SNMP public access	591
SYN-FIN scan!	51192

As can be seen, there were many alerts triggered by snort. As noted above, some deserve looking at a little closer. Some are obviously false positives. As a recommendation, I would install a firewall to protect my.net from the probes, scans and attacks that are obvious from the alerts and scans. If you can block most of the probes, or at least responses to these probes, then you can help minimize the possibility of compromise. Next, as noted above, the snort ids should be reevaluated and adjusted to minimize the number of false positives. This will also save the time your analyst would have to examine the logs looking for attacks. I may also suggest adding a different type of ids, a commercial product or something like tcpdump/shadow, to verify the alerts that snort produces. Simply put, snort is doing a good job of pattern matching and alerting

when a match is found. However, you may want to take proactive actions, like installation a firewall, to product my.net from the world.

Last Updated: May 27th, 2017

Upcoming Training

Security Operations Center Summit & Training	**Washington, DC**	**Jun 05, 2017 - Jun 12, 2017**	**Live Event**
SANS Houston 2017	**Houston, TX**	**Jun 05, 2017 - Jun 10, 2017**	**Live Event**
SANS Columbia, MD 2017	**Columbia, MD**	**Jun 26, 2017 - Jul 01, 2017**	**Live Event**
SANS London July 2017	**London, United Kingdom**	**Jul 03, 2017 - Jul 08, 2017**	**Live Event**
SANSFIRE 2017	**Washington, DC**	**Jul 22, 2017 - Jul 29, 2017**	**Live Event**
SANSFIRE 2017 - SEC503: Intrusion Detection In-Depth	**Washington, DC**	**Jul 24, 2017 - Jul 29, 2017**	**vLive**
SANS San Antonio 2017	**San Antonio, TX**	**Aug 06, 2017 - Aug 11, 2017**	**Live Event**
SANS Boston 2017	**Boston, MA**	**Aug 07, 2017 - Aug 12, 2017**	**Live Event**
SANS Adelaide 2017	**Adelaide, Australia**	**Aug 21, 2017 - Aug 26, 2017**	**Live Event**
SANS Virginia Beach 2017	**Virginia Beach, VA**	**Aug 21, 2017 - Sep 01, 2017**	**Live Event**
SANS Network Security 2017	**Las Vegas, NV**	**Sep 10, 2017 - Sep 17, 2017**	**Live Event**
SANS vLive - SEC503: Intrusion Detection In-Depth	**SEC503 - 201709,**	**Sep 11, 2017 - Oct 18, 2017**	**vLive**
Baltimore September 2017 - SEC503: Intrusion Detection In-Depth	**Baltimore, MD**	**Sep 25, 2017 - Sep 30, 2017**	**vLive**
SANS Baltimore Fall 2017	**Baltimore, MD**	**Sep 25, 2017 - Sep 30, 2017**	**Live Event**
SANS London September 2017	**London, United Kingdom**	**Sep 25, 2017 - Sep 30, 2017**	**Live Event**
Community SANS Scottsdale SEC503	**Scottsdale, AZ**	**Oct 02, 2017 - Oct 07, 2017**	**Community SANS**
SANS October Singapore 2017	**Singapore, Singapore**	**Oct 09, 2017 - Oct 28, 2017**	**Live Event**
Community SANS Boston SEC503	**Boston, MA**	**Oct 09, 2017 - Oct 14, 2017**	**Community SANS**
Community SANS Ottawa SEC503	**Ottawa, ON**	**Oct 16, 2017 - Oct 21, 2017**	**Community SANS**
SANS Seattle 2017	**Seattle, WA**	**Oct 30, 2017 - Nov 04, 2017**	**Live Event**
SANS San Diego 2017	**San Diego, CA**	**Oct 30, 2017 - Nov 04, 2017**	**Live Event**
San Diego Fall 2017 - SEC503: Intrusion Detection In-Depth	**San Diego, CA**	**Oct 30, 2017 - Nov 04, 2017**	**vLive**
SIEM & Tactical Analytics Summit & Training	**Scottsdale, AZ**	**Nov 28, 2017 - Dec 05, 2017**	**Live Event**
SANS Cyber Defense Initiative 2017	**Washington, DC**	**Dec 12, 2017 - Dec 19, 2017**	**Live Event**
SANS OnDemand	**Online**	**Anytime**	**Self Paced**
SANS SelfStudy	**Books & MP3s Only**	**Anytime**	**Self Paced**