Global Information Assurance Certification Paper

Fred Portnoy

GCIA Practical – Intrusion Detection, New Orleans, 2001

I. NETWORK DETECTS

1. Source of Trace:
2. Detect was generated by:
3. Probability the source address was spoofed:
4. Description of attack:
5. Attack mechanism:
6. Correlation:
7. Evidence of active targeting:
8. Severity:
9. Defensive recommendation:
10. Multiple choice test question:

DETECT #1: PORT 27374 SNIFFER® TRACES

General examinations of activity on port 27374. This is an interesting study of what can happen if you pursue a hunch or a curiosity. Sometimes a stupid question can show up unexpected answers, and/or can bury one in many more questions. This came from Sniffer traces set up to filter on 27374 at the offset of UDP/TCP source/destination ports. I got mired in seeing 27374 showing up as ICMP checksums, at the same IP offset, which may well have been incidental, over time, but I also saw some host scans for exposures.

Sniffer Traces

Frame - Frame number in trace.
Status - Whether the packet is marked as a reference point in the trace.
Source Address - From the IP header
Destination Address - From the IP header
Size - total length of the captured packet. The Sniffer HEX display doesn't include the ethernet trailing checksum, so minimum Ethernet frame is 60 bytes.
Rel. Time - The elapsed time since the trace began or from a set marker.

Delta Time - The time since the previous frame.
Abs. Time - The date and time-of-day the packet was received by the Sniffer.
Summary - The Sniffer®®'s brief description of the packet, containing any messages generated by the Sniffer®® "Expert" followed by a summary of the highest level header information the Sniffer®® was able to decode from the packet.
HEX section - the ADDR column is simply for ease of reading the HEX information, showing the HEX representation of the offset location of the first byte in each row.
- the HEX section shows the hexadecimal representation of each byte in the packet.
- the ASCII section shows the ASCII translation of each byte that maps to the ASCII code.

These traces include the ethernet headers.

These first two frames are part of a dns ddos and probably just happen to have the source port 27375.

```
- - - - - - - - - - - - - - - - - - - - Frame 3 - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time              Summary
     3        [194.204.49.250]  [MY.NET.1.99]       67 1:46:35.780   61.555.814
02/19/2001 07:04:33 PM DNS: C ID=19483 OP=QUERY NAME=aol.com
ADDR  HEX                                                ASCII
0000: 00 90 27 46 4e f5 00 00 ef 06 1f 50 08 00 45 00 | ..'FN......P..E.
0010: 00 35 88 d3 40 00 32 11 2b 33 c2 cc 31 fa xx xx | .5.Ó@.2.+3..1...
0020: 01 63 6a ee 00 35 00 21 00 00 4c 1b 01 00 00 01 | .cj..5.!..L.....
0030: 00 00 00 00 00 00 03 61 6f 6c 03 63 6f 6d 00 00 | .......aol.com..
0040: 0f 00 01                                       | ...

- - - - - - - - - - - - - - - - - - - - Frame 4 - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time              Summary
     4        [194.204.49.254]  [MY.NET.1.100]      67 1:53:38.790   423.009.646
02/19/2001 07:11:36 PM DNS: C ID=36367 OP=QUERY NAME=aol.com
ADDR  HEX                                                ASCII
0000: 00 90 27 9c 1e 0a 00 00 ef 06 1f 50 08 00 45 00 | ..'........P..E.
0010: 00 35 50 6a 40 00 32 11 63 97 c2 cc 31 fe xx xx | .5Pj@.2.c...1þ..
0020: 01 64 6a ee 00 35 00 21 00 00 8e 0f 01 00 00 01 | .dj..5.!........
0030: 00 00 00 00 00 00 03 61 6f 6c 03 63 6f 6d 00 00 | .......aol.com..
0040: 0f 00 01                                       | ...
```

These next frames are pings. Extra data beyond the "IP header length"?

```
- - - - - - - - - - - - - - - - - - - - Frame 1 - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time              Summary
     1 M      [MY.NET.173.119] [128.206.250.116]    60 0:00:00.000   0.000.000
02/20/2001 02:36:31 AM ICMP: Echo reply
ADDR  HEX                                                ASCII
0000: 00 00 ef 06 1f 50 00 e0 16 7f eb 82 08 00 45 00 | .....P........E.
0010: 00 1c 9b 21 00 00 7e 01 da 7c xx xx ad 77 80 ce | ...!..~..|...w..
0020: fa 74 00 00 6a ee 02 00 93 11 e5 0f 27 19 02 00 | .t..j.......'...
0030: 00 00 00 00 00 00 00 00 00 00 00 00             | ............

Another One


- - - - - - - - - - - - - - - - - - - - Frame 1 - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time              Summary
     1 M      [MY.NET.173.163] [216.161.137.231]    60 0:00:00.000   0.000.000
02/19/2001 05:50:54 PM ICMP: Echo reply
```

```
ADDR  HEX                                                 ASCII
0000: 00 00 ef 06 1f 50 00 e0 16 7f eb 82 08 00 45 00 | .....P........E.
0010: 00 1c f4 e6 00 00 7e 01 99 45 xx xx ad a3 d8 a1 | ......~..E...£..
0020: 89 e7 00 00 6a ee 02 00 93 11 9d 4a 9b 51 02 00 | ....j..... J.Q..
0030: 00 00 00 00 00 00 00 00 00 00 00 00             | ............
```

This time the extra data is in the request frame.

```
- - - - - - - - - - - - - - - - - - - - Frame 1 - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time              Summary
     1 M      [MY.NET.173.182] [152.19.210.81]      60 0:00:00.000   0.000.000
02/20/2001 03:12:37 AM ICMP: Echo
ADDR  HEX                                                 ASCII
0000: 00 00 ef 06 1f 50 00 e0 16 7f eb 82 08 00 45 00 | .....P........E.
0010: 00 1c c7 a2 00 00 7e 01 be 9a xx xx ad b6 98 13 | ..Ç...~.........
0020: d2 51 08 00 6a ee 03 00 8a 11 92 8c 66 ca 02 00 | .Q..j.......f...
0030: 00 00 00 00 00 00 00 00 00 00 00 00             | ............

- - - - - - - - - - - - - - - - - - - - Frame 2 - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time              Summary
     2        [152.19.210.81]   [MY.NET.173.182]    60 0:00:00.072   0.072.496
02/20/2001 03:12:38 AM ICMP: Echo reply
ADDR  HEX                                                 ASCII
0000: 00 e0 16 7f eb 82 00 00 ef 06 1f 50 08 00 45 00 | ...........P..E.
0010: 00 1c 74 32 00 00 74 01 1c 0b 98 13 d2 51 xx xx | ..t2..t......Q..
0020: ad b6 00 00 6a ee 03 00 92 11 00 00 00 00 00 00 | ....j...........
0030: 00 00 00 00 00 00 00 00 00 00 00 00             | ............
```

and so forth ...

```
- - - - - - - - - - - - - - - - - - - - Frame 1 - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time              Summary
     1 M      [172.146.225.142] [MY.NET.174.171]    60 0:00:00.000   0.000.000
02/20/2001 06:07:10 AM ICMP: Echo
ADDR  HEX                                                 ASCII
0000: 00 e0 16 7f eb 82 00 00 ef 06 1f 50 08 00 45 00 | ...........P..E.
0010: 00 1c 1d 9e 00 00 74 01 4d ee ac 92 e1 8e xx xx | ......t.M...á...
0020: ae ab 08 00 6a ee 02 00 8b 11 00 00 00 00 00 00 | ....j...........
0030: 00 00 00 00 00 00 00 00 00 00 00 00             | ............

- - - - - - - - - - - - - - - - - - - - Frame 2 - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time              Summary
     2        [MY.NET.174.171] [172.146.225.142]    60 0:00:00.058   0.058.598
02/20/2001 06:07:10 AM ICMP: Echo reply
ADDR  HEX                                                 ASCII
0000: 00 00 ef 06 1f 50 00 e0 16 7f eb 82 08 00 45 00 | .....P........E.
0010: 00 1c fa f2 00 00 7e 01 66 99 xx xx ae ab ac 92 | ......~.f.......
0020: e1 8e 00 00 6a ee 02 00 93 11 54 bd 34 ec 02 00 | á...j.....T.4...
0030: 00 00 00 00 00 00 00 00 00 00 00 00             | ............
```

****

Here's a TCP scan for destination port 27374. Some hosts respond with a RESET and others don't.

```
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time              Summary
    28        [212.252.28.163]  [MY.NET.211.1]      62 3:16:17.082   624.266.093
02/19/2001 08:34:15 PM TCP: D=27374 S=3368 SYN SEQ=3605236 LEN=0 WIN=8192
    29        [212.252.28.163]  [MY.NET.211.1]      62 3:16:20.062   2.980.694
02/19/2001 08:34:18 PM TCP: D=27374 S=3368 SYN SEQ=3605236 LEN=0 WIN=8192
    30        [212.252.28.163]  [MY.NET.211.1]      62 3:16:26.083   6.020.866
02/19/2001 08:34:24 PM TCP: D=27374 S=3368 SYN SEQ=3605236 LEN=0 WIN=8192
    31        [212.252.28.163]  [MY.NET.211.1]      62 3:16:38.148   12.064.969
02/19/2001 08:34:36 PM TCP: D=27374 S=3368 SYN SEQ=3605236 LEN=0 WIN=8192
```

```
    32        [212.252.28.163]  [MY.NET.211.2]      62 3:17:02.275   24.126.943
02/19/2001 08:35:00 PM TCP: D=27374 S=3385 SYN SEQ=3650381 LEN=0 WIN=8192
    33        [212.252.28.163]  [MY.NET.211.2]      62 3:17:05.258   2.982.916
02/19/2001 08:35:03 PM TCP: D=27374 S=3385 SYN SEQ=3650381 LEN=0 WIN=8192
    34        [212.252.28.163]  [MY.NET.211.2]      62 3:17:11.238   5.979.685
02/19/2001 08:35:09 PM TCP: D=27374 S=3385 SYN SEQ=3650381 LEN=0 WIN=8192
    35        [212.252.28.163]  [MY.NET.211.2]      62 3:17:23.271   12.033.532
02/19/2001 08:35:21 PM TCP: D=27374 S=3385 SYN SEQ=3650381 LEN=0 WIN=8192
    36        [212.252.28.163]  [MY.NET.211.3]      62 3:17:47.439   24.168.174
02/19/2001 08:35:45 PM TCP: D=27374 S=3409 SYN SEQ=3695526 LEN=0 WIN=8192
    37        [212.252.28.163]  [MY.NET.211.3]      62 3:17:50.359   2.919.942
02/19/2001 08:35:48 PM TCP: D=27374 S=3409 SYN SEQ=3695526 LEN=0 WIN=8192
    38        [212.252.28.163]  [MY.NET.211.3]      62 3:17:56.537   6.177.675
02/19/2001 08:35:54 PM TCP: D=27374 S=3409 SYN SEQ=3695526 LEN=0 WIN=8192
    39        [212.252.28.163]  [MY.NET.211.3]      62 3:18:08.484   11.947.492
02/19/2001 08:36:06 PM TCP: D=27374 S=3409 SYN SEQ=3695526 LEN=0 WIN=8192
    40        [212.252.28.163]  [MY.NET.211.4]      62 3:18:32.529   24.044.295
02/19/2001 08:36:30 PM TCP: D=27374 S=3424 SYN SEQ=3740681 LEN=0 WIN=8192
    41        [MY.NET.211.4]   [212.252.28.163]     60 3:18:32.540   0.011.192
02/19/2001 08:36:30 PM TCP: D=3424 S=27374 RST ACK=3740682 WIN=0
    42        [212.252.28.163]  [MY.NET.211.4]      62 3:18:33.639   1.098.622
02/19/2001 08:36:31 PM TCP: D=27374 S=3424 SYN SEQ=3740681 LEN=0 WIN=8192
    43        [MY.NET.211.4]   [212.252.28.163]     60 3:18:33.652   0.012.978
02/19/2001 08:36:31 PM TCP: D=3424 S=27374 RST ACK=3740682 WIN=0
    44        [212.252.28.163]  [MY.NET.211.4]      62 3:18:34.712   1.060.823
02/19/2001 08:36:32 PM TCP: D=27374 S=3424 SYN SEQ=3740681 LEN=0 WIN=8192
    45        [MY.NET.211.4]   [212.252.28.163]     60 3:18:34.731   0.018.339
02/19/2001 08:36:32 PM TCP: D=3424 S=27374 RST ACK=3740682 WIN=0
    46        [212.252.28.163]  [MY.NET.211.4]      62 3:18:35.835   1.104.717
02/19/2001 08:36:33 PM TCP: D=27374 S=3424 SYN SEQ=3740681 LEN=0 WIN=8192
    47        [MY.NET.211.4]   [212.252.28.163]     60 3:18:35.852   0.016.417
02/19/2001 08:36:33 PM TCP: D=3424 S=27374 RST ACK=3740682 WIN=0
    48        [212.252.28.163]  [MY.NET.211.5]      62 3:18:36.556   0.703.706
02/19/2001 08:36:34 PM TCP: D=27374 S=3429 SYN SEQ=3744629 LEN=0 WIN=8192
    49        [MY.NET.211.5]   [212.252.28.163]     60 3:18:36.573   0.017.143
02/19/2001 08:36:34 PM TCP: D=3429 S=27374 RST ACK=3744630 WIN=0
    50        [212.252.28.163]  [MY.NET.211.5]      62 3:18:37.698   1.125.300
02/19/2001 08:36:35 PM TCP: D=27374 S=3429 SYN SEQ=3744629 LEN=0 WIN=8192
    51        [MY.NET.211.5]   [212.252.28.163]     60 3:18:37.712   0.014.213
02/19/2001 08:36:35 PM TCP: D=3429 S=27374 RST ACK=3744630 WIN=0
    52        [212.252.28.163]  [MY.NET.211.5]      62 3:18:38.844   1.132.047
02/19/2001 08:36:36 PM TCP: D=27374 S=3429 SYN SEQ=3744629 LEN=0 WIN=8192
    53        [MY.NET.211.5]   [212.252.28.163]     60 3:18:38.862   0.017.725
02/19/2001 08:36:36 PM TCP: D=3429 S=27374 RST ACK=3744630 WIN=0
    54        [212.252.28.163]  [MY.NET.211.5]      62 3:18:40.019   1.156.607
02/19/2001 08:36:38 PM TCP: D=27374 S=3429 SYN SEQ=3744629 LEN=0 WIN=8192
    55        [MY.NET.211.5]   [212.252.28.163]     60 3:18:40.028   0.009.846
02/19/2001 08:36:38 PM TCP: D=3429 S=27374 RST ACK=3744630 WIN=0
    56        [212.252.28.163]  [MY.NET.211.6]      62 3:18:40.657   0.628.206
02/19/2001 08:36:38 PM TCP: D=27374 S=3434 SYN SEQ=3748811 LEN=0 WIN=8192
    57        [212.252.28.163]  [MY.NET.211.6]      62 3:18:43.659   3.002.727
02/19/2001 08:36:41 PM TCP: D=27374 S=3434 SYN SEQ=3748811 LEN=0 WIN=8192
    58        [212.252.28.163]  [MY.NET.211.6]      62 3:18:49.691   6.031.326
02/19/2001 08:36:47 PM TCP: D=27374 S=3434 SYN SEQ=3748811 LEN=0 WIN=8192
    59        [212.252.28.163]  [MY.NET.211.6]      62 3:19:01.753   12.062.636
02/19/2001 08:36:59 PM TCP: D=27374 S=3434 SYN SEQ=3748811 LEN=0 WIN=8192
    60        [212.252.28.163]  [MY.NET.211.7]      62 3:19:25.874   24.120.627
02/19/2001 08:37:23 PM TCP: D=27374 S=3451 SYN SEQ=3793991 LEN=0 WIN=8192
    61        [MY.NET.211.7]   [212.252.28.163]     60 3:19:25.892   0.018.201
02/19/2001 08:37:23 PM TCP: D=3451 S=27374 RST ACK=3793992 WIN=0
    62        [212.252.28.163]  [MY.NET.211.7]      62 3:19:26.951   1.058.777
02/19/2001 08:37:24 PM TCP: D=27374 S=3451 SYN SEQ=3793991 LEN=0 WIN=8192
    63        [MY.NET.211.7]   [212.252.28.163]     60 3:19:26.962   0.011.434
02/19/2001 08:37:24 PM TCP: D=3451 S=27374 RST ACK=3793992 WIN=0
    64        [212.252.28.163]  [MY.NET.211.7]      62 3:19:28.017   1.054.695
02/19/2001 08:37:26 PM TCP: D=27374 S=3451 SYN SEQ=3793991 LEN=0 WIN=8192
    65        [MY.NET.211.7]   [212.252.28.163]     60 3:19:28.036   0.018.537
02/19/2001 08:37:26 PM TCP: D=3451 S=27374 RST ACK=3793992 WIN=0
    66        [212.252.28.163]  [MY.NET.211.7]      62 3:19:29.160   1.124.184
02/19/2001 08:37:27 PM TCP: D=27374 S=3451 SYN SEQ=3793991 LEN=0 WIN=8192
```

```
    67        [MY.NET.211.7]   [212.252.28.163]     60 3:19:29.173   0.012.899
02/19/2001 08:37:27 PM TCP: D=3451 S=27374 RST ACK=3793992 WIN=0
    68        [212.252.28.163]  [MY.NET.211.8]      62 3:19:29.833   0.659.867
02/19/2001 08:37:27 PM TCP: D=27374 S=3452 SYN SEQ=3797963 LEN=0 WIN=8192
    69        [212.252.28.163]  [MY.NET.211.8]      62 3:19:32.767   2.934.662
02/19/2001 08:37:30 PM TCP: D=27374 S=3452 SYN SEQ=3797963 LEN=0 WIN=8192
    70 #      [212.252.28.163]  [MY.NET.211.8]      62 3:19:38.757   5.989.783
02/19/2001 08:37:36 PM Expert: Idle Too Long
              TCP: D=27374 S=3452 SYN SEQ=3797963 LEN=0 WIN=8192
    71        [212.252.28.163]  [MY.NET.211.8]      62 3:19:50.846   12.089.080
02/19/2001 08:37:48 PM TCP: D=27374 S=3452 SYN SEQ=3797963 LEN=0 WIN=8192
    72 #      [209.47.152.156]  [MY.NET.210.105]    60 4:00:55.489   2464.642.988
02/19/2001 09:18:53 PM Expert: Idle Too Long ICMP: Echo
```

```
    73        [65.25.212.128]   [MY.NET.219.142]    62 4:01:48.114   52.624.711
02/19/2001 09:19:46 PM TCP: D=27374 S=4632 SYN SEQ=17788039 LEN=0 WIN=8192
    74        [65.25.212.128]   [MY.NET.219.142]    62 4:01:51.077   2.963.476
02/19/2001 09:19:49 PM TCP: D=27374 S=4632 SYN SEQ=17788039 LEN=0 WIN=8192
    75        [65.25.212.128]   [MY.NET.219.142]    62 4:01:57.174   6.096.282
02/19/2001 09:19:55 PM TCP: D=27374 S=4632 SYN SEQ=17788039 LEN=0 WIN=8192
    76        [65.25.212.128]   [MY.NET.219.142]    62 4:02:09.183   12.009.459
02/19/2001 09:20:07 PM TCP: D=27374 S=4632 SYN SEQ=17788039 LEN=0 WIN=8192
```

Here's an expanded view of some of the above frames. MY.NET.211.4 rejects SYN to 27374.

```
- - - - - - - - - - - - - - - - - - - - Frame 39 - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time              Summary
    39        [212.252.28.163]  [MY.NET.211.3]      62 3:18:08.484   11.947.492
02/19/2001 08:36:06 PM TCP: D=27374 S=3409 SYN SEQ=3695526 LEN=0 WIN=8192
ADDR  HEX                                                ASCII
0000: 00 e0 16 7f eb 82 00 00 ef 06 1f 50 08 00 45 00 | ...........P..E.
0010: 00 30 65 22 40 00 6f 06 43 7a d4 fc 1c a3 xx xx | .0e"@.o.Cz...£..
0020: d3 03 0d 51 6a ee 00 38 63 a6 00 00 00 00 70 02 | Ó..Qj..8c.....p.
0030: 20 00 27 72 00 00 02 04 02 18 01 01 04 02       |  .'r..........

- - - - - - - - - - - - - - - - - - - - Frame 40 - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time              Summary
    40        [212.252.28.163]  [MY.NET.211.4]      62 3:18:32.529   24.044.295
02/19/2001 08:36:30 PM TCP: D=27374 S=3424 SYN SEQ=3740681 LEN=0 WIN=8192
ADDR  HEX                                                ASCII
0000: 00 e0 16 7f eb 82 00 00 ef 06 1f 50 08 00 45 00 | ...........P..E.
0010: 00 30 9c 22 40 00 6f 06 0c 79 d4 fc 1c a3 xx xx | .0."@.o..y...£..
0020: d3 04 0d 60 6a ee 00 39 14 09 00 00 00 00 70 02 | Ó..`j..9......p.
0030: 20 00 76 fe 00 00 02 04 02 18 01 01 04 02       |  .vþ..........

- - - - - - - - - - - - - - - - - - - - Frame 41 - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time              Summary
    41        [MY.NET.211.4]   [212.252.28.163]     60 3:18:32.540   0.011.192
02/19/2001 08:36:30 PM TCP: D=3424 S=27374 RST ACK=3740682 WIN=0
ADDR  HEX                                                ASCII
0000: 00 00 ef 06 1f 50 00 e0 16 7f eb 82 08 00 45 00 | .....P........E.
0010: 00 28 44 aa 00 00 7f 06 93 f9 xx xx d3 04 d4 fc | .(D.........Ó...
0020: 1c a3 6a ee 0d 60 00 00 00 00 00 39 14 0a 50 14 | .£j..`.....9..P.
0030: 00 00 c0 12 00 00 20 20 20 20 20 20             | ......

- - - - - - - - - - - - - - - - - - - - Frame 42 - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time              Summary
    42        [212.252.28.163]  [MY.NET.211.4]      62 3:18:33.639   1.098.622
02/19/2001 08:36:31 PM TCP: D=27374 S=3424 SYN SEQ=3740681 LEN=0 WIN=8192
ADDR  HEX                                                ASCII
0000: 00 e0 16 7f eb 82 00 00 ef 06 1f 50 08 00 45 00 | ...........P..E.
```

```
0010: 00 30 9d 22 40 00 6f 06 0b 79 d4 fc 1c a3 xx xx | .0 "@.o..y...£..
0020: d3 04 0d 60 6a ee 00 39 14 09 00 00 00 00 70 02 | Ó..`j..9......p.
0030: 20 00 76 fe 00 00 02 04 02 18 01 01 04 02       |  .vþ..........

- - - - - - - - - - - - - - - - - - - - Frame 43 - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time             Summary
    43        [MY.NET.211.4]   [212.252.28.163]     60 3:18:33.652   0.012.978
02/19/2001 08:36:31 PM TCP: D=3424 S=27374 RST ACK=3740682 WIN=0
ADDR  HEX                                                  ASCII
0000: 00 00 ef 06 1f 50 00 e0 16 7f eb 82 08 00 45 00 | .....P........E.
0010: 00 28 46 aa 00 00 7f 06 91 f9 xx xx d3 04 d4 fc | .(F.........Ó...
0020: 1c a3 6a ee 0d 60 00 00 00 00 00 39 14 0a 50 14 | .£j..`.....9..P.
0030: 00 00 c0 12 00 00 20 20 20 20 20 20             | ......
```

Key to Firewall Log
[Date and Time][Action][Protocol:Type][src addr][dest addr]

```
Log Excerpts:

(from a different day)
Do not have enough information to know if this was some sort of
response ... it was caught as an illegal source address egress attempt
... happens to also have the 27374 port address on 3 frames.
fw# grep 11020 /var/log/security
Feb 11 08:31:57 fw /kernel: ipfw: 11020 Deny TCP 172.139.121.34:27374
213.130.11.133:62236 in via xl0
Feb 11 08:31:57 fw /kernel: ipfw: 11020 Deny TCP 172.139.121.34:27374
213.130.11.133:62236 in via xl0
Feb 11 09:54:29 fw /kernel: ipfw: 11020 Deny TCP 172.132.14.217:27374
213.122.242.119:4596 in via xl0
Feb 11 11:39:30 fw /kernel: ipfw: 11020 Deny ICMP:3.3 172.132.52.69
62.163.74.127 in via xl0
fw#
```

1. Source of Traces:

Where: Mynet.Edu

2. Detect was generated by:

Curiosity. What would happen if I just looked for activity on port 27374? The Sniffer® was set up outside the campus firewall with capture filter set to capture source or destination UDP/TCP port 27374 (0x6AEE)(Ethernet packet offset 0x22 or 0x24).

3. Probability the source address was spoofed:

In the case of the dns frames, absolutely. This was part of a **dns dos** attack against the alleged source address dns server.

In the case of the last set of TCP packets, from the firewall logs, definitely spoofed. They were logged by mynet.edu's egress filter firewall rule. As for the other TCP's, of the 27374 scan, There's only one source address rather than randomly changing source addresses, and so I'm going to guess that it is not spoofed.

The pings may be entirely incidental in the context of searching for 27374. Because I was searching for only 27374 at a particular offset in the header, I have no knowledge of the context in which these frames occurred, as far as ongoing pings that might have been present. Since the inter-arrival times of the captured pings are quite large (except for request/reply pairs), I am going to assume that these are spurious captures, whose checksums just happen to match the pattern and offset for which I was filtering, and that there is no indication of malicious activity here. In a later test, in fact, I set up the Sniffer to capture ICMP echo and response traffic only, with no other conditions. Out of over 196,000 frames, 13 of them had the checksum 27374.

Note the "delta time", below.

Sniffer Network Analyzer data from 19-Feb-2001 at 17:15:34, file C:\ENCAP\27375.ENC, Page 1

[frame number][delta t][dest ip][src ip][proto or service][summary]

SUMMARY	Delta T	Destination	Source	Summary
M 1		host.mynet.edu	[194.204.49.250]	DNS C ID=58535 OP=QUERY NAME=aol.com
2	1418.0892	[MY.NET.210.138]	[213.46.146.79]	ICMP Echo
3	0.8783	[213.46.146.79]	[MY.NET.210.138]	ICMP Echo reply
4	557.7088	[216.161.137.231]	[MY.NET.173.163]	ICMP Echo reply
5	182.5917	[MY.NET.212.25]	[24.241.105.85]	ICMP Echo
6	1403.6064	[MY.NET.211.161]	[172.164.170.80]	ICMP Echo
7	6.7880	[172.164.170.80]	[MY.NET.211.161]	ICMP Echo reply
8	155.5364	[MY.NET.211.161]	[216.70.132.161]	ICMP Echo
9	311.5116	[MY.NET.174.98]	[217.50.155.95]	ICMP Echo
10	6.8272	[217.50.155.95]	[MY.NET.174.98]	ICMP Echo reply
11	177.0861	[193.153.231.75]	[MY.NET.211.147]	ICMP Echo reply
12	1228.2510	[24.200.31.81]	[MY.NET.73.240]	ICMP Echo reply
13	885.3538	host.mynet.edu	[194.204.49.250]	DNS C ID=22296 OP=QUERY NAME=aol.com
14	61.5559	host.mynet.edu	[194.204.49.250]	DNS C ID=19483 OP=QUERY NAME=aol.com
15	423.0099	host.mynet.edu	[194.204.49.254]	DNS C ID=36367 OP=QUERY NAME=aol.com
16	105.0127	[MY.NET.211.157]	[208.6.8.102]	ICMP Echo
17	0.2159	[208.6.8.102]	[MY.NET.211.157]	ICMP Echo reply
18	213.3465	[MY.NET.209.217]	[144.80.169.23]	ICMP Echo
19	1.5129	[144.80.169.23]	[MY.NET.209.217]	ICMP Echo reply
20	572.4076	[MY.NET.212.25]	[24.188.71.27]	ICMP Echo

4. Description of attack:

When

Sniffer trace taken from 2/19/01 17:17:58 EST to 2/20/01 8:35:57 EST. (UTC -5:00) The firewall trace came from Feb. 11.

Who: (pings)
(no two source addresses alike)
A sample selection follows of our pings to the source addresses using Windows> ping -a to get name resolution.

Pinging mu-250116.dhcp.missouri.edu [128.206.250.116]
Reply from 128.206.250.116: bytes=32 time=716ms TTL=110

Pinging slip139-92-226-176.hul.be.prserv.net [139.92.226.176]
Request timed out. (This was a destination of a port 137 unreachable)

Pinging adsl-141-150-142-136.nnj.adsl.bellatlantic.net [141.150.142.136]
Reply from 141.150.4.77: Destination host unreachable.

Pinging utw2a2s10.resnet.iup.edu [144.80.169.23]
Request timed out. (13 hops to destination unreachable (from MyHomeIsp))

- snipped for brevity - They appear to be dialups or edu's (residential).

What:

Captured some of those dns queries for AOL/Mailrelay, with source port UDP 27374. [Reference: "Handler Comments" in http://www.sans.org/y2k/021001.htm]
These are typical of 17 frames captured; replies were not sent as these dns request packets were being denied at the firewall; these were included in the Sniffer trace due to 0x6aee at offset 0x22. Rather long "Delta Time" times suggest the capture of these port numbers could be a random effect of ephemeral source port assignment.

An accidental side effect of searching for particular udp/tcp port number was that I captured series of pings with icmp header checksums all set to 0x6AEE (27374). Again it should be double checked whether this was a result of collecting data over a long enough time period that these numbers just happened to show up from time to time, as I currently believe to be the case. I also noticed ICMP echo frames with bytes that could have been data extending out beyond the end of the "total length" value from the IP header. I found this curious, but, again, in observing ICMP echo traffic subsequently, I have noticed this to occur fairly often. It can be a sign of an exploit, but in the absence of a multitude of responses being triggered by a single request, I would not assume it in this case.

Frame	Status	Source Address	Dest. Address	Size	Rel. Time	Delta Time	Abs. Time	Summary

```
     1 M      [MY.NET.173.182] [152.19.210.81]      60 0:00:00.000   0.000.000
02/20/2001 03:12:37 AM ICMP: Echo
ADDR  HEX                                                 ASCII
0000: 00 00 ef 06 1f 50 00 e0 16 7f eb 82 08 00 45 00 | .....P........E.
0010: 00 1c c7 a2 00 00 7e 01 be 9a xx xx ad b6 98 13 | ..Ç...~.........
0020: d2 51 08 00 6a ee 03 00 8a 11 92 8c 66 ca 02 00 | .Q..j.......f...
0030: 00 00 00 00 00 00 00 00 00 00 00 00             | ............
```

The total length field in the above ip header is 0x1c = 28; counting from byte #0xE the dec28th ip byte is the 0x29th byte in the whole hex dump, the one equal to 0x11. Following that there are 5 more bytes of data, followed by padding, to yield the required ethernet frame size of 60 (64, less the four ethernet checksum bytes that are not part of the hexdump.)

Captured some TCP SYNs to port 27374, some unanswered, others answered with RES/ACK. This could well be a sub-seven scan, which is the reason for the interest in this port.

Stations that responded with a RST-ACK:

MY.NET.211.4
MY.NET.211.5
MY.NET.211.7
MY.NET.74.82

All RST-ACKs showed
-IP total length field=48
All SYNs showed
-SYN options 02 04 02 18 01 01 04 02

5. Attack mechanism:

PINGS

- There are stimulus/response pairs both with the noted checksum (How strange is that? I have since seen other examples of the checksum remaining the same between request and reply.)
- There are some replies to stations that did not appear in this trace as a stimulus ... different checksum? But the reply has the noted checksum.
- There are some requests from local stations for which no reply comes (different checksum? or just no reply?).
- No off campus address shows up for more than one transmit and/or receive frame. All show with the IP header total length field of 28, but some have data, not just 00's or 20's beyond that point.
- Incoming TTL's mostly in the range 111-121 (one at 21).
- Many minutes between pings.
- Destination addresses are all plausible Mynet.Edu addresses.

Are we seeing what is noted in SANS New Orleans 2001 textbook for class 3.2, in the Appendix, line 27, "echo request and reply packet payloads do not match in a request response pair." I have also seen this elsewhere, and so would have to look for more correlation to see this as a definite problem. See frames 1 and 2 from 2/20/2001:

```
- - - - - - - - - - - - - - - - - - - - - - Frame 1 - - - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time              Summary
     1 M      [172.146.225.142] [MY.NET.174.171]    60 0:00:00.000   0.000.000
02/20/2001 06:07:10 AM ICMP: Echo
```

```
ADDR  HEX                                                  ASCII
0000: 00 e0 16 7f eb 82 00 00 ef 06 1f 50 08 00 45 00 | ...........P..E.
0010: 00 1c 1d 9e 00 00 74 01 4d ee ac 92 e1 8e 9e 88 | ......t.M...á...
0020: ae ab 08 00 6a ee 02 00 8b 11 00 00 00 00 00 00 | ....j...........
0030: 00 00 00 00 00 00 00 00 00 00 00 00             | ............

- - - - - - - - - - - - - - - - - - - - Frame 2 - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time     Abs.
Time             Summary
     2        [MY.NET.174.171] [172.146.225.142]    60 0:00:00.058   0.058.598
02/20/2001 06:07:10 AM ICMP: Echo reply
ADDR  HEX                                                  ASCII
0000: 00 00 ef 06 1f 50 00 e0 16 7f eb 82 08 00 45 00 | .....P........E.
0010: 00 1c fa f2 00 00 7e 01 66 99 9e 88 ae ab ac 92 | ......~.f.......
0020: e1 8e 00 00 6a ee 02 00 93 11 54 bd 34 ec 02 00 | á...j.....T.4...
0030: 00 00 00 00 00 00 00 00 00 00 00 00             | ............
```

6. Correlations:

This exercise was an attempt to "correlate", or learn more about, previous firewall logs which showed activity on 27374.

These firewall log entries are from Feb 19, 2001. Here are some more illegal outgoing addresses, this time with 27374 as the source address.

```
fw# grep 27374 security.219*
security.219-1900:Feb 19 13:40:34 fw /kernel: ipfw: 14100 Count TCP
195.121.20.212:1886 MY.NET.173.3:27374 in via xl1
security.219-1900:Feb 19 15:37:06 fw /kernel: ipfw: 11020 Deny TCP
172.129.204.190:27374 204.60.42.181:3919 in via xl0
security.219-1900:Feb 19 15:49:06 fw /kernel: ipfw: 11020 Deny TCP
172.140.180.44:27374 212.204.142.247:2978 in via xl0
security.219-1900:Feb 19 15:58:36 fw /kernel: ipfw: 11020 Deny TCP
172.152.69.43:27374 62.100.41.90:2530 in via xl0
security.219-1900:Feb 19 17:23:37 fw /kernel: ipfw: 11020 Deny TCP
172.132.96.191:27374 209.246.98.134:2495 in via xl0
security.219-2300:Feb 19 19:37:40 fw /kernel: ipfw: 11020 Deny TCP
172.142.3.179:27374 63.210.76.46:2937 in via xl0
security.219-2300:Feb 19 20:33:10 fw /kernel: ipfw: 14020 Deny TCP
212.252.28.163:3368 MY.NET.211.1:27374 in via xl1
security.219-2300:Feb 19 20:33:10 fw /kernel: ipfw: 14020 Deny TCP
212.252.28.163:3368 MY.NET.211.1:27374 in via xl1
security.219-2300:Feb 19 20:34:10 fw /kernel: ipfw: 14100 Count TCP
212.252.28.163:3385 MY.NET.211.2:27374 in via xl1
security.219-2300:Feb 19 20:34:10 fw /kernel: ipfw: 14100 Count TCP
212.252.28.163:3385 MY.NET.211.2:27374 in via xl1
security.219-2300:Feb 19 20:34:40 fw /kernel: ipfw: 14100 Count TCP
212.252.28.163:3409 MY.NET.211.3:27374 in via xl1
security.219-2300:Feb 19 20:35:00 fw /kernel: ipfw: 14100 Count TCP
212.252.28.163:3409 MY.NET.211.3:27374 in via xl1
security.219-2300:Feb 19 20:35:40 fw /kernel: ipfw: 14100 Count TCP
212.252.28.163:3424 MY.NET.211.4:27374 in via xl1
security.219-2300:Feb 19 20:35:40 fw /kernel: ipfw: 14100 Count TCP
212.252.28.163:3429 MY.NET.211.5:27374 in via xl1
security.219-2300:Feb 19 20:35:40 fw /kernel: ipfw: 14100 Count TCP
212.252.28.163:3434 MY.NET.211.6:27374 in via xl1
security.219-2300:Feb 19 20:36:10 fw /kernel: ipfw: 14100 Count TCP
212.252.28.163:3434 MY.NET.211.6:27374 in via xl1
```

```
security.219-2300:Feb 19 20:36:40 fw /kernel: ipfw: 14100 Count TCP
212.252.28.163:3451 MY.NET.211.7:27374 in via xl1
security.219-2300:Feb 19 20:36:40 fw /kernel: ipfw: 14100 Count TCP
212.252.28.163:3452 MY.NET.211.8:27374 in via xl1
security.219-2300:Feb 19 20:36:40 fw /kernel: ipfw: 14100 Count TCP
212.252.28.163:3452 MY.NET.211.8:27374 in via xl1
security.219-2300:Feb 19 20:36:41 fw /kernel: ipfw: 14100 Count TCP
212.252.28.163:3452 MY.NET.211.8:27374 in via xl1
security.219-2300:Feb 19 21:18:41 fw /kernel: ipfw: 14100 Count TCP
65.25.212.128:4632 MY.NET.219.142:27374 in via xl1
security.219-2300:Feb 19 21:18:41 fw /kernel: ipfw: 14100 Count TCP
65.25.212.128:4632 MY.NET.219.142:27374 in via xl1
security.219-2300:Feb 19 21:19:11 fw /kernel: ipfw: 14100 Count TCP
65.25.212.128:4632 MY.NET.219.142:27374 in via xl1
security.219-2300:Feb 19 21:19:11 fw /kernel: ipfw: 14100 Count TCP
65.25.212.128:4632 MY.NET.219.142:27374 in via xl1
security.219-2300:Feb 19 22:03:12 fw /kernel: ipfw: 14100 Count TCP
211.221.132.48:2278 MY.NET.74.82:27374 in via xl1
```

Correlation from others regarding the 27374 scans:

```
from giac
http://www.sans.org/y2k/010901.htm
Greetings, Over the past week we have detected a slow scan of ports
137, 139, 12345, 27374 in
 part of our network. This scan appears to have originated from
211.61.86.222.

 portscan log file for Snort IDS.

 ODD Packet - SubSeven Trojan = port 27374

 Jan  3 01:25:40 24.27.132.159:2809 -> a.b.c.1:27374 SYN **S*****
 Jan  3 01:25:40 24.27.132.159:2811 -> a.b.c.3:27374 SYN **S*****
```

http://www.sans.org/y2k/010901-1300.htm

```
Dec 29 15:51:12 226 IP  packet dropped (211.61.86.222->204.167.28.132:
 Protocol=TCP[SYN] Port 2071->27374)
Search the APNIC Whois database
 Search results for '211.61.86.222'
 inetnum              211.52.0.0 - 211.63.255.255
 netname              KRNIC-KR-24
 descr                Korea Network Information Center
 descr                14F, NARA Bldg, 1328-3, Seocho-Dong, Seocho-Ku
 descr                Seoul, Korea, 137-070
 country              KR
 e-mail               hostmaster@nic.or.kr, inverse
 nic-hdl              WK1-AP, inverse
 mnt-by               MNT-KRNIC-AP, inverse
 changed              hostmaster@nic.or.kr 20000927
 source               APNIC
```

http://www.sans.org/y2k/011001.htm

```
Here is a list of all my SubSeven 2.1 probes against my cable modem for
the month of December
 2000. You will also not some repeat offender.

 Dec  1 01:15:45 mybox kernel: Packet log: inp DENY eth0 PROTO=6
   4.33.36.39:1075 192.168.30.1:27374 L=48 S=0x00 I=160 F=0x4000 T=115
SYN (#54)
 Dec  1 06:57:50 mybox snort[23339]: IDS279 - BACKDOORATTEMPT-Subseven
   v2.1: 24.12.19.185:2845 -> 192.168.30.1:27374
 Dec  1 14:57:18 mybox snort[23339]: IDS279 - BACKDOOR ATTEMPT-Subseven
   v2.1: 216.224.148.38:3292 -> 192.168.30.1:27374... more
```

http://www.sans.org/y2k/011501.htm

```
from broad band connections: Sub seven search detect early yesterday (EST)
 from an NYC segment from user:
 24-168-83-212.nyc.rr.com [24.168.83.212]
 Source Address    Dest. Address      Size   Abs. Time              Summary
 [24.168.83.212]   [my.box.nat.ip]      60  01/05/2001 05:09:24 AM DLC:
 Ethertype=0800, size=60 bytes
 IP:  D=[my.box.nat.ip] S=[24.168.83.212] LEN=24 ID=7209
 TCP: D=27374 S=4573 SYN SEQ=25679698 LEN=0 WIN=5840
 ADDR  HEX                                                ASCII
 0000: 00 60 8c bb ec e4 00 20 78 c5 b3 74 08 00 45 00 | .-...U...E......
 0010: 00 2c 1c 29 40 00 19 06 4b 77 18 a8 53 d4 c0 a8 | .... ......y.M{y
 0020: cd 07 11 dd 6a ee 01 87 d7 52 00 00 00 00 60 02 | ....]..gP.....-.
 0030: 16 d0 31 85 00 00 02 04 05 b4 00 00             | .}.e
```

http://www.sans.org/y2k/011501-1500.htm

```
Jan  5 03:58:03 socretes kernel: Packet log: input DENY eth1 PROTO=6
   209.212.169.8:10101 x.x.x.x:27374 L=40 S=0x00 I=51124 F=0x0000 T=242
SYN
 Jan  5 09:38:35 socretes kernel: Packet log: input DENY eth1 PROTO=6
   209.212.169.8:10101 x.x.x.x:27374 L=40 S=0x00 I=20650 F=0x0000 T=242
SYN
 Jan  7 00:16:45 socretes kernel: Packet log: input DENY eth1 PROTO=6
```

http://www.sans.org/y2k/011601.htm

"Seems like people are not giving up on the NetBus and Sub-7 scans."

```
 Jan 03 08:28:46(EST) : 12.27.42.100->204.167.30.58: Protocol=TCP[SYN]
   Port 1554->12345
 Jan 03 08:28:46(EST) : 12.27.42.100->204.167.30.58: Protocol=TCP[SYN]
   Port 1553->27374
 Jan 03 08:28:49(EST) : 12.27.42.100->204.167.30.58: Protocol=TCP[SYN]
   Port 1554->12345
 Jan 03 08:28:49(EST) : 12.27.42.100->204.167.30.58: Protocol=TCP[SYN]
   Port 1553->27374
 Jan 03 08:28:55(EST) : 12.27.42.100->204.167.30.58: Protocol=TCP[SYN]
   Port 1554->12345
 Jan 03 08:28:55(EST) : 12.27.42.100->204.167.30.58: Protocol=TCP[SYN]
   Port 1553->27374
```

http://www.sans.org/y2k/011801-1330.htm

```
(Guy Bruneau)

 Here are my statistics from my cable modem for year 2000. I have
listed the top five of both the
 Trojans and services. Top five Trojans: Complete statistics for year
2000 available at:
 http://members.home.com/gbruneau1/port_scan2000.htm

 1 - SubSeven 2.1   (TCP 27374)
 2 - NetBus         (TCP 12345)
 3 - SubSeven 1.9   (TCP 1243)
 4 - DeepThroat 3.1 (UDP 2140)
 5 - Hack'a'Tack    (UDP 31789) and
Unknown        (UDP 5154)

 Top five services:
 1 - Web HTTP (TCP 80)
 2 - RPC      (TCP 111)
 3 - FTP      (TCP 21)
 4 - Telnet   (TCP 23)
 5 - SNMP     (UDP 161)
```

7. Evidence of active targeting:

Incoming requests for the most part are achieving a response, whether echo returns or RST/ACKS in the case of some of the TCP scan packets. The scan I feel is not specifically targeted, as it is likely to be going to other sites as well as ours.

I have come to believe that the presence of the pattern 0x6aee in the dns and icmp packets is of no relevance. However, since the packets were captured, I'll comment on them.

The dns packets are actively targeted, though not only at us, because they are part of a dns ddos which depends upon eliciting a response from a dns server. They can't be too actively targeted, in the sense that we were blocking them from reaching our dns server, and so they had ceased to elicit the desired responses.

The pings most likely are actively targeted, as someone was trying to find out the responsiveness of hosts on our campus. Virtually all of the MY.NET addresses are within our residential subnets, where the heaviest use of Napster takes place. I know from previous work with the Napster software that one of a Napster client's methods of choosing among possible download hosts is to send pings to them, to determine the response time of the network path.

8. Severity:

Severity is moderate. TCP portion of this looks like subseven remote access.

(vulnerability + criticality) - (system countermeasures + network countermeasures) = severity

Vulnerability is high, but criticality is low due to addresses in this trace belonging not to servers but to individual hosts (however these hosts could be used to gain passwords to campus servers, and so this traffic should be watched.) System countermeasures are unknown, since most are privately owned. Network countermeasures for this pattern are not in place.

(4 + 2) - (2 + 1) = 3

9. Defensive recommendation:

Implement measures to help distinguish between attack signatures and legitimate ephemeral port numbers, so that real connections are not broken and so that real threats can be looked into.

Make an examination of the ping packets with unequal payloads. Closely monitor those hosts.

Consider implementing a stateful firewall so that more granular protections can be implemented on our perimeter connections.

10. Multiple choice test question:

What is the significance of a "stateful" IDS or firewall?:

A. It's state is "enabled".

B. A connection is established between the IDS and the attacking host.

C. It looks at packets one at a time.

D. It takes into account a packet's relationship to other traffic between the hosts participating in a packet exchange.

Answer: D

DETECT #2: OPPORTUNISTIC HONEYPOT

Mynet.edu has a few subnets that are exclusively used for internal routing or network management. All outside traffic to these subnets is prohibited at the firewall. These rules regularly take hits, and log denials. This is an opportunity to see what internet background noise looks like.

Sniffer Summary Information:
[Frame Number][SrcIPAddr][DstIPAddr][Size][Elapsed Time since start][Time since previous frame][clock time][contents summary]

Frame	Status	Source Address	Dest. Address	Size	Rel. Time	Delta Time	Abs. Time	Summary
1	M	[205.171.4.70]	[MY.NET.CCC.21]	70	0:00:00.000	0.000.000	02/13/1992 10:51:45 AM	ICMP: Destination unreachable (Host unreachable)
2		[193.231.238.137]	[MY.NET.AAA.202]	60	0:01:01.312	61.312.620	02/13/1992 10:52:46 AM	TCP: D=32680 S=226 RST ACK=1371415177 WIN=0
3	#	[195.158.225.81]	[MY.NET.AAA.202]	70	0:01:31.023	29.710.434	02/13/1992 10:53:16 AM	Expert: ICMP Host Unreachable ICMP: Destination unreachable (Host unreachable)
4	#	[193.231.227.181]	[MY.NET.AAA.202]	82	0:13:21.439	710.416.153	02/13/1992 11:05:06 AM	Expert: Time-to-live exceeded in transmit ICMP: Time exceeded (Time to live exceeded in transit)
5	#	[193.231.227.181]	[MY.NET.AAA.202]	82	0:13:26.265	4.826.199	02/13/1992 11:05:11 AM	Expert: Time-to-live exceeded in transmit ICMP: Time exceeded (Time to live exceeded in transit)
6	#	[193.231.227.181]	[MY.NET.AAA.202]	82	0:13:26.285	0.020.036	02/13/1992 11:05:11 AM	Expert: Time-to-live exceeded in transmit ICMP: Time exceeded (Time to live exceeded in transit)
7	#	[193.231.227.181]	[MY.NET.AAA.202]	82	0:13:34.036	7.750.755	02/13/1992 11:05:19 AM	Expert: Time-to-live exceeded in transmit ICMP: Time exceeded (Time to live exceeded in transit)
8	#	[193.231.227.181]	[MY.NET.AAA.202]	82	0:13:49.171	15.135.514	02/13/1992 11:05:34 AM	Expert: Time-to-live exceeded in transmit ICMP: Time exceeded (Time to live exceeded in transit)
9	#	[213.174.71.2]	[MY.NET.AAA.202]	70	0:14:49.706	60.535.009	02/13/1992 11:06:34 AM	Expert: ICMP Host Unreachable ICMP: Destination unreachable (Host unreachable)
10	#	[213.174.71.2]	[MY.NET.AAA.202]	70	0:14:49.884	0.177.654	02/13/1992 11:06:34 AM	Expert: ICMP Host Unreachable ICMP: Destination unreachable (Host unreachable)
11	#	[213.174.71.2]	[MY.NET.AAA.202]	70	0:14:55.391	5.506.721	02/13/1992 11:06:40 AM	Expert: ICMP Host Unreachable ICMP: Destination unreachable (Host unreachable)
12	#	[205.171.4.70]	[MY.NET.CCC.112]	70	0:25:04.740	609.349.834	02/13/1992 11:16:49 AM	Expert: ICMP Host Unreachable ICMP: Destination unreachable (Host unreachable)
13	#	[205.171.4.70]	[MY.NET.BBB.68]	70	0:27:26.975	142.234.264	02/13/1992 11:19:11 AM	Expert: ICMP Host Unreachable ICMP: Destination unreachable (Host unreachable)
14	#	[207.96.164.25]	[MY.NET.AAA.202]	70	0:28:30.271	63.296.756	02/13/1992 11:20:15 AM	Expert: Time-to-live exceeded in transmit ICMP: Time exceeded (Time to live exceeded in transit)
15		[157.22.160.100]	[MY.NET.CCC.99]	92	0:33:35.225	304.953.557	02/13/1992 11:25:20 AM	WINS: C ID=2530 OP=QUERY NAME=*<00000000000000000000000000><00>
16	#	[157.22.160.100]	[MY.NET.CCC.99]	92	0:33:36.632	1.407.243	02/13/1992 11:25:21 AM	Expert: WINS No Response WINS: C ID=2536 OP=QUERY NAME=*<00000000000000000000000000><00>

```
  17 #     [202.178.243.254] [MY.NET.CCC.0]      70 0:39:57.264   380.631.352   02/13/1992 11:31:42 AM Expert: Time-to-live
exceeded in transmit
                                                    ICMP: Time exceeded (Time to live exceeded in transit)
  18 #     [202.178.243.254] [MY.NET.CCC.0]      70 0:42:23.764   146.500.041   02/13/1992 11:34:08 AM Expert: Time-to-live
exceeded in transmit
                                                    ICMP: Time exceeded (Time to live exceeded in transit)
  19 #     [205.171.4.70]    [MY.NET.BBB.125]    70 0:42:24.732   0.968.581     02/13/1992 11:34:09 AM Expert: ICMP Host
Unreachable
                                                    ICMP: Destination unreachable (Host unreachable)
  20 #     [158.123.219.96]  [MY.NET.CCC.121]    92 0:54:15.578   710.845.603   02/13/1992 11:46:00 AM Expert: WINS No
Response
                                                    WINS: C ID=13856 OP=QUERY
NAME=*<00000000000000000000000000><00>
  21       [158.123.219.96]  [MY.NET.CCC.121]    92 0:54:17.079   1.500.962     02/13/1992 11:46:02 AM WINS: C ID=13858
OP=QUERY NAME=*<00000000000000000000000000><00>
  22 #     [158.123.219.96]  [MY.NET.CCC.121]    92 0:54:18.576   1.497.392     02/13/1992 11:46:03 AM Expert: WINS No
Response
                                                    WINS: C ID=13860 OP=QUERY
NAME=*<00000000000000000000000000><00>
  23 #     [212.163.65.75]   [MY.NET.AAA.103]    92 0:59:30.247   311.671.163   02/13/1992 11:51:15 AM Expert: WINS No
Response
                                                    WINS: C ID=2324 OP=QUERY
NAME=*<00000000000000000000000000><00>
  24       [212.163.65.75]   [MY.NET.AAA.103]    92 0:59:31.855   1.607.744     02/13/1992 11:51:16 AM WINS: C ID=2326
OP=QUERY NAME=*<00000000000000000000000000><00>
  25 #     [212.163.65.75]   [MY.NET.AAA.103]    92 0:59:33.320   1.465.048     02/13/1992 11:51:18 AM Expert: WINS No
Response
                                                    WINS: C ID=2328 OP=QUERY
NAME=*<00000000000000000000000000><00>
  26 #     [205.171.4.70]    [MY.NET.BBB.107]    70 1:01:30.380   117.059.855   02/13/1992 11:53:15 AM Expert: ICMP Host
Unreachable
                                                    ICMP: Destination unreachable (Host unreachable)
  27       [200.199.245.2]   [MY.NET.AAA.93]     60 1:04:58.953   208.572.539   02/13/1992 11:56:43 AM TCP: D=17952 S=57226
RST WIN=0
  28 #     [146.188.12.161]  [MY.NET.AAA.202]    70 1:31:26.400   1587.447.581  02/13/1992 12:23:11 PM Expert: Time-to-live
exceeded in transmit
                                                    ICMP: Time exceeded (Time to live exceeded in transit)
  29 #     [206.111.14.194]  [MY.NET.AAA.202]    70 1:33:32.514   126.114.156   02/13/1992 12:25:17 PM Expert: ICMP Host
Unreachable
                                                    ICMP: Destination unreachable (Host unreachable)
  30 #     [202.178.243.254] [MY.NET.AAA.0]      70 1:41:44.442   491.928.102   02/13/1992 12:33:29 PM Expert: Time-to-live
exceeded in transmit
                                                    ICMP: Time exceeded (Time to live exceeded in transit)
  31       [200.199.245.2]   [MY.NET.AAA.24]     60 1:45:38.359   233.917.114   02/13/1992 12:37:23 PM TCP: D=18491 S=53139
RST ACK=1 WIN=0
  32 #     [202.178.243.254] [MY.NET.AAA.0]      70 1:48:24.177   165.817.267   02/13/1992 12:40:09 PM Expert: Time-to-live
exceeded in transmit
                                                    ICMP: Time exceeded (Time to live exceeded in transit)
  33 #     [61.132.23.66]    [MY.NET.BBB.0]      70 1:48:56.370   32.192.833    02/13/1992 12:40:41 PM Expert: Time-to-live
exceeded in transmit
                                                    ICMP: Time exceeded (Time to live exceeded in transit)
  34 #     [202.178.243.254] [MY.NET.AAA.0]      70 1:51:34.388   158.018.269   02/13/1992 12:43:19 PM Expert: Time-to-live
exceeded in transmit
                                                    ICMP: Time exceeded (Time to live exceeded in transit)
  35 #     [62.128.1.6]      [MY.NET.AAA.202]    70 1:56:32.244   297.855.771   02/13/1992 12:48:17 PM Expert: Time-to-live
exceeded in transmit
                                                    ICMP: Time exceeded (Time to live exceeded in transit)
  36       [200.199.245.2]   [MY.NET.CCC.121]    60 2:06:16.121   583.877.514   02/13/1992 12:58:01 PM TCP: D=1881 S=64958
RST ACK=1 WIN=0
  37 #     [61.132.74.1]     [MY.NET.AAA.0]      70 2:15:30.543   554.422.147   02/13/1992 01:07:15 PM Expert: Time-to-live
exceeded in transmit
                                                    ICMP: Time exceeded (Time to live exceeded in transit)
```

38 # [61.132.74.1] [MY.NET.BBB.0] 70 2:40:03.793 1473.249.509 02/13/1992 01:31:48 PM Expert: Time-to-live exceeded in transmit
ICMP: Time exceeded (Time to live exceeded in transit)
39 # [134.222.199.22] [MY.NET.AAA.202] 70 3:14:44.396 2080.602.962 02/13/1992 02:06:29 PM Expert: Time-to-live exceeded in transmit
ICMP: Time exceeded (Time to live exceeded in transit)
40 # [202.178.243.254] [MY.NET.AAA.0] 70 3:14:55.988 11.591.887 02/13/1992 02:06:40 PM Expert: Time-to-live exceeded in transmit
ICMP: Time exceeded (Time to live exceeded in transit)
41 # [205.171.4.70] [MY.NET.BBB.45] 70 3:38:41.223 1425.235.089 02/13/1992 02:30:26 PM Expert: ICMP Host Unreachable
ICMP: Destination unreachable (Host unreachable)
42 # [205.171.4.70] [MY.NET.CCC.93] 70 3:47:13.439 512.215.947 02/13/1992 02:38:58 PM Expert: ICMP Host Unreachable
ICMP: Destination unreachable (Host unreachable)

Frame	Status	Source Address	Dest. Address	Size	Rel. Time	Delta Time	Abs. Time	Summary
1216	#	[193.140.188.188]	[MY.NET.CCC.103]	590	4:04:10.961	18.888.972	02/13/1992 02:55:55 PM	Expert: ICMP Destination Unreachable
								ICMP: Destination unreachable (Protocol unreachable)
1217	#	[202.178.243.254]	[MY.NET.BBB.0]	70	4:05:51.264	100.302.645	02/13/1992 02:57:36 PM	Expert: Time-to-live exceeded in transmit
								ICMP: Time exceeded (Time to live exceeded in transit)
1218	#	[205.171.4.70]	[MY.NET.BBB.96]	70	4:11:08.481	317.217.520	02/13/1992 03:02:53 PM	Expert: ICMP Host Unreachable
								ICMP: Destination unreachable (Host unreachable)
1219	#	[202.178.243.254]	[MY.NET.AAA.0]	70	4:12:01.079	52.597.467	02/13/1992 03:03:46 PM	Expert: Time-to-live exceeded in transmit
								ICMP: Time exceeded (Time to live exceeded in transit)
1220	#	[202.178.243.254]	[MY.NET.CCC.0]	70	4:13:29.079	88.000.136	02/13/1992 03:05:14 PM	Expert: Time-to-live exceeded in transmit
								ICMP: Time exceeded (Time to live exceeded in transit)
1221	#	[202.178.243.254]	[MY.NET.BBB.0]	70	4:14:18.472	49.393.437	02/13/1992 03:06:03 PM	Expert: Time-to-live exceeded in transmit
								ICMP: Time exceeded (Time to live exceeded in transit)
1222		[63.144.122.149]	[MY.NET.AAA.202]	60	4:17:41.587	203.114.317	02/13/1992 03:09:26 PM	TCP: D=6568 S=6225 RST ACK=1048286345 WIN=0
1223		[24.64.46.13]	[MY.NET.AAA.143]	60	4:31:44.349	842.762.480	02/13/1992 03:23:29 PM	TCP: D=40021 S=17920 RST ACK=237085914 WIN=0

Sniffer Trace

Frame - Frame number in trace.
Status - Whether the packet is Marked as a reference point in the trace.
Source Address - From the IP header
Destination Address - From the IP header
Size - total length of the captured packet. The Sniffer doesn't capture the ethernet trailing checksum.
Rel. Time - The elapsed time since the trace began or from a set marker.
Delta Time - The time since the previous frame.
Abs. Time - The date and time-of-day the packet was received by the Sniffer®.
Summary - The Sniffer®'s brief description of the packet, containing any messages generated by the Sniffer "Expert" followed by a summary of the highest level header information the Sniffer was able to decode from the packet.

HEX section - the ADDR column is simply for ease of reading the HEX information, showing the HEX representation of the offset location of the first byte in each row.
- the HEX section shows the hexadecimal representation of each byte in the packet.
- the ASCII section shows the ASCII translation of each byte that maps to the ASCII code.
These traces include the ethernet headers.

```
- - - - - - - - - - - - - - - - - - - - - Frame 1 - - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time              Summary
     1 M      [193.231.238.137] [my.domain.aaa.202]    60 0:00:00.000   0.000.000
02/13/2001 10:52:46 AM TCP: D=32680 S=226 RST ACK=1371415177 WIN=0
ADDR  HEX                                                    ASCII
0000: 00 e0 16 7f eb 82 00 00 ef 06 1f 50 08 00 45 00 |
...........P..E.
0010: 00 28 f7 2f 00 00 e5 06 16 dc c1 e7 ee 89 XX XX |
.(./......Á.....
0020: XX ca 00 e2 7f a8 00 00 00 00 51 be 22 89 50 14 |
x..â......Q.".P.
0030: 00 00 f3 3a 00 00 00 00 00 00 00 00             | ...:........

- - - - - - - - - - - - - - - - - - - - - Frame 2 - - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time              Summary
     2 #      [195.158.225.81]  [my.domain.aaa.202]    70 0:00:29.710   29.710.434
02/13/2001 10:53:16 AM Expert: ICMP Host Unreachable

ICMP: Destination unreachable (Host unreachable)
ADDR  HEX                                                    ASCII
0000: 00 e0 16 7f eb 82 00 00 ef 06 1f 50 08 00 45 00 |
...........P..E.
0010: 00 38 00 00 00 00 f4 01 0a 82 c3 9e e1 51 XX XX |
.8..........áQ..
0020: XX ca 03 01 6a 17 00 00 00 00 45 00 00 28 8d 0a |
x...j.....E..(..
0030: 00 00 0f 06 57 02 XX XX XX ca c1 e7 ee 89 da a8 |
....W...x.Á.....
0040: 00 e2 45 d4 71 88                                | .âE.q.

- - - - - - - - - - - - - - - - - - - - - Frame 3 - - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time              Summary
     3 #      [193.231.227.181] [my.domain.aaa.202]    82 0:12:20.126   710.416.153
02/13/2001 11:05:06 AM Expert: Time-to-live exceeded in transmit

ICMP: Time exceeded (Time to live exceeded in transit)
ADDR  HEX                                                    ASCII
0000: 00 e0 16 7f eb 82 00 00 ef 06 1f 50 08 00 45 c0 |
...........P..E.
0010: 00 44 2a f2 00 00 e6 01 ec 16 c1 e7 e3 b5 XX XX |
.D*.......Á.ãµ..
0020: XX ca 0b 00 4b c8 00 00 00 00 45 00 00 28 5d 0a |
x...KÈ....E..(].
0030: 00 00 01 06 06 19 XX XX XX ca c1 e2 7d 78 b6 a8 |
........x.Áâ}x..
0040: 00 e2 5d 26 5d 88 00 00 00 00 50 02 ff ff e6 fb |
.â]&].....P....û
0050: 00 00                                            | ..
```

```
- - - - - - - - - - - - - - - - - - - - Frame 4 - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time             Summary
     4 #      [193.231.227.181] [my.domain.aaa.202]    82 0:12:24.952   4.826.199
02/13/2001 11:05:11 AM Expert: Time-to-live exceeded in transmit

ICMP: Time exceeded (Time to live exceeded in transit)
ADDR  HEX                                                       ASCII
0000: 00 e0 16 7f eb 82 00 00 ef 06 1f 50 08 00 45 c0 |
...........P..E.
0010: 00 44 75 38 00 00 e7 01 a0 d0 c1 e7 e3 b5 XX XX |
.Du8.....ÐÁ.ãµ..
0020: XX ca 0b 00 4b c8 00 00 00 00 45 00 00 28 c1 0a |
x...KÈ....E..(Á.
0030: 00 00 01 06 a2 18 XX XX XX ca c1 e2 7d 78 11 a8 |
........x.Áâ}x..
0040: 00 e2 51 3c ac 88 00 00 00 00 50 02 ff ff 48 e6 |
.âQ<......P...H.
0050: 00 00                                           | ..
```

```
- - - - - - - - - - - - - - - - - - - - Frame 5 - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time             Summary
     5 #      [193.231.227.181] [my.domain.aaa.202]    82 0:12:24.972   0.020.036
02/13/2001 11:05:11 AM Expert: Time-to-live exceeded in transmit

ICMP: Time exceeded (Time to live exceeded in transit)
ADDR  HEX                                                       ASCII
0000: 00 e0 16 7f eb 82 00 00 ef 06 1f 50 08 00 45 c0 |
...........P..E.
0010: 00 44 75 8e 00 00 e7 01 a0 7a c1 e7 e3 b5 XX XX |
.Du......zÁ.ãµ..
0020: XX ca 0b 00 4b c8 00 00 00 00 45 00 00 28 c1 0a |
x...KÈ....E..(Á.
0030: 00 00 01 06 a2 18 XX XX XX ca c1 e2 7d 78 11 a8 |
........x.Áâ}x..
0040: 00 e2 51 3c ac 88 00 00 00 00 50 02 ff ff 48 e6 |
.âQ<......P...H.
0050: 00 00                                           | ..

-snip-


- - - - - - - - - - - - - - - - - - - - Frame 16 - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time             Summary
    16 #      [146.188.12.161]  [my.domain.aaa.202]    70 1:30:25.087   1587.447.581
02/13/2001 12:23:11 PM Expert: Time-to-live exceeded in transmit

ICMP: Time exceeded (Time to live exceeded in transit)
ADDR  HEX                                                       ASCII
0000: 00 e0 16 7f eb 82 00 00 ef 06 1f 50 08 00 45 c0 |
...........P..E.
0010: 00 38 35 61 00 00 f0 01 dd f3 92 bc 0c a1 XX XX |
.85a.......¼....
0020: XX ca 0b 00 f5 8d 00 00 00 00 45 00 00 28 cd 0a |
x.........E..(Í.
0030: 00 00 01 06 5b c4 XX XX XX ca c1 e6 b7 c8 0a a8 |
....[...x.Á..È..
```

```
0040: 00 e1 12 60 e1 88                                  | .á.`á.

- - - - - - - - - - - - - - - - - - - - Frame 17 - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time             Summary
    17 #      [206.111.14.194]  [my.domain.aaa.202]    70 1:32:31.202   126.114.156
02/13/2001 12:25:17 PM Expert: ICMP Host Unreachable

ICMP: Destination unreachable (Host unreachable)
ADDR  HEX                                                  ASCII
0000: 00 e0 16 7f eb 82 00 00 ef 06 1f 50 08 00 45 00 |
...........P..E.
0010: 00 38 00 00 00 00 f2 01 d4 40 ce 6f 0e c2 XX XX |
.8.......@.o....
0020: XX ca 03 01 3f d1 00 00 00 00 45 00 00 28 a9 0a |
x...?.....E..(©.
0030: 00 00 0c 06 74 c4 XX XX XX ca c1 e6 b7 c8 9f a8 |
....t...x.Á..È..
0040: 00 e1 5a 1b c2 88                                  | .áZ...

- - - - - - - - - - - - - - - - - - - - Frame 18 - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time             Summary
    18        [200.199.245.2]   [my.domain.aaa.24]     60 1:44:37.047   725.845.216
02/13/2001 12:37:23 PM TCP: D=18491 S=53139 RST ACK=1 WIN=0
ADDR  HEX                                                  ASCII
0000: 00 e0 16 7f eb 82 00 00 ef 06 1f 50 08 00 45 00 |
...........P..E.
0010: 00 28 8a 86 00 00 f4 06 67 de c8 c7 f5 02 XX XX |
.(......g.ÈÇ....
0020: XX 18 cf 93 48 3b 6f ba 04 56 00 00 00 01 50 14 |
x...H;o..V....P.
0030: 00 00 4f 85 00 00 00 00 00 00 00 00              | ..O.........
```

1. Source of Trace:

MyNet.edu

2. Detect was generated by:

- Sniffer outside the firewall filtering for traffic to specific subnets. *All* traffic from outside to these subnets is prohibited by firewall rules.

3. Probability the source address was spoofed:

There are 3 types of incoming packets represented in this particular trace ... TCP Resets, ICMP messages, NetBios queries.

The TCP resets have a better than even chance of having come from real source hosts, as a response to stimuli which originally spoofed our

my.domain.aaa.0 host addresses. Of those that responded to pings, the TTL's seem plausible. A couple did not respond, but may have been filtered or taken down as a result of such attacks. Packets look normal.

The NetBios queries ...
On the hypothesis that the query may have a reconnaissance purpose, the chances are good the source host addresses are not spoofed. The packet's TTL's do not suggest spoofing.

The ICMP messages ...
Are likely to be from real source hosts, in response to stimuli wherein our my.domain.aaa.0 subnet host addresses had been used as spoofed source addresses. In the HEX portion of the traces, above, I have bolded the alleged original source address of the IP packet that triggered the ICMP message, "XX XX XX ca". This represents an address on one of our subnets which is not and has never been in use, but which triggers lots of these ICMP messages.

4. Description of attack:

The trace was taken on 2/13/01 between 10:52:46 and 15:23:29. Inter-arrival times range from less than 1 second to over 1 hour.

All destination hosts seen in the trace are non-existent (the subnets exist but with only few hosts on them, which have no need to communicate outside of our LAN).

Since setting up the firewall to deny any incoming traffic to particular subnets and particular addresses, the firewall logs reveal a constant level of "noise" being filtered by these rules. Of course these rules miss the noise coming to other subnets in our domain, which are required to be open, or which are not yet in use at all, and so are not filtered.

The activity on the firewall logs led me to look further, by setting up the Sniffer outside of the firewall to accomplish high fidelity packet capture for traffic addressed to the subnets in question. (If similar packets are sent also to subnets other than this one they were not captured by this Sniffer filter - thus my experiment in Detect #1 with capturing all port 27374 traffic, for instance. This limitation shows the potential value, if it was practicable, of maintaining and analyzing tcpdump header capture traces of all traffic, or of utilizing some form of semi-automatic IDS.)

Identification of Source:

The following section lists source host addresses, what sort of packets they sent, and results of pings or traceroutes used to determine their nature. Most of the pings were done on a Windows host from myhomeisp.net.

TCP
The following were seen to send Resets (or RST ACK) to non-existent host addresses:

[Source Addresses] - [Nature of Packet] - [Subnet MY.NET.aaa.0]

This host answers a ping.
193.231.238.137 - **RST ACK** - 1 packet to host 202

```
Pinging 193.231.238.137.catv.rdsor.ro [193.231.238.137]
Reply from 193.231.238.137: bytes=32 time=350ms TTL=233
```

This host answers a ping.
63.144.122.149 - **RST ACK** - 1 packet to 202

```
Pinging 63.144.122.149 with 32 bytes of data:
Reply from 63.144.122.149: bytes=32 time=265ms TTL=240
```

This host is filtered from receiving pings.
200.199.245.2 - **RST ACK** or just **RST** but *not* to host 202 ... 3 packets received more than 20 minutes apart addressed to 3 different nonexistent hosts.

```
Ping test from windows.myhomeisp.net:
Pinging 200.199.245.2 with 32 bytes of data:
Reply from 200.255.153.246: Destination net unreachable.

Then I tested from unix.mydomain.net:
PING 200.199.245.2 (200.199.245.2): 56 data bytes
36 bytes from 200.193.234.65: Communication prohibited by filter
(host may have been taken down or protected since original capture).
```

This host did not respond.
24.64.46.13 - RST ACK - 1 to 120.143

```
Pinging 24.64.46.13 with 32 bytes of data:
Request timed out.
(From unix.mydomain.net, traceroute timed out after 12 hops, somewhere
in Canada.)
```

24.64.46.13 and 200.199.245.2 neither resolve nor respond to ICMP echo.

ICMP packets

193.230.183.200 is represented 3 times, as an "original" destination address in the embedded IP headers inside the ICMP packets, however there are 3 seperate source addresses represented, as follows: note that Alter.Net and KPNQwest.net both appear as stops along the way in a traceroute to the original destination address:

Source Addresses - ICMP type - Subnet my.domain.aaa.0

146.188.12.161 - **TTL** - 1 to host 202

```
Pinging 112.ATM0-0-0.XR1.BUD1.Alter.Net [146.188.12.161]
Reply from 146.188.12.161: bytes=32 time=308ms TTL=243
```

206.111.14.194 - **HOST UNREACHABLE** - 1 to 202

```
Pinging 206.111.14.194 with 32 bytes of data:
Reply from 206.111.14.194: bytes=32 time=301ms TTL=245
```

134.222.199.22 - ICMP **TTL Expired** - 1 to 202

```
Pinging PanTel-gw1.KPNQwest.net [134.222.199.22] with
Reply from 134.222.199.22: bytes=32 time=282ms TTL=240
```

I decided to test to the "original desitination address" from the embedded ip header in the ICMP packet header:

[testing "original destination address"

```
From windows.myhomeisp.net:
Pinging drtvbuc.p.ew.ro [193.230.183.200] with 32 bytes of data:
Reply from 193.230.183.200: bytes=32 time=394ms TTL=240
```

```
From unix.my.domain:
% traceroute drtvbuc.p.ew.ro
traceroute: unknown host drtvbuc.p.ew.ro
% traceroute 193.230.183.200
traceroute to 193.230.183.200 (193.230.183.200), 30 hops max, 40 byte
packets

*** snipped ***

20  412.ATM12-0-0.GW1.BUD1.Alter.Net (146.188.12.141)  203.981 ms
205.095 ms  s
21  Pantel-gw.customer.ALTER.NET (146.188.48.130)  222.219 ms  194.895
ms  205.s
22  atm6-1-0-1.bud2core1.pantel.net (212.24.160.201)  207.869 ms
204.344 ms  1s
23  * * * timed out]
```

The discrepancy between my work net's behavior and my home net's behavior is interesting, but beyond the scope of the present study. Suffice it to say that the address in question does seem to exist, and that two at least of the source networks from which I captured packets are in the path to the original destination address, and so would have been in a position to send the icmp messages.

212.163.65.75 - **WINS Query** - 3 packets to host 103

```
Pinging 212.163.65.75 with 32 bytes of data:
Reply from 212.163.65.75: bytes=32 time=541ms TTL=111
responds to ping
TTL plausible
1/second
possibly looking for host to respond to WINS query?
```

195.158.225.81 - **HOST UNREACHABLE** - 1 packet to host 202

```
Pinging bebru203-tc-p2-0.ebone.net [195.158.225.81]
Reply from 195.158.225.81: bytes=32 time=284ms TTL=245
```

193.231.227.181 - **TTL** - 5 packets to host 202

```
Pinging bucharest-bb2-fe0.rdsnet.ro [193.231.227.181]
Request timed out.
```

213.174.71.2 - **HOST UNREACHABLE** - 3 packets to host 202

```
Pinging nlams303-tc-p1-0.ebone.net [213.174.71.2]
Reply from 213.174.71.2: bytes=32 time=255ms TTL=244
```

207.96.164.25 **- TTL** - 1 packet to host 202

```
Pinging fa5-1-0-tdl-beaubien.videotron.net [207.96.164.25]
Reply from 207.96.164.25: bytes=32 time=193ms TTL=244
```

62.128.1.6 - **TTL** - 1 packet to host 202

```
Pinging f00-nbg3.noris.net [62.128.1.6] with 32 by
Reply from 62.128.1.6: bytes=32 time=298ms TTL=240
```

5. Attack mechanism:

NetBios Queries

Only the NetBios queries might elicit any response if they encountered an actual host, or they might elicit a "host not available" if allowed through the firewall, so they could be an effort at reconnaissance of our network.

ICMP, TCP RST,

All the Resets and ICMP messages in this trace are likely to be second order responses to stimuli from crafted, spoofed packets. Analysis of the source addresses, the "original destination addresses" of the ICMP packets, and their TTL's suggest for the most part these are second order responses.

The host-202 address is a regular target of this kind of traffic, not always from the same sources.

This traffic would not elicit any response from us. If part of a larger pattern it could represent a Denial of Service against the networks whose hosts or routers are reflecting these Reset and ICMPs back to us.

One way to determine the likelyhood of spoofed source addresses is to see whether a ping or traceroute from my location to that address correlates with the reported TTL of the packet under study. In the "TCP/UDP" section, below, I show the reported TTL's in the second column, and the TTL's I got when pinging the source addresses, in the fourth column. I note that in all but one instance, the values are close to one another, suggesting that the packets under study could have come from the alleged source addresses. Then, in the "ICMP" section below, I check to see whether the reported source address of the ICMP message could plausibly have been the actual sender. I sent a traceroute to the alleged destination address of the original ip packet from the ICMP header, and looked to see whether the origin of the ICMP packet appeared in the network path. In the entry for original destination address 193.226.195.120 for instance, we see that the source address of the ICMP message did appear in the traceroute (it is shown resolved to it's dns name), so that we know it plausibly could have been the one to send that ICMP packet, and we note that the sum of the TTL value in the received packet and the hop count to the source address is 255, a plausible starting TTL value, so there's a lack of hard evidence to suggest the ICMP packet did not come from where it said it came from.

TCP/UDP				
source addr	TTL	ping result	TTL	
193.231.238.137	229	.catv.rdsor.ro	232	
212.163.65.75	112	yes	112	
200.199.245.2	244	200.193.234.65: Communication prohibited by filter	247	
63.144.122.149	52	call.me.yer-daddy.com	244	
24.64.46.13	114	h24-64-46-13.cg.shawcable.net	115	
ICMP				
original dest addr	TTL	src ip	src ip in tracert?	hops
193.231.238.137	244	195.158.225.81	yes, ebone.net	13
193.226.125.120	231	193.231.227.181	bucharest-bb2-fe0.rdsnet.ro	24
193.226.125.120	243	213.174.71.2	nlams303-tc-p1-0.ebone.net	14
24.201.74.83	243	207.96.164.25	fa5-1-0-tdl-beaubien.videotron.net	12
193.230.183.200	240	146.188.12.161 112.ATM0-0-0.XR1.BUD1.Alter.Net	no	original target in 12 hops - drtvbuc.p.ew.ro
193.230.183.200	242	206.111.14.194	no	original target in 12 hops - drtvbuc.p.ew.ro
62.128.24.131	242	62.128.1.6 f00-nbg3.noris.net TTL 240	no but plausible	original target in 17 hops - weltherrschaft.maze.de
193.230.183.200	245	134.222.199.22 PanTel-gw1.KPNQwest.net	no	original target in 12 hops - drtvbuc.p.ew.ro

6. Correlations:

As evidence of the ongoing nature of "attacks" against this subnet and the 202 host address in particular, firewall logs and tcpdump is shown from 1 1/2 weeks later:

The following "destination unreachable" entries appeared in the FreeBSD ipfw firewall logs co-incidently with the tcpdump collection shown below.

Key to Firewall Log
[Date and Time][Action][Protocol:Type][src addr][dest addr]

```
Feb 24 12:36:26  Deny ICMP:3.1 152.63.25.189 my.domain.aaa.202
```

```
Feb 24 12:37:56  Deny ICMP:3.1 152.63.25.189 my.domain.aaa.202
Feb 24 12:39:26  Deny ICMP:3.1 152.63.25.189 my.domain.aaa.202
Feb 24 12:39:56  Deny ICMP:3.1 152.63.25.189 my.domain.aaa.202
Feb 24 12:40:56  Deny ICMP:3.1 152.63.25.189 my.domain.aaa.202
Feb 24 12:41:26  Deny ICMP:3.1 152.63.25.189 my.domain.aaa.202
Feb 24 12:41:27  Deny ICMP:3.1 152.63.25.189 my.domain.aaa.202
Feb 24 12:42:26  Deny ICMP:3.1 152.63.25.189 my.domain.aaa.202
Feb 24 12:45:57  Deny ICMP:3.1 152.63.25.189 my.domain.aaa.202
Feb 24 12:46:27  Deny ICMP:3.1 152.63.25.189 my.domain.aaa.202
Feb 24 12:47:57  Deny ICMP:3.1 152.63.25.189 my.domain.aaa.202
Feb 24 12:48:27  Deny ICMP:3.1 152.63.25.189 my.domain.aaa.202
Feb 24 12:49:27  Deny ICMP:3.1 152.63.25.189 my.domain.aaa.202
Feb 24 12:49:57  Deny ICMP:3.1 152.63.25.189 my.domain.aaa.202
Feb 24 12:51:27  Deny ICMP:3.1 152.63.25.189 my.domain.aaa.202
Feb 24 12:52:27  Deny ICMP:3.1 152.63.25.189 my.domain.aaa.202
```

DNS Address confirmation from Windows ping, for correlation:

```
Pinging 192.ATM7-0.GW4.EWR1.ALTER.NET [152.63.25.189]
Reply from 152.63.25.189: bytes=32 time=184ms TTL=248)
```

Summary of packets, using tcpdump to read a tcpdump file called "tcpdump022401":

Key to tcpdump:
[hh:mm:sec.xxxxx][src addr]>[dest addr]:[icmp packet]:[icmp message]

```
% tcpdump -r tcpdump022401
12:39:04.312940 192.ATM7-0.GW4.EWR1.ALTER.NET > my.domain.aaa.202:
icmp: host 208.216.112.166 unreachable
12:39:48.528088 192.ATM7-0.GW4.EWR1.ALTER.NET > my.domain.aaa.202:
icmp: host 208.216.112.166 unreachable
12:40:30.546955 192.ATM7-0.GW4.EWR1.ALTER.NET > my.domain.aaa.202:
icmp: host 208.216.112.166 unreachable
12:41:13.412967 192.ATM7-0.GW4.EWR1.ALTER.NET > my.domain.aaa.202:
icmp: host 208.216.112.166 unreachable
12:41:16.665768 192.ATM7-0.GW4.EWR1.ALTER.NET > my.domain.aaa.202:
icmp: host 208.216.112.166 unreachable
12:42:00.752591 192.ATM7-0.GW4.EWR1.ALTER.NET > my.domain.aaa.202:
icmp: host 208.216.112.166 unreachable
12:42:39.396350 192.ATM7-0.GW4.EWR1.ALTER.NET > my.domain.aaa.202:
icmp: host 208.216.112.166 unreachable
12:42:44.642254 192.ATM7-0.GW4.EWR1.ALTER.NET > my.domain.aaa.202:
icmp: host 208.216.112.166 unreachable
12:45:41.270484 192.ATM7-0.GW4.EWR1.ALTER.NET > my.domain.aaa.202:
icmp: host 208.216.112.166 unreachable
```

Looking at the same tcpdump file, showing the hex representation of the IP packet, we note that the IP and ICMP packets seem normal enough, and we conclude it is highly likely that once again we are seeing second order effects of an attack upon 192.ATM7-0.GW4.EWR1.ALTER.NET.

Looking at the original IP packet header embedded in the ICMP packet: The TCP destination port is 226 ; source port varies.(Port 226 is shown originally listed as "reserved" in *ftp://ftp.isi.edu/in-notes/iana/assignments/port-numbers)*

The HEX section of these tcpdump traces begin with the IP header.

```
% tcpdump -r tcpdump022401 -x
12:39:04.312940 192.ATM7-0.GW4.EWR1.ALTER.NET > my.domain.aaa.202:
icmp: host 208.216.112.166 unreachable
                        4500 0038 0000 0000 f601 fb75 983f 19bd
                        XXXX XXca 0301 e6ed 0000 0000 4500 0028
                        710a 0000 1206 def4 XXXX XXca d0d8 70a6
                        d5a8 00e2 5efd e088
12:39:48.528088 192.ATM7-0.GW4.EWR1.ALTER.NET > my.domain.aaa.202:
icmp: host 208.216.112.166 unreachable
                        4500 0038 0000 0000 f601 fb75 983f 19bd
                        XXXX XXca 0301 a8ae 0000 0000 4500 0028
                        a90a 0000 1206 a6f4 XXXX XXca d0d8 70a6
                        5fa8 00e2 713d 8288
12:40:30.546955 192.ATM7-0.GW4.EWR1.ALTER.NET > my.domain.aaa.202:
icmp: host 208.216.112.166 unreachable
                        4500 0038 0000 0000 f601 fb75 983f 19bd
                        XXXX XXca 0301 aac1 0000 0000 4500 0028
                        390a 0000 1206 16f5 XXXX XXca d0d8 70a6
                        8ba8 00e2 472a 7e88
12:41:13.412967 192.ATM7-0.GW4.EWR1.ALTER.NET > my.domain.aaa.202:
icmp: host 208.216.112.166 unreachable
                        4500 0038 0000 0000 f601 fb75 983f 19bd
                        XXXX XXca 0301 0cab 0000 0000 4500 0028
                        9d0a 0000 1206 b2f4 XXXX XXca d0d8 70a6
                        e6a8 00e2 3b40 cd88
12:41:16.665768 192.ATM7-0.GW4.EWR1.ALTER.NET > my.domain.aaa.202:
icmp: host 208.216.112.166 unreachable
                        4500 0038 0000 0000 f601 fb75 983f 19bd
                        XXXX XXca 0301 6c82 0000 0000 4500 0028
                        710a 0000 1206 def4 XXXX XXca d0d8 70a6
                        15a8 00e2 596a 2088
12:42:00.752591 192.ATM7-0.GW4.EWR1.ALTER.NET > my.domain.aaa.202:
icmp: host 208.216.112.166 unreachable
                        4500 0038 0000 0000 f601 fb75 983f 19bd
                        XXXX XXca 0301 ce6b 0000 0000 4500 0028
                        d50a 0000 1206 7af4 XXXX XXca d0d8 70a6
                        70a8 00e2 4d80 6f88
12:42:39.396350 192.ATM7-0.GW4.EWR1.ALTER.NET > my.domain.aaa.202:
icmp: host 208.216.112.166 unreachable
                        4500 0038 0000 0000 f601 fb75 983f 19bd
                        XXXX XXca 0301 d07e 0000 0000 4500 0028
                        650a 0000 1206 eaf4 XXXX XXca d0d8 70a6
                        9ca8 00e2 236d 6b88
```

-snip-

7. Evidence of active targeting:

The destination addresses of the captured traffic do not exist on our network. For some time now, we have been denying any traffic coming from outside to the subnets in question. The traffic to host 202 in particular, and generally the traffic that results in our receipt of Resets and ICMP messages such as TTL exceed or host unreachable, would do the attacker no particular good, except for the annoyance factor. It

seems more likely that the apparent source addresses of these packets are the real target.

In the case of the NetBios traffic, this could well be part of larger network scans, which might actually elicit a response of some kind should it be allowed into our network, where there are hosts to respond.

The large amount of traffic that we see to host 202 in particular is strange, one might speculate that this particular host address may be coded into some exploit that is widely distributed and used in generating spoofed addresses.

8. Severity:

Severity is calculated by looking at the sum of the ratings for the severity of the attack and the criticality of the target, and subtracting from that the sum of the network and system countermeasures in effect.

(severity + criticality) - (network countermeasures + system countermeasures) = severity

Severity - The netbios traffic seen above could represent reconnaissance. - 2
Criticality - These traces are against non-existent hosts. - 1
Network Countermeasures - These conditions came to our attention only because they showed up as being denied in firewall logs. - 5
System Countermeasures. - No systems were exposed in these particular traces. - 1

(2+1)-(5+1)= -3

9. Defensive recommendation:

Although the severity of the traces shown here is low, we should be wondering about the traffic that was not caught either by the firewall or the Sniffer and tcpdump filters that were used to study this traffic. As an .edu that still tries to keep an open channel to the internet for the benefit of our user community, we should find a way to monitor for the case of scans that might return information about our network or servers, or other more directly lethal attacks. It would be good to set up some form of Intrusion Detection system that would allow us to view potentially harmful activity that would otherwise be allowed into our network.

10. Multiple choice test question:

What information begins at offset 0x08 in some types of ICMP headers?

A. The destination port of the ICMP message.

B. The number of hops to the source network.

C. The IP header of the original packet that prompted the ICMP response.

D. The ICMP message type.

Answer: C.

***
what is port 226? Reserved?
http://www.snort.org/Database/portsearch.asp?Port=226
responded "no records returned"

DETECT #3: FIREWALL DETECTS INCOMING NFS PACKETS IN VIOLATION OF RULE

Detect - Firewall detects incoming NFS traffic prohibited by rule.

Key to FreeBSD ipfw Firewall Log
[Date and Time][Action][Protocol][src addr:port][dest addr:port][interface]

Network Detect NFS traffic from outside our network

```
excerpt
Feb 24 13:21:27 Deny TCP 206.253.222.77:80 MY.NET.1.103:2049 in via xl1
Feb 24 13:21:57 Deny TCP 206.253.222.77:80 MY.NET.1.103:2049 in via xl1
Feb 24 13:21:57 Deny TCP 206.253.222.77:80 MY.NET.1.103:2049 in via xl1
Feb 24 13:21:57 Deny TCP 206.253.222.77:80 MY.NET.1.103:2049 in via xl1
Feb 24 13:38:27 Deny TCP 208.184.29.50:80 MY.NET.1.103:4045 in via xl1
Feb 24 13:49:58 Deny TCP 205.219.162.10:113 MY.NET.1.110:4045 in via
xl1
Feb 24 13:49:58 Deny TCP 205.219.162.10:113 MY.NET.1.110:4045 in via
xl1
-snip-
```

Key to Sniffer Summary Information
Frame - Frame number in trace.
Status - Whether the packet is Marked as a reference point in the trace.
Source Address - From the IP header
Destination Address - From the IP header
Size - total length of the captured packet. The Sniffer doesn't capture the ethernet trailing checksum.
Rel. Time - The elapsed time since the trace began or from a set marker.
Delta Time - The time since the previous frame.
Abs. Time - The date and time-of-day the packet was received by the Sniffer®.

Summary - The Sniffer®'s brief description of the packet, containing any messages generated by the Sniffer "Expert" followed by a summary of the highest level header information the Sniffer was able to decode from the packet.

```
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time              Summary
     1 M      [MY.NET.1.103]   [206.41.20.6]        60 0:00:00.000   0.000.000
02/16/2001 01:50:39 PM TCP: D=80 S=2049 SYN SEQ=3792052757 LEN=0 WIN=16384
     2        [206.41.20.6]      [MY.NET.1.103]       60 0:00:00.180   0.180.778
02/16/2001 01:50:39 PM TCP: D=2049 S=80 SYN ACK=3792052758 SEQ=2843940323 LEN=0 WIN=8760
     3        [206.41.20.6]      [MY.NET.1.103]       60 0:00:01.658   1.478.192
02/16/2001 01:50:40 PM TCP: D=2049 S=80 SYN ACK=3792052758 SEQ=2843940323 LEN=0 WIN=8760
     4        [MY.NET.1.103]   [206.41.20.6]        60 0:00:02.987   1.328.207
02/16/2001 01:50:42 PM TCP: D=80 S=2049 SYN SEQ=3792052757 LEN=0 WIN=16384
     5        [206.41.20.6]      [MY.NET.1.103]       60 0:00:03.090   0.103.511
02/16/2001 01:50:42 PM TCP: D=2049 S=80     ACK=3792052758 WIN=8760
     6        [206.41.20.6]      [MY.NET.1.103]       60 0:00:04.027   0.936.363
02/16/2001 01:50:43 PM TCP: D=2049 S=80 SYN ACK=3792052758 SEQ=2843940323 LEN=0 WIN=8760
     7        [206.41.20.6]      [MY.NET.1.103]       60 0:00:08.823   4.796.413
02/16/2001 01:50:47 PM TCP: D=2049 S=80 SYN ACK=3792052758 SEQ=2843940323 LEN=0 WIN=8760
     8        [MY.NET.1.103]   [206.41.20.6]        60 0:00:08.986   0.162.977
02/16/2001 01:50:48 PM TCP: D=80 S=2049 SYN SEQ=3792052757 LEN=0 WIN=16384
     9        [206.41.20.6]      [MY.NET.1.103]       60 0:00:09.105   0.118.642
02/16/2001 01:50:48 PM TCP: D=2049 S=80     ACK=3792052758 WIN=8760
    10        [206.41.20.6]      [MY.NET.1.103]       60 0:00:18.435   9.330.080
02/16/2001 01:50:57 PM TCP: D=2049 S=80 SYN ACK=3792052758 SEQ=2843940323 LEN=0 WIN=8760
    11        [MY.NET.1.103]   [206.41.20.6]        60 0:00:20.987   2.552.756
02/16/2001 01:51:00 PM TCP: D=80 S=2049 SYN SEQ=3792052757 LEN=0 WIN=16384
    12 #      [206.41.20.6]      [MY.NET.1.103]       60 0:00:37.610   16.622.167
02/16/2001 01:51:16 PM Expert: Ack Too Long (16622ms)
              TCP: D=2049 S=80 SYN ACK=3792052758 SEQ=2843940323 LEN=0 WIN=8760
    13        [MY.NET.1.103]   [206.41.20.6]        60 0:00:44.994   7.383.974
02/16/2001 01:51:24 PM TCP: D=80 S=2049 SYN SEQ=3792052757 LEN=0 WIN=16384
    14        [206.41.20.6]      [MY.NET.1.103]       60 0:00:45.168   0.174.665
02/16/2001 01:51:24 PM TCP: D=2049 S=80 SYN ACK=3792052758 SEQ=1194538129 LEN=0 WIN=8760
    15        [206.41.20.6]      [MY.NET.1.103]       60 0:00:46.605   1.436.970
02/16/2001 01:51:25 PM TCP: D=2049 S=80 SYN ACK=3792052758 SEQ=1194538129 LEN=0 WIN=8760
    16        [206.41.20.6]      [MY.NET.1.103]       60 0:00:49.027   2.422.269
02/16/2001 01:51:28 PM TCP: D=2049 S=80 SYN ACK=3792052758 SEQ=1194538129 LEN=0 WIN=8760
    17        [206.41.20.6]      [MY.NET.1.103]       60 0:00:53.797   4.769.761
02/16/2001 01:51:32 PM TCP: D=2049 S=80 SYN ACK=3792052758 SEQ=1194538129 LEN=0 WIN=8760
    18        [206.41.20.6]      [MY.NET.1.103]       60 0:01:03.396   9.599.064
02/16/2001 01:51:42 PM TCP: D=2049 S=80 SYN ACK=3792052758 SEQ=1194538129 LEN=0 WIN=8760
    19        [206.41.20.6]      [MY.NET.1.103]       60 0:01:22.672   19.276.091
02/16/2001 01:52:01 PM TCP: D=2049 S=80 SYN ACK=3792052758 SEQ=1194538129 LEN=0 WIN=8760
    20        [206.41.20.6]      [MY.NET.1.103]       60 0:02:01.064   38.391.729
02/16/2001 01:52:40 PM TCP: D=2049 S=80 SYN ACK=3792052758 SEQ=1194538129 LEN=0 WIN=8760
    21 #      [206.41.20.6]      [MY.NET.1.103]       60 0:03:17.891   76.826.872
02/16/1992 01:53:56 PM Expert: Idle Too Long
              TCP: D=2049 S=80 SYN ACK=3792052758 SEQ=1194538129 LEN=0 WIN=8760
              http request from client:2049 to server:80

- - - - - - - - - - - - - - - - - - - - Frame 1 - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time              Summary
     1 M      [209.94.119.131]  [MY.NET.1.103]       60 0:00:00.000   0.000.000
02/16/2001 12:46:03 PM TCP: D=80 S=2049 SYN SEQ=1496400322 LEN=0 WIN=8192
ADDR  HEX                                                ASCII
0000: 00 90 27 9c 23 da 00 00 ef 06 1f 50 08 00 45 00 | ..'.#......P..E.
0010: 00 2c 40 0f 40 00 75 06 dc eb d1 5e 77 83 xx xx | .,@.@.u....^w...
0020: 01 67 08 01 00 50 59 31 41 c2 00 00 00 00 60 02 | .g...PY1A.....`.
0030: 20 00 ec 10 00 00 02 04 05 b4 00 00             |  ............

- - - - - - - - - - - - - - - - - - - - Frame 2 - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time              Summary
     2        [MY.NET.1.103]   [209.94.119.131]     60 0:00:00.000   0.000.024
02/16/2001 12:46:03 PM TCP: D=2049 S=80 SYN ACK=1496400323 SEQ=1509209913 LEN=0 WIN=17520
```

```
ADDR  HEX                                                 ASCII
0000: 00 00 ef 06 1f 50 00 90 27 9c 23 da 08 00 45 00 | .....P..'.#...E.
0010: 00 2c 26 11 40 00 40 06 2b ea xx xx 01 67 d1 5e | .,&.@.@.+....g.^
0020: 77 83 00 50 08 01 59 f4 b7 39 59 31 41 c3 60 12 | w..P..Y..9Y1A.`.
0030: 44 70 b6 61 00 00 02 04 05 b4 00 00             | Dp.a........

- - - - - - - - - - - - - - - - - - - - Frame 3 - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time             Summary
     3        [209.94.119.131]  [MY.NET.1.103]      60 0:00:00.218   0.218.667
02/16/2001 12:46:03 PM TCP: D=80 S=2049     ACK=1509209914 WIN=8760
ADDR  HEX                                                 ASCII
0000: 00 90 27 9c 23 da 00 00 ef 06 1f 50 08 00 45 00 | ..'.#......P..E.
0010: 00 28 b2 0f 40 00 75 06 6a ef d1 5e 77 83 xx xx | .(..@.u.j..^w...
0020: 01 67 08 01 00 50 59 31 41 c3 59 f4 b7 3a 50 10 | .g...PY1A.Y..:P.
0030: 22 38 f0 56 00 00 00 00 00 00 00 00             | "8.V........

- - - - - - - - - - - - - - - - - - - - Frame 4 - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time             Summary
     4        [209.94.119.131]  [MY.NET.1.103]     403 0:00:00.220   0.002.280
02/16/2001 12:46:03 PM HTTP: C Port=0 GET /welcome/slideshow/8.html HTTP/1.0
ADDR  HEX                                                 ASCII
0000: 00 90 27 9c 23 da 00 00 ef 06 1f 50 08 00 45 00 | ..'.#......P..E.
0010: 01 85 b3 0f 40 00 75 06 68 92 d1 5e 77 83 xx xx | ....@.u.h..^w...
0020: 01 67 08 01 00 50 59 31 41 c3 59 f4 b7 3a 50 18 | .g...PY1A.Y..:P.
0030: 22 38 e3 75 00 00
What is this HTTP port 0 about?
- - - - - - - - - - - - - - - - - - - - Frame 5 - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time             Summary
     5        [MY.NET.1.103]   [209.94.119.131]  1514 0:00:00.234   0.013.793
02/16/2001 12:46:03 PM HTTP: R Port=2049 HTML Data
ADDR  HEX                                                 ASCII
0000: 00 00 ef 06 1f 50 00 90 27 9c 23 da 08 00 45 00 | .....P..'.#...E.
0010: 05 dc 26 50 40 00 40 06 25 fb xx xx 01 67 d1 5e | ..&P@.@.%û...g.^
0020: 77 83 00 50 08 01 59 f4 b7 3a 59 31 43 20 50 10 | w..P..Y..:Y1C P.
0030: 44 70 28 3b 00 00

- - - - - - - - - - - - - - - - - - - - Frame 6 - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time             Summary
     6        [MY.NET.1.103]   [209.94.119.131]  1514 0:00:00.236   0.001.241
02/16/2001 12:46:03 PM HTTP: R Port=2049 HTML Data
ADDR  HEX                                                 ASCII
0000: 00 00 ef 06 1f 50 00 90 27 9c 23 da 08 00 45 00 | .....P..'.#...E.
0010: 05 dc 26 51 40 00 40 06 25 fa xx xx 01 67 d1 5e | ..&Q@.@.%....g.^
0020: 77 83 00 50 08 01 59 f4 bc ee 59 31 43 20 50 10 | w..P..Y.¼.Y1C P.
0030: 44 70 66 9f 00 00

- - - - - - - - - - - - - - - - - - - - Frame 7 - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time             Summary
     7        [209.94.119.131]  [MY.NET.1.103]      60 0:00:00.387   0.151.403
02/16/2001 12:46:03 PM TCP: D=80 S=2049     ACK=1509212834 WIN=8760
ADDR  HEX                                                 ASCII
0000: 00 90 27 9c 23 da 00 00 ef 06 1f 50 08 00 45 00 | ..'.#......P..E.
0010: 00 28 d2 0f 40 00 75 06 4a ef d1 5e 77 83 xx xx | .(..@.u.J..^w...
0020: 01 67 08 01 00 50 59 31 43 20 59 f4 c2 a2 50 10 | .g...PY1C Y...P.
0030: 22 38 e3 91 00 00 00 00 00 00 00 00             | "8ã.........

- - - - - - - - - - - - - - - - - - - - Frame 8 - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time             Summary
     8        [MY.NET.1.103]   [209.94.119.131]    619 0:00:00.387   0.000.534
02/16/2001 12:46:03 PM HTTP: R Port=2049 HTML Data
ADDR  HEX                                                 ASCII
0000: 00 00 ef 06 1f 50 00 90 27 9c 23 da 08 00 45 00 | .....P..'.#...E.
0010: 02 5d 26 85 40 00 40 06 29 45 xx xx 01 67 d1 5e | .]&.@.@.)E...g.^
0020: 77 83 00 50 08 01 59 f4 c2 a2 59 31 43 20 50 19 | w..P..Y...Y1C P.
0030: 44 70 34 04 00 00
```

```
- - - - - - - - - - - - - - - - - - - - - Frame 9 - - - - - - - - - - - - - - - - - - - -
 Frame Status Source Address    Dest. Address      Size Rel. Time     Delta Time    Abs.
Time              Summary
     9        [209.94.119.131]  [MY.NET.1.103]      60 0:00:00.502   0.114.376
02/16/2001 12:46:03 PM TCP: D=80 S=2049     ACK=1509213400 WIN=8195
ADDR  HEX                                                  ASCII
0000: 00 90 27 9c 23 da 00 00 ef 06 1f 50 08 00 45 00 | ..'.#.......P..E.
0010: 00 28 f9 0f 40 00 75 06 23 ef d1 5e 77 83 xx xx | .(..@.u.#..^w...
0020: 01 67 08 01 00 50 59 31 43 20 59 f4 c4 d8 50 10 | .g...PY1C Y...P.
0030: 20 03 e3 90 00 00 00 00 00 00 00 00             |  .ã.........  -snip-
```

1. Source of Trace:

source mynet.edu

2. Detect was generated by:

The original detect came from firewall logs and was subsequently enhanced by Sniffer traces.

3. Probability the source address was spoofed:

Low.
Source addresses are virtually all showing up as legitimate addresses. Analysis of traffic shows apparently genuine connection attempts.

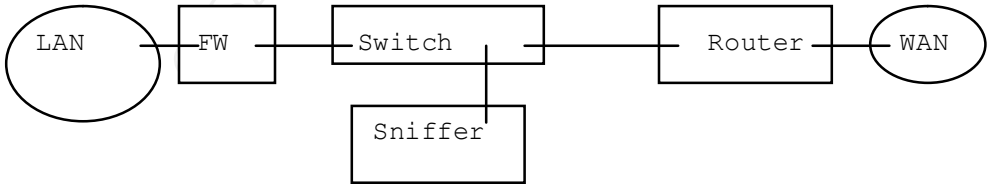
4. Description of attack:

Duration of original trace: 1:07:53.912
Absolute time: 2/16/01 12:46:03 UTC -5:00
This has been seen repeatedly on subsequent occasions.

The servers use NFS to communicate with a disc server. A firewall rule was put in place to protect mynet.edu's server subnet against unwanted NFS traffic coming to the server subnet from anywhere else.

FW logs are catching violations of the rule against NFS traffic (ports 2049, 4045)* addressed from outside of campus to mynet.edu's server subnet. Curiously, most of the log entries show source port of http. Others are port TCP 113.

As a followup, Sniffer traces were taken to look at the traffic which generates packets to port 2049, 4045 addressed to the server subnet. The Sniffer traces were taken via a switch mirror port between the campus firewall and the internet router.



DESCRIPTION OF DESTINATION HOSTS:
MY.NET.1.103 is the main institutional web and proxy server (among other services).
MY.NET.1.110 is a server hosting individuals' network file services, and web pages, and is also the main mail server.

CHECKING THE ALLEGED SOURCE ADDRESSES:
- outside hosts' source addresses - pings from a Windows host –

```
152.163.180.56
Pinging ads.web.aol.com [152.163.180.56] with 32 byte
Reply from 152.163.180.56: bytes=32 time=188ms TTL=48

199.221.131.101
Pinging 199.221.131.101 with 32 bytes of data: (shopathome.com)
Reply from 199.221.131.101: bytes=32 time=267ms TTL=245

204.60.148.65
Pinging 204.60.148.65 with 32 bytes of data: (no response at http)
Reply from 204.60.148.65: bytes=32 time=315ms TTL=233
```

```
>whois -h whois.arin.net 204.60.148.65

Southern New England Telephone (NETBLK-SNET-CIDR001)
   27 Butler St.
   Meriden, CT 06451-4101
   US

   Netname: SNET-CIDR001
   Netblock: 204.60.0.0 - 204.60.255.255

   Coordinator:
      Devetzis, Taso N  (TND-ARIN)  devetzis@SNET.NET
      +1 203 771 8917 (FAX) +1 203 771 2008

   Domain System inverse mapping provided by:

   NS1.SNET.NET                 204.60.0.2
   NS2.SNET.NET                 204.60.0.3

   ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

   Record last updated on 04-Aug-1999.
   Database last updated on 23-Mar-2001 22:42:43 EDT.
```

```
205.188.140.162
Pinging wads-d22.blue.aol.com [205.188.140.162] with 3
Reply from 205.188.140.162: bytes=32 time=184ms TTL=45

205.188.140.167
Pinging wads-d27.blue.aol.com [205.188.140.167] with 3
Reply from 205.188.140.167: bytes=32 time=178ms TTL=45

205.188.140.175
Pinging wads-d35.blue.aol.com [205.188.140.175] with 3
Reply from 205.188.140.175: bytes=32 time=180ms TTL=46

205.188.140.179
Pinging wads-d39.blue.aol.com [205.188.140.179]
```

```
Request timed out.

205.188.140.185
Pinging ads.web.aol.com [205.188.140.185] with 32 byte
Reply from 205.188.140.185: bytes=32 time=191ms TTL=50

206.41.20.6
Pinging 206.41.20.6 with 32 bytes of data: (MatchLogic Test server. )
Reply from 206.41.20.6: bytes=32 time=188ms TTL=245

209.94.119.131
Pinging www.tps.k12.ny.us [209.94.119.131]
Reply from 209.94.119.131: bytes=32 time=233ms TTL=117
```

The sources almost all look like identifiable, plausible hosts that we could have reason to communicate with.

What about these exceptions?:

```
-Feb 24 12:15:56 Deny TCP 206.67.50.46:5556 MY.NET.1.103:2049 in via
xl1
```

The source address responds to pings but does not resolve; Was this a response to the server? Port 5556 has been associated with "BO facil" and with HP/UX Remote Watch exploit: references:
http://advice.networkice.com/Advice/Exploits/Ports/5556/default.htm
http://www.sans.org/newlook/resources/IDFAQ/oddports.htm

```
-Feb 24 13:49:58 Deny TCP 205.219.162.10:113 MY.NET.1.110:4045 in via
xl1
Mar  3 18:31:11 Deny TCP 216.115.105.204:113 MY.NET.1.110:4045 in via
xl1
```

Are the communicating hosts mail servers? Yes.

```
Pinging web4704.mail.yahoo.com [216.115.105.204]
Reply from 216.115.105.204: bytes=32 time=250ms TTL=244
PING 205.219.162.10: 56 data bytes
64 bytes from mail1.javanet.com (205.219.162.10): icmp_seq=0. time=38.
ms
```

Does mailserver regularly issue port 113 queries? Yes, to authenticate smtp connnection attempts: "UNIX offers a service called ident or auth which will identify the user of a TCP connection. In the intended operation of this feature, when a user connects to a server, the server sends back a request to the ident service to discover the user's identity. However, it can also be used in a reverse way. If a server itself also has the ident feature turned on, when a user connects the the server, the user can query the identify of the service it is connecting to. This helps discover possible accounts that can be broken into."
http://advice.networkice.com/Advice/Underground/Hacking/Methods/Technical/Port_Scan/reverse_ident/default.htm

Does mailserver sometimes use NFS ports as ephemeral ports, as the web server does? They both use the same operating system. In order to find out I performed the following test, and yes it does:

```
firewall# tcpdump -i xl0 -c 1 -w mailtest.dump src host MY.NET.1.110
and src port \(4045 or 2049\) and dst port 113 &
[1] 6570
firewall# tcpdump: listening on xl0
firewall#
```

```
Key to tcpdump:
[hh:mm:sec.xxxxx][src addr:port]>[dest addr:port]:[flags
sequence][(data length)][window size][<maximum segment
size>][(fragmentation}]
```

```
%tcpdump -r mailtest.dump
19:21:20.899853 mail.mydomain.edu.lockd >
johnson.mail.mindspring.net.auth: S 1578461469:1578461469(0) win 16384
<mss 1460> (DF)

%tcpdump -n -r mailtest.dump
19:21:20.899853 MY.NET.1.110.4045 > 207.69.200.177.113: S
1578461469:1578461469(0) win 16384 <mss 1460> (DF)
```

5. Attack mechanism:

Is port 2049 being used as an ephemeral? As shown by the Sniffer traces at the top, when used as ephemeral source port by an outside host, the session succeeds. When used as ephemeral source port by an inside server, the session fails and is logged by the firewall.

The remote hosts eventually send resets when the several connection attempts to destination mynet.edu on port 2049 all fail.

6. Correlations:

Symptoms shown by the firewall logs are consistent with those shown by the Sniffer traces. These are ongoing and repeatable.

7. Evidence of active targeting:

As it turned out, (most of) the traffic in question was in response to session initiation by our server. Deliberate "targeting". Yes. An attack? No.

8. Severity:

(severity + criticality) - (network countermeasures + system countermeasures) = severity

Severity 3; If someone were to succeed in interference with NFS sessions it would be a problem for the servers in question.
Criticality 5; The servers in question are important to the institution.
Network Countermeasures 4; The firewall stopped the suspicious traffic from outside. The server subnet is partially protected from insider

attacks with router packet filters. Not all the routers' alleged packet filtering capabilities actually work.
System Countermeasures 3; The server admin has established an independent "back channel" upon which the servers communicate with the disk farm.

(3+5)-(4+3)=1

9. Defensive recommendation:

This was a false positive but the question of filtering these incoming port numbers is still a good one. Discuss with server admin person whether server allows customization of available ephemeral ports? Implement a more sophisticated firewall which keeps state on sessions initiated from within.

10. Multiple choice test question:

When might you suspect a TCP port number is being used ephemerally rather than for a particular service?

A. When it is below 1024.
B. When it occurs as the source port in a TCP SYN packet.
C. When the IP protocol field is set to 1.
D. When it occurs as the destination port in a TCP SYN packet.

Answer B.

```
*% grep 2049 /etc/services
nfsd            2049/tcp        nfs             # NFS server daemon
nfsd            2049/udp        nfs             # NFS server daemon
#shilp          2049/tcp
#shilp          2049/udp
% grep 4045 /etc/services
lockd           4045/udp                        # NFS lock
daemon/manager
lockd           4045/tcp
%
from ftp://ftp.isi.edu/in-notes/iana/assignments/port-numbers:
#                          <== NOTE Conflict on 2049 !
shilp           2049/tcp
shilp           2049/udp
nfs             2049/tcp   Network File System - Sun Microsystems
nfs             2049/udp   Network File System - Sun Microsystems
#               4043-4095  Unassigned
```

DETECT #4: TCP SCAN

```
(binette@home)

 Jan 26 09:18:15 cc1014244-a kernel: securityalert: tcp if=ef0 from
```

```
  204.33.212.44:2265 to 24.3.21.199 on unserved port 22
Jan 26 09:18:15 cc1014244-a kernel: securityalert: tcp if=ef0 from
  204.33.212.44:2268 to 24.3.21.199 on unserved port 42
Jan 26 09:18:15 cc1014244-a kernel: securityalert: tcp if=ef0 from
  204.33.212.44:2270 to 24.3.21.199 on unserved port 69
Jan 26 09:18:15 cc1014244-a kernel: securityalert: tcp if=ef0 from
  204.33.212.44:2272 to 24.3.21.199 on unserved port 80
Jan 26 09:18:15 cc1014244-a kernel: securityalert: tcp if=ef0 from
  204.33.212.44:2274 to 24.3.21.199 on unserved port 111
Jan 26 09:18:15 cc1014244-a kernel: securityalert: tcp if=ef0 from
  204.33.212.44:2276 to 24.3.21.199 on unserved port 143
Jan 26 09:18:15 cc1014244-a kernel: securityalert: tcp if=ef0 from
  204.33.212.44:2277 to 24.3.21.199 on unserved port 1080
Jan 26 09:18:15 cc1014244-a kernel: securityalert: tcp if=ef0 from
  204.33.212.44:2278 to 24.3.21.199 on unserved port 1745
Jan 26 09:18:15 cc1014244-a kernel: securityalert: tcp if=ef0 from
  204.33.212.44:2279 to 24.3.21.199 on unserved port 2301
Jan 26 09:18:16 cc1014244-a kernel: securityalert: tcp if=ef0 from
  204.33.212.44:2280 to 24.3.21.199 on unserved port 5190
Jan 26 09:18:16 cc1014244-a kernel: securityalert: tcp if=ef0 from
  204.33.212.44:2281 to 24.3.21.199 on unserved port 5191
Jan 26 09:18:16 cc1014244-a kernel: securityalert: tcp if=ef0 from
  204.33.212.44:2282 to 24.3.21.199 on unserved port 5192
Jan 26 09:18:16 cc1014244-a kernel: securityalert: tcp if=ef0 from
  204.33.212.44:2283 to 24.3.21.199 on unserved port 5193
Jan 26 09:18:16 cc1014244-a kernel: securityalert: tcp if=ef0 from
  204.33.212.44:2284 to 24.3.21.199 on unserved port 5631
Jan 26 09:18:16 cc1014244-a kernel: securityalert: tcp if=ef0 from
  204.33.212.44:2285 to 24.3.21.199 on unserved port 5632
Jan 26 09:18:16 cc1014244-a kernel: securityalert: tcp if=ef0 from
  204.33.212.44:2286 to 24.3.21.199 on unserved port 5800
```

1. Source of Trace:

http://www.sans.org/y2k/013001.htm

2. Detect was generated by:

A firewall log, possibly Gauntlet.

Suggestion about the "Detect was generated by" came from:
http://www.tis.com/support/unserved-port.html#solution

3. Probability the source address was spoofed:

This scan would not be likely to cause any harm in and of itself, unless replies were sent back to the perpetrator, in order to reveal information about the target host's operating system and suite of available services. Thus, it is unlikely that the source address was spoofed, unless a third party was interested in getting the owner of the source address in trouble for scanning. The source address is from the provider ICG NetAhead in San Jose, CA. and the destination address is part of the @Home network in Redwood City, CA., a provider of broadband internet service.

```
> whois -h whois.arin.net 204.33.212.44
ICG NetAhead, Inc. (NET-ICG-BLK-BLK8)
   532 Race St.
   San Jose, CA 95126
   US

   Netname: ICG-BLK-BLK8
   Netblock: 204.30.0.0 - 204.33.255.255



   Maintainer: ICGN

   Coordinator:
      Taylor, Stacy  (ST452-ARIN)  abuse@icgcom.com
      408-579-5000

   Domain System inverse mapping provided by:

   AS1.ICG.NET                  209.111.89.220
   AS2.ICG.NET                  209.111.89.221

   Record last updated on 16-Jan-2001.
   Database last updated on 24-Mar-2001 22:50:23 EDT.


> whois -h whois.arin.net \!NETBLK-MD-COMCAST-HWRD-1
@Home Network (NETBLK-MD-COMCAST-HWRD-1)
   425 Broadway
   Redwood City, CA 94063
   US

   Netname: MD-COMCAST-HWRD-1
   Netblock: 24.3.16.0 - 24.3.23.255

   Coordinator:
      Operations, Network  (HOME-NOC-ARIN)  noc-abuse@noc.home.net
      (650) 556-5599

   Record last updated on 30-Jul-1997.
   Database last updated on 24-Mar-2001 22:50:23 EDT.
```

4. Description of attack:

A TCP scan for known services.

5. Attack mechanism:

Sender attempts contact with 16 tcp source:dest port pairs within two seconds.

```
2265:22 ssh or pcanywhere
2268:42 host name server
2270:69 tftp
2272:80 http
2274:111 sunrpc
2276:143 imap
2277:1080 socks
2278:1745 remote-winsock
2279:2301 cpq-wbem
2280:5190 AOL AIM
2281:5191 AOL
2282:5192 AOL
2283:5193 AOL
2284:5631 pcANYWHEREdata
2285:5632 pcANYWHEREstat
2286:5800 vnc
```

6. Correlation:

None, really, but www.google.com search for ICG NetAhead turned up some interesting results. The search for provider ICG NetAhead yielded a spam complaint at:
http://www.tmisnet.com/~strads/spamhunt/benchmark/icg00827.html
I was unable at first to get name resolution on their own web-site, http://www.icgcomm.com. But later I did get it. They are a provider of dial-up and higher speed Internet services.

7. Evidence of active targeting:

Skipping of some source port numbers suggest source host may have been scanning others at the same time, or at least doing other work at the same time. This pattern appears more like a preliminary mapping exercise, which may result in more directed targeting in a follow-up attack.

8. Severity:

I suspect this is a home host on a broadband connection, and it seems that it's owner is watching out for questionable traffic, so I'm going to give it a criticality of 3 and a vulnerability of 3. I don't know about the host countermeasures, but there seems to be a firewall in the loop, whether on the network or on board, so I'll give it network countermeasure of 4 and host countermeasure of 3.

(3 + 3) - (4 + 3) = -1

9. Defensive recommendation:

The logs seem to imply that unneeded services are already closed down, but if not they should be, and where possible the services that are offered should not be offered anonymously but should require "good" passwords.

10. Multiple choice test question:

What do port numbers 22, 42, 111, and 5190 have in common?

A. They are all registered services.
B. They are owned by Microsoft.
C. They might be used to download music.
D. They're good for exchanging messages with your friends.

Answer: A.

DETECT #5: ATTEMPT TO CONNECT TO FIREWALL

```
209.140.138.230:3786 -> aaa.bbb.99.254:524
group of 4 - same 4 detects from 2 different logs with time =
16:58 and 21:58 (local vs. UTC time.)
211.53.59.98:2910 -> aaa.bbb.99.254:98

Attempted connections to firewall are dropped, as they should be.
```

[Date, Universal Time][Protocol,Action Message][Source IP, port, Interface] [Destination IP, port, interface][Relevant Rule Number]

(Bruce Lilly)

UTC 01/23/2001 21:58:12.768 - TCP connection dropped - Source:209.140.138.230, 3786, WAN - Destination:192.168.99.254, 524, LAN - - Rule 10
UTC 01/23/2001 21:58:13.496 - TCP connection dropped - Source:209.140.138.230, 3786, WAN - Destination:192.168.99.254, 524, LAN - - Rule 10
UTC 01/23/2001 21:58:14.208 - TCP connection dropped - Source:209.140.138.230, 3786, WAN - Destination:192.168.99.254, 524, LAN - - Rule 10
UTC 01/23/2001 21:58:14.928 - TCP connection dropped - Source:209.140.138.230, 3786, WAN - Destination:192.168.99.254, 524, LAN - - Rule 10

~~[Date, Local Time][Type of Notice][name of host][host type id]~~
[sequence number?][Date, Universal Time][fw address][pri,c,m ??]
[firewall message][source ip:port:interface][destination ip:port:interface][fw rule]

01-23-2001 16:58:13 Local0.Notice wall.blilly.com id=firewall sn=00D096BF23C5 time="2001-01-23 21:58:12 UTC" fw=192.168.99.254 pri=5 c=64 m=36 msg="TCP connection dropped" src=209.140.138.230:3786: WAN dst=192.168.99.254:524:LAN rule=10
01-23-2001 16:58:14 Local0.Notice wall.blilly.com id=firewal sn=00D096BF23C5 time="2001-01-23 21:58:13 UTC" fw=192.168.99.254 pri=5 c=64 m=36 msg="TCP connection dropped" src=209.140.138.230:3786: WAN dst=192.168.99.254:524:LAN rule=10
01-23-2001 16:58:15 Local0.Notice wall.blilly.com id=firewall sn=00D096BF23C5 time="2001-01-23 21:58:14 UTC" fw=192.168.99.254 pri=5 c=64 m=36 msg="TCP connection dropped" src=209.140.138.230:3786: WAN dst=192.168.99.254:524:LAN rule=10
01-23-2001 16:58:16 Local0.Notice wall.blilly.com id=firewall sn=00D096BF23C5 time="2001-01-23 21:58:14 UTC" fw=192.168.99.254 pri=5 c=64 m=36 msg="TCP connection dropped" src=209.140.138.230:3786: WAN dst=192.168.99.254:524:LAN rule=10

UTC 01/24/2001 01:23:13.576 - TCP connection dropped - Source:211.53.59.98, 2910, WAN - Destination:192.168.99.254, 98, LAN - - Rule 10

01-23-2001 20:23:14 Local0.Notice wall.blilly.com id=firewall sn=00D096BF23C5 time="2001-01-24 01:23:13 UTC" fw=192.168.99.254 pri=5 c=64 m=36 msg="TCP connection dropped" src=211.53.59.98:2910: WAN dst=192.168.99.254:98:LAN rule=10

1. Source of Trace:

http://www.sans.org/y2k/012501.htm

2. Detect was generated by:

A firewall. We have the same information from two different logs. Clearly they are firewall related, since the messages say so, but also because they indicate "TCP connection dropped". Perhaps the firewall program itself packages information in one way, but also forwards it to a syslog file where it is formatted somewhat differently. It looks like the site is in the Eastern time zone.

3. Probability the source address was spoofed:

At this time there is no response to a ping of the first source address.
Http://www.harvestinc.com/ shows .... "This site is currently under construction ... " .... Hosted by Network Solutions... So the owner of the source address is not quite up to speed? A defunct San Franciso dot.com? So, who is using these addresses? Is someone just having fun with them? This could be a case of a spoofed address.

The source of the attempt on port 98 comes from a Korean ISP - http://bora.net/eng/boranet/bora_ind.html
The address responds to a ping. Even though there's only one packet to go on, I'm going to say it was not spoofed.

```
> whois -h whois.arin.net \!NETBLK-HARVESTINC-WSTR
HARVEST TECHNOLOGIES, INC. (NETBLK-HARVESTINC-WSTR)
   164 TOWNSEND STREET, #2
   SAN FRANCISCO, CA 94107
   US

   Netname: HARVESTINC-WSTR
   Netblock: 209.140.138.0 - 209.140.138.255

   Coordinator:
      HAINES, BRENT  (BH448-ARIN)  BRENT@HARVESTINC.COM
      415.908.6806

   Record last updated on 19-Nov-1999.
   Database last updated on 24-Mar-2001 22:50:23 EDT.
```

http://www.harvestinc.com/
shows .... "This site is currently under construction ... " .... Hosted by Network Solutions...

```
# ENGLISH

IP Address          : 211.53.59.96-211.53.59.127
Connect ISP Name    : BORANET
Connect Date        : 20000229
Registration Date   : 20000310
Network Name        : IAKOREACOL39616D
```

```
[ Organization Information ]
Orgnization ID      : ORG102357
Name                : IA Korea Co., Ltd
State               : SEOUL
Address             : 190-1 Poi-Dong Kangnam-Gu
Zip Code            : 135-260

[ Admin Contact Information]
Name                : Donga Shim
Org Name            : IA Korea Co., Ltd
State               : SEOUL
Address             : 190-1 Poi-Dong Kangnam-Gu
Zip Code            : 135-260
Phone               : +82-2-578-3523
Fax                 : +82-2-578-3536
E-Mail              : b0039616@users.bora.net

[ Technical Contact Information ]
Name                : Donga Shim
Org Name            : IA Korea Co., Ltd
Address             : 190-1 Poi-Dong Kangnam-Gu
Zip Code            : 135-260
Phone               : +82-2-578-3523
Fax                 : +82-2-578-3536
E-Mail              : b0039616@users.bora.net
```

4. Description of attack:

Attempted connection to TCP ports 524 and 98 on the ip address of a firewall on 1/23/01 around UTC 21:58 - 16:58 EST.

Port 524 is not "registered" at
http://www.sans.org/newlook/resources/IDFAQ/oddports.htm
nor at http://advice.networkice.com/Advice/Exploits/Ports/default.htm
but it turned out that in
ftp://ftp.isi.edu/in-notes/iana/assignments/port-numbers:

```
ncp             524/tcp    NCP
ncp             524/udp    NCP
tacnews          98/tcp    TAC News
tacnews          98/udp    TAC News
```

While in
- http://advice.networkice.com/Advice/Exploits/Ports/98/default.htm
- Port 98 is LinuxConf.

TAC news:
http://auction2.eecs.umich.edu/index.html

5. Attack mechanism:

Alleged source address host from HARVEST TECHNOLOGIES, INC. sends TCP connection attempts to aaa.bbb.99.254:524.

Alleged source address 211.53.59.98 from Korea sends TCP to port 98 on aaa.bbb.99.254.

6. Correlations:

FROM:
http://lists.gnac.net/firewalls/mhonarc/firewalls.200101/msg00326.html

> "Since the new year, I've seen a marked rise in denied packets heading to port 524."

However, FROM:
http://archives.neohapsis.com/archives/incidents/2000-10/0172.html

> "Port 524 is registered as NCP. It is used by Netware 5.x server & clients (anything else?). These shouldn't be straying outside of the local networks though. Now that I've looked we've had a couple of connections to 524 the past few days. Nothing of note though (and no captures).”
>
> >>> <Suzanne.Hernandez@GUNTER.AF.MIL> 10/24/00 04:28PM >>>
> What's with the increased attempts on tcp port 524? These are coming from networks all over the place.... "

7. Evidence of active targeting:

I do not see evidence of active targeting due to the small number of packets involved - a four packet TCP attempt, followed by just a single packet TCP attempt. I’m inclined to think this is a part of a couple of larger scans for openings on the two ports involved.

8. Severity:

Since we are dealing with a firewall host target, I am assigning level 5 to the first two parameters. CVE does not list any known exposures regarding ports 524 or 98. Since the attacks were stopped and logged, I'm assigning 5 to system countermeasures as well. If the firewall itself is the network countermeasure, then perhaps it too gets a 5.

Criticality 5
Lethality 5
System Countermeasures 5
Network Countermeasures 5

(5 + 5) - (5 + 5) = 0

9. Defensive recommendation:

Double check to see that no services are open on the firewall machine that do no absolutely need to be. If possible and not already done, enable "stealth" on the firewall so that it does not so much as send

any kind of reply at all to any attempts to connect to it's own address.

10. Multiple choice test question:

What is the best way to manage IP connectivity to the firewall machine itself?

1. Allow connections to the firewall in from anywhere, to allow yourself to manage the machine wherever you are.

2. Set up a web server to distribute firewall statistics to the upper management.

3. Allow secure connections only, exclusively for firewall management, from one or a few trusted hosts only. Do not accept or respond to unwanted connection attempts.

4. Block all IP access to the firewall, use only the console for management.

Answer: 3

PART II - INTRUSION DETECTION TOPIC

INTRODUCTION TO THE SNORT TOOL

In the course book for Choosing and Justifying the Right Intrusion Detection and Vulnerability Analysis Tools by Alan Paller*, the Snort system by Martin Roesch was listed as not only one of the leaders in "market share", if you will, but also one of the top choices of those professionals polled at a previous SANS conference. Snort was also represented in the IDS track as one of the tools of choice for Intrusion Detection Analysts. Snort provides ongoing logging and alerting of attack conditions, as defined by the user through the configuration file and command line switches.

Capabilities

Snort offers the analyst log files showing "alerts", whose nature is defined by the ruleset invoked when Snort is run. It generates SCAN logs, highlighting instances of suspected network scans in particular. The packet capture functionality provides the analyst a view of the packets' actual headers and content, and can be designated to create logs of "out of specification" packets.

Snort's actions can be based upon protocol and port numbers, source or destination addresses, or information from the scan watching portion of the program. The scan functionality watches for patterns of activity, at various threshold levels. More complex triggers can be built up using the Berkeley Packet Filter syntax.

Compare to ...

Tcpdump is a program available for Unix and Windows hosts which captures and displays packets entering or leaving the host. With the interface in promiscuous mode, and connected to shared media or a mirroring switch port, other traffic can be captured as well. Typically is only the packet headers that can be captured. The analysis of the packets must be done by "hand" or by some other tool. Unlike tcpdump, which is a packet capture tool, Snort can do more than just packet capture, it can also perform either real-time or retrospective analysis of the captured traffic, in order to detect potentially threatening traffic entering or leaving a network. Tcpdump is very handy in that it can be run on the host and detect the traffic to and from that particular host, whether a server, firewall, or IDS box. In the absence of an official IDS, for instance, if firewall logs show suspicious activity from a particular address or port, and tcpdump is available on the firewall machine or on another machine with access to the same traffic flow, the analyst can very quickly, on the fly configure tcpdump to capture at least a handful of sample packets to allow the analyst to see what is going on.

Shadow is a store-and-analyze type IDS, often seen as a fitting companion to Snort. Often compromises are necessary in real-time configured systems, to achieve the required throughput. Snort can co-exist nicely with Shadow, in that Snort can read and write tcpdump data files, which are the basis of Shadow.

Sniffer – A commercial packet capture and protocol analysis package, Sniffer offers not only display but built in decode of many protocols from the link layer on up beyond the transport layer. If the analyst knows what she or he is looking for, Sniffer can find it. If not, the evidence will be there, but it won't be flagged necessarily as an error or detect, unless Sniffer sees it as some sort of network error. It's display and decoding of network traffic can be very helpful in Intrusion Detection but Sniffer's "Expert" symptoms and diagnosis is geared to network performance issues. Network Associates also offers other products in the security arena, however, such as a firewall product and virus detection software. The software is subject to license and support fees.

RealSecure - ISS/RealSecure is a commercially available security detection and response system meant to be applied to both networks and servers in a centrally managed environment. ISS RealSecure is what they would call an "enterprise" security system - a system of inter-related network, server, and operating system monitors and protection schemes. It appears that it would be appropriate for a large organization in which a security team is responsible for many sites or resources, and where financial resources are available for deploying such a system. The system is not purchased or licensed as a whole, but each module is licensed separately. Snort seems to be more suitable for the condition where there are personnel at each site who could configure and run the IDS locally, and respond to it's output. ISS/RealSecure features a client-server architecture in which network sensors, server sensors, and operating system log monitors all report to a management client. The client can be a stand-alone module or can be integrated with HP OpenView or Tivoli management systems. The suite also features hooks into the Lucent Managed Firewall management server, for display of RealSecure Alerts on the LMF manager screen, and hooks into the CheckPoint Firewall-1 product for automated response to attack alerts. The operating system audit feature can detect suspicious server conditions, even if they do not come as a result of network traffic, but from a console or modem user. The server component allows the configuration of a false services feature, which the product literature denies is a honeypot strategy, but kind of looks like one. The policy enforcement features allow scanning of web connections, remote file service connections, and email to see that company policy is not being violated by employees.

http://documents.iss.net/literature/RealSecure/rs5_0faq.pdf

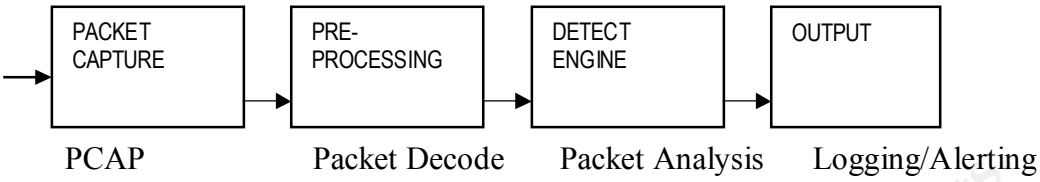
Why Choose Snort

In a survey of professionals a SANS conference, Snort was highly regarded in both effectiveness and users' preference. An interesting observation from this writer's recent experience and observations of the FreeBSD community, is that the support available from the free software community, at least for users who are capable of actively participating in the solution (no small feat in some cases), is potentially faster and more effective than the type of support one can get by contract from a vendor's help desk. Just recently, on the other hand, contract support for Snort has been offered by Silicon Defense. Following the snort-users mailing list, one realizes that the enhancement and development of Snort is a non-stop process, with updates and improvements available daily.

I am interested in following up my SANS training and certification activities by installing and learning use of the Snort tool for these reasons: 1. We can do so without going through a budget process. (We are a small, state .edu - get it?) 2. Since implementing a firewall, which *is* playing an important role for us as a packet filter and bandwidth limiter, we have learned the limitations of firewall logs as far as giving us a complete picture of the traffic which may seek to exploit our vulnerabilities. The firewall only guards against those exploits we know about and can write rules for.

My efforts since we implemented a firewall, over the last year or so, to effectively respond to daily firewall log output has shown me the value of some at least semi-automatic tool to scan for known exploits. The reasons are threefold. 1. The sheer volume of information generated each day. We haven't the luxury of assigning one or more full time employees to just look at security logs all day every day. 2. In the case of the firewall, we only log rule violations of the rules we have set up regarding particular hosts and subnets, and against some of the simplest and best known exploits. The most granular we get in firewall rules is to allow or deny traffic based on IP addresses and/or UDP or TCP service ports. And, being that .edu, as mentioned, we do follow the practice so reviled among security professionals, of running our firewall in default-accept mode, denying only those particular signatures we can reliably describe via firewall rules. As such, running an IDS separate from the firewall (although perhaps on the same machine if possible), should help us identify exploits, scans, reconnaissance or what-have you which are sure to be coming in along with the 'allowed' traffic. 3. From the firewall logs we get information for each time a rule is violated: the rule that triggered the deny; the source and destination addresses and ports, and the date and time. We can only recognize a scan by scanning the logs ourselves, either visually or by using grep, etc. Once I get some experience with Snort, Snortsnarf and other related tools, I expect to increase the amount of review I can do of our security situation, respond to at least some of the exploits that come in each day, and possibly have the time to do some other work as well.

Snort Architecture



In Mr. Roesch's presentation at SANS New Orleans 2001, he pointed out that while Snort is called a lightweight IDS, it is acknowledged to be not so lightweight with all of it's options and a full ruleset applied. Plug-in modules are available to customize the function of each of the stages beyond the packet capture stage, to make Snort into something other than, or more than, a plain IDS. It is possible to custom configure not only the ruleset to be used with Snort, but also the balance between fast response and granular output. Snort was originally designed with relatively small, low budget networks in mind, and though it has grown up over time, so have the speed and complexity of network traffic. If traffic is heavy and the analyst expects Snort to keep up, it may be necessary to reduce the granularity of the real-time reporting to just the most critical parameters. Some analysts like to deploy Snort on multiple stations to get a view of network traffic from various perspectives, as well.

Rules

The simplest of Snort rules involve just the ip address space, direction, protocol, and ports. Option keywords allow the user to specify more in detail just what type of HTTP messages, for instance, are of concern. Content pattern matching can be used, even if the content may not be found always at a predictable data or header offset in the packet. Rules can be designed using Berkeley Packet Filter syntax, or using command line input. The rules are applied in a modular fashion, so that it is more efficient to ask for Snort to detect more than one condition, say, from a given source address.

Actions

The analyst may configure Snort to either ignore certain traffic, letting it pass with no comment, or to log the detect, and/or to issue an alert on certain detects. The alerts could be in the form of an entry into a separate "alert" file, passed to an SMP WinPopup message, or to some other process for actively alerting the NOC or sysadmin. Though it is a controversial topic in the IDS community, someone is sure to add functionality to Snort to allow it to dynamically configure firewall action to directly stop the questionable traffic without waiting for the intervention of an operator. This is a controversial topic because of the possibility of over-reaction, and of acting on "false positives" – that is, traffic that looks like an attack to the IDS but is actually legitimate.

Resources, References and Web Sites

Advanced Intrusion Detection – Snort Style by Martin Roesch; SANS Security New Orleans; The SANS Institute; 2001

Snort - Lightweight Intrusion Detection for Networks; by Martin Roesch; found at http://www.snort.org/

Snort Architecture Part 1: The Packet Decoder; by Martin Roesch; from <snort-users@lists.sourceforge.net>

http://www.silicondefense.com/techsupport/
http://www.nai.com/
http://www.snort.org

Information about PCAP:
http://src.openresources.com/debian/src/libs/HTML/S/libpcap_0.4a6.orig%20libpcap-0.4a6.orig%20pcap-int.h.html
http://www.isi.edu/nsnam/ns/doc-stable/node473.html
http://ee.lbl.gov/
ftp://ftp.lbl.gov/libpcap.tar.Z

*Allen Paller quoted in Roesch: Advanced Intrusion Detection – Snort Style

Note: While researching this topic I was reminded of another free security related product called Nessus. There is an important distinction to keep in mind, in that it is a remote security *scanner* as distinct from an intrusion *detection* system. Nessus' role is to *create* attack traffic to test *your* network for susceptibility to known exposures.

http://www.nessus.org/intro.html

PART III "ANALYZE THIS"

Certification candidates were supplied with three groups of files, collected between November 25, 2000 and January 9, 2001. The files include Snort* Scan reports, Snort Alert Reports, and Snort 'OOS' Out of Specification reports. Upon inspection it turned out that the files as numbered were not in order of date so it was necessary, to begin to get a handle on what was there, and to attempt correlation among them, to get them in some order.

It must be noted that the dates attached to the Alert and Scan files only are the day the file was written, which means that the data contained there-in is from the day before. It also should be noted that there are some duplicates, such as SnortS29 and S32.txt, and SnortS11, 13, 14.txt. Not every day is covered, and not every day that's covered is covered by all three types of files, each file may or may not represent a whole day. In other words, there are gaps in the data.

*Snort by Martin Roesch

OVERVIEW

Using the Snortsnarf tool to list out the alert types contained in the collection of files, I arrived at the following: (Note that Snortsnarf's assertion that the earliest alert came on 01/01 and the latest on 12/31 does not take into account that the files cover a period from November '00 through January of '01. Also note that Snortsnarf did not enumerate the quantities, nor the source/destination addresses successfully. This could possibly be due to difference between the file format expected by Snortsnarf, and that supplied from "MY.NET". In any case, I used this information only as a guide to the most frequent alerts in the data set.)

http://myserver/Snortsnarf/snfout.sansAlert.txt/

SnortSnarf start page

All Snort signatures

SnortSnarf v011601.1

Earliest alert at **00:00:46**.876474 *on 01/01*
Latest alert at **23:45:47**.026613 *on 12/31*

Signature (click for definition)	# Alerts	# Sources	# Destinations	Detail link
STATDX UDP attack	1	1	1	Summary
Happy 99 Virus	1	1	1	Summary
site exec - Possible wu-ftpd exploit - GIAC000623	2	1	1	Summary
SITE EXEC - Possible wu-ftpd exploit - GIAC000623	2	1	1	Summary
Probable NMAP fingerprint attempt	8	1	1	Summary
External RPC call	67	1	1	Summary
Back Orifice	77	1	1	Summary
TCP SMTP Source Port traffic	100	1	1	Summary
Broadcast Ping to subnet 70	154	1	1	Summary
connect to 515 from inside	159	1	1	Summary
SUNRPC highport access!	204	1	1	Summary
SMB Name Wildcard	518	1	1	Summary
Russia Dynamo - SANS Flash 28-jul-00	546	1	1	Summary
NMAP TCP ping!	567	1	1	Summary
SNMP public access	591	1	1	Summary
Queso fingerprint	714	1	1	Summary
Null scan!	834	1	1	Summary
Attempted Sun RPC high port access	2248	1	1	Summary
WinGate 1080 Attempt	2299	1	1	Summary
Watchlist 000222 NET-NCFC	2416	1	1	Summary
connect to 515 from outside	4951	1	1	Summary
Tiny Fragments - Possible Hostile Activity	5356	1	1	Summary
DNS udp DoS attack described on unisog	16146	1	1	Summary
SYN-FIN scan!	51193	1	1	Summary
Watchlist 000220 IL-ISDNNET-990517	109077	1	1	Summary

SnortSnarf brought to you courtesy of Silicon Defense
Authors: Jim Hoagland and Stuart Staniford
See also the Snort Page by Marty Roesch
Page generated at Thu Mar 15 11:25:17 2001

Inside exploits.

As for the overview of the files, it was also interesting to look at the issue of suspicious activity originating within the network or seeming to, according to the alleged source addresses. This is a summary of the most often reported MY.NET source addresses, and how many hits each accounted for.

MY.NET. HOST SOURCE ADDRESSES

SOURCE MY.NET - GLOBAL		SOURCE MY.NET - ALERTS		SOURCE MY.NET - SCANS		SOURCE MY.NET - OOS	
58763	MY.NET.100.230	24246	MY.NET.214.166	58763	MY.NET.100.230	5417	MY.NET.217.150
54674	MY.NET.213.186	14219	MY.NET.253.24	54674	MY.NET.213.186	3903	MY.NET.217.158
35149	MY.NET.202.94	12514	MY.NET.217.150	35149	MY.NET.202.94	1133	MY.NET.217.126
33734	MY.NET.217.94	11515	MY.NET.100.230	33734	MY.NET.217.94	1064	MY.NET.217.182
32406	MY.NET.98.200	9999	MY.NET.217.150	32406	MY.NET.98.200	784	MY.NET.219.126
31840	MY.NET.253.24	9824	MY.NET.217.158	31840	MY.NET.253.24	56	MY.NET.98.152
28377	MY.NET.217.150	6869	MY.NET.213.186	27825	MY.NET.214.166	16	MY.NET.97.36
27825	MY.NET.214.166	6865	MY.NET.202.94	25190	MY.NET.218.158	12	MY.NET.97.148
25191	MY.NET.218.158	6094	MY.NET.217.158	22960	MY.NET.217.150	10	MY.NET.98.163
24246	MY.NET.214.166	5805	MY.NET.100.230	18478	MY.NET.218.130	10	MY.NET.219.2
18478	MY.NET.218.130	4988	MY.NET.218.130	16587	MY.NET.156.110	9	MY.NET.98.185
16587	MY.NET.156.110	4373	MY.NET.219.126	14164	MY.NET.201.50	7	MY.NET.97.137
14219	MY.NET.253.24	3875	MY.NET.253.24	13858	MY.NET.217.106	7	MY.NET.207.254
14164	MY.NET.201.50	3159	MY.NET.156.110	13684	MY.NET.140.21	7	MY.NET.202.46
13858	MY.NET.217.106	3049	MY.NET.217.182	11581	MY.NET.71.38	6	MY.NET.98.190
13684	MY.NET.140.21	2758	MY.NET.219.126	10850	MY.NET.201.46	6	MY.NET.98.140
12514	MY.NET.217.150	2605	MY.NET.97.154	10543	MY.NET.98.177	6	MY.NET.97.70
12023	MY.NET.217.158	2208	MY.NET.217.126	10289	MY.NET.217.142	6	MY.NET.222.62
11581	MY.NET.71.38	2157	MY.NET.60.8	10044	MY.NET.97.93	5	MY.NET.98.122
11515	MY.NET.100.230	2005	MY.NET.217.230	9577	MY.NET.206.186	4	MY.NET.97.187
10850	MY.NET.201.46	1879	MY.NET.1.3	9415	MY.NET.212.150	3	MY.NET.98.156
10543	MY.NET.98.177	1836	MY.NET.201.46	9101	MY.NET.97.234	3	MY.NET.217.190
10289	MY.NET.217.142	1757	MY.NET.217.106	9055	MY.NET.217.58	3	MY.NET.211.130
10044	MY.NET.97.93	1748	MY.NET.201.50	8120	MY.NET.217.158	3	MY.NET.181.131
9999	MY.NET.217.150	1623	MY.NET.217.182	7972	MY.NET.97.148	2	MY.NET.98.202
9824	MY.NET.217.158	1536	MY.NET.202.6	7544	MY.NET.212.34	2	MY.NET.98.157
9577	MY.NET.206.186	1519	MY.NET.1.5	7410	MY.NET.1.3	2	MY.NET.97.227
9415	MY.NET.212.150	1440	MY.NET.97.165	7175	MY.NET.97.208	2	MY.NET.97.195

9101	MY.NET.97.234	1415	MY.NET.217.126	7148	MY.NET.98.238	2	MY.NET.227.86

This is a summary of the most often reported remote host source addresses represented in the data set, and how many times each appeared.

REMOTE HOST SOURCE ADDRESSES

	SOURCE REMOTE – GLOBAL		SOURCE REMOTE - ALERTS		SOURCE REMOTE - SCANS		SOURCE REMOTE - OOS
33502	24.180.134.156	17604	211.34.40.1:53	33502	24.180.134.156	14919	195.56.182.206
29530	212.187.94.162	9878	195.56.182.206:21	29530	212.187.94.162	12473	194.234.48.26
29528	24.4.196.167	9307	212.179.79.2:38318	29528	24.4.196.167	5907	147.8.182.157
22545	212.64.74.169	8565	194.234.48.26:21	22545	212.64.74.169	3596	194.204.224.131
22005	24.191.63.215	5581	212.179.27.111:2310	22005	24.191.63.215	2614	139.130.61.206
21920	62.158.93.109	5078	212.179.79.2:40227	21920	62.158.93.109	2133	63.204.152.253
18744	24.29.40.11	4951	outside	18744	24.29.40.11	2088	200.194.102.99
17604	211.34.40.1:53	4198	212.179.95.5:4260	16874	216.6.8.25	1973	194.197.170.7
16874	216.6.8.25	4096	147.8.182.157:109	15042	133.1.36.184	1030	193.253.202.9
15042	133.1.36.184	3879	212.179.27.111:2047	13647	207.29.192.114	893	132.68.37.141
14919	195.56.182.206	3254	212.179.27.111:1929	12710	64.167.160.235	148	24.113.198.51
13647	207.29.192.114	3052	194.204.224.131:109	10284	216.17.174.253	60	63.78.39.192
12710	64.167.160.235	2826	212.179.27.111:1854	9283	216.99.200.242	44	63.124.243.34
12473	194.234.48.26	2816	212.179.27.111:1792	9262	66.20.207.21	40	141.30.228.36
10284	216.17.174.253	2540	212.179.79.2:31835	8998	64.5.206.84	35	141.30.228.199
10003	147.8.182.157	2341	212.179.27.111:2316	8514	152.163.206.134	31	63.229.92.11
9878	195.56.182.206:21	2028	212.179.27.111:2253	8252	24.3.0.36	31	141.30.228.43
9307	212.179.79.2:38318	1951	139.130.61.206:109	8098	24.226.126.93	18	204.42.254.5
9283	216.99.200.242	1908	212.179.79.2:12128	7955	140.128.123.5	15	141.30.228.178
9262	66.20.207.21	1905	212.179.79.2:31012	7141	62.227.243.120	14	141.30.228.175
8998	64.5.206.84	1897	212.179.27.111:2317	6307	131.161.49.140	13	141.30.228.115
8565	194.234.48.26:21	1856	212.179.27.111:2196	5754	193.89.241.53	12	141.30.228.182
8514	152.163.206.134	1823	212.179.27.111:1778	5219	24.3.0.37	9	130.239.129.109
8252	24.3.0.36	1790	200.194.102.99:21	4990	213.51.67.218	8	24.112.150.20
8098	24.226.126.93	1722	212.179.27.111:2306	4455	193.159.98.85	8	128.46.156.117
7955	140.128.123.5	1713	212.179.27.111:2311	4429	151.196.73.156	7	64.80.118.241
7141	62.227.243.120	1632	212.179.27.111:2176	4167	134.192.143.247	7	63.252.94.96
6651	194.204.224.131	1581	212.179.27.111:2315	4096	147.8.182.157	7	62.29.16.82
6307	131.161.49.140	1580	194.197.170.7:9055	3651	146.203.28.14	7	141.30.228.58

Looking at some of the specific alerts .....

Watchlist 000220

GENERAL DESCRIPTION

It seems that any traffic coming from address space 212.179.x.y triggers a Watchlist 000220 alert. Traffic is of many types, including Napster, Audiogalaxy, Gaming, scans, etc. It will be most advantageous to depart from the Alert files and look more closely at the Scan and OOS, to try and determine the character of this traffic.

There are 100 MY.NET destination addresses associated with Watchlist 000220. These are the top 10 destination *ips* of alerts associated with Watchlist 000220:

```
> cat 000220.dstips.sorted | sort -rn
37604  MY.NET.201.222
25182  MY.NET.220.126
9309  MY.NET.225.234
5181  MY.NET.202.94
5080  MY.NET.229.114
4445  MY.NET.228.214
2288  MY.NET.202.30
1912  MY.NET.201.130
1517  MY.NET.130.187
1438  MY.NET.217.138
```

These are some of the destination *ports* of alerts associated with Watchlist 000220:

```
> cat 000220.dstports.sorted | sort -rn
hits  port
37767 6688 Napster
29194 6699 Napster
9525 4876
9315 4967
4191 1525
1914 6346 gnutella
1517 2209
1388 443 secure web
1221 4078
1062 41033 possible Audiogalaxy
1054 23     telnet
 960 7000
...
 386 4285
 349 41038 Audiogalaxy?
 269 4846
 178 41022 Audiogalaxy?
 124 4394
  99 25     mail
```

```
  57 4951
 ...
```

These are some of the 466 different source *ports* associated with Watchlist 000220.

```
> cat 000220.srcports.sorted | sort -rn
hits  port
9307 38318
5581 2310
5078 40227
4198 4260
3879 2047
3254 1929
2826 1854
2816 1792
2540 31835
2375 2317
...
1581 2315
1517 1      tcpmux
1463 44160
...
```

These are some of the 46 source *ips* assocated with Watchlist 000220. Note that they are all from 212.179.0.0.

```
> cat 000220.srcips.sorted | sort -rn
hits   source address
48786  212.179.79.2
39015  212.179.27.111
4563  212.179.95.5
2353  212.179.77.20
1517  212.179.44.105
1387  212.179.42.102
1221  212.179.38.135
1054  212.179.58.12
1002  212.179.45.241
 926  212.179.56.5
```

Who is the "attacker"?

In this whois excerpt, we see that the name of the Watchlist alert "[**] Watchlist 000220 IL-ISDNNET-990517 [**]" includes the 'netname' of the address space, "IL-ISDNNET-990517". Further whois requests shows that this address space is subdivided among various owners.

```
inetnum:     212.179.0.0 - 212.179.255.255
netname:     IL-ISDNNET-990517
descr:       PROVIDER
country:     IL
admin-c:     NP469-RIPE
tech-c:      OR214-RIPE
tech-c:      TP1233-RIPE
tech-c:      ZV140-RIPE
tech-c:      ES4966-RIPE
```

```
status:      ALLOCATED PA
mnt-by:      RIPE-NCC-HM-MNT
changed:     hostmaster@ripe.net 19990517
changed:     hostmaster@ripe.net 20000406
source:      RIPE

inetnum:     212.179.58.0 - 212.179.58.255
netname:     NV-PICTUREVISION
descr:       network
country:     IL
admin-c:     NP469-RIPE
tech-c:      NP469-RIPE
status:      ASSIGNED PA
notify:      hostmaster@isdn.net.il
changed:     hostmaster@isdn.net.il 20000229
source:      RIPE

***

inetnum:     212.179.79.0 - 212.179.79.63
netname:     CREOSCITEX
descr:       CREOSCITEX-SIFRA
country:     IL
admin-c:     ZV140-RIPE
tech-c:      NP469-RIPE
status:      ASSIGNED PA
notify:      hostmaster@isdn.net.il
changed:     hostmaster@isdn.net.il 20001109
source:      RIPE
```

TIME/DATE

-scan log activity takes place on Dec 17, 20, 21, 31 Jan 1, 2, 3, 12
-alert log; The first appearance of this alert in this sample of log files was Nov 24, 2000. The last appearance was Jan 16, 2001.
There were 35 days on which hits were noted.

WHAT IS GOING ON?

Many types of traffic, possibly not dangerous, or maybe so, are represented.

The top destination ip address in the Watchlist section is MY.NET.201.222. Looking at activity of MY.NET.201.222 in it's totality, we find a few isolated attacks not associated with Watchlist (this are single, isolated hits).

```
> grep  MY.NET.201.222 ../ALERT/*A50.txt
01/05-07:24:51.861113  [**] Null scan! [**] 62.31.28.201:18245 ->
MY.NET.201.222:21504
> grep MY.NET.201.222 ../ALERT/*A38.txt
01/10-12:17:58.718025  [**] SYN-FIN scan! [**] 195.56.182.206:21 ->
MY.NET.201.222:21
> grep MY.NET.201.222 ../ALERT/*A45.txt
```

```
01/07-04:04:15.644885  [**] SYN-FIN scan! [**] 211.34.40.1:53 ->
MY.NET.201.222:53
```

but we find over 37,000 hits associated with the watchlist, all on one day, all associated with 212.179.27.111,

```
> grep MY.NET.201.222 ../ALERT/*A51.txt | grep -c 212.179.27.111
37604
> grep -c MY.NET.201.222 ../ALERT/*A51.txt
37604
```

which turns out to be a Napster application, by the use of port 6688.

```
excerpt
>  grep MY.NET.201.222 ../ALERT/*A51.txt | head
01/04-02:54:06.872039  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.27.111:1778 -> MY.NET.201.222:6688
01/04-02:54:07.917555  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.27.111:1778 -> MY.NET.201.222:6688
01/04-02:54:08.343293  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.27.111:1778 -> MY.NET.201.222:6688
```

The host with the most hits does not seem to indicate a dangerous attack, although use of Napster could possibly constitute a.) A violation of policy, or b.) a hogging of internet bandwidth.

KNOWN EXPLOIT?

-Surprisingly, I am not having success locating any info on just who or what is "Watchlist". It does not seem to relate so much to the attack per se as to the source address. Maybe it's referenced in the Snort rules database?

CORRELATION?

Either due to missing files, or due to a lack of actual OOS packets, there were no hits of 212.179 in the OOS files.

```
> grep -c 212.179 ../OOS/*.txt
no hits
>
```

Are there instances of 212.179 in the scan logs?
Yes, but some of them represent MY.NET.212.179 ... do any represent 212.179.0.0? There aren't that many, so it would be useful to simply grep for 212.179 in the scan files.

COLLATERAL REVELATIONS

Given an alert as generalized as a Watchlist alert, one finds many different types of traffic caught under this umbrella. One collateral item revealed by grepping for "212.179" in the scan files is a possible ftp server at MY.NET.212.179, which was the destination of a SYNFIN packet with reflexive port 21 on Dec 25 at 15:33:22 from source ip

133.1.36.184. The source ip 133.1.36.184 shows up in the scan logs over 15,000 times, all on Dec. 25.

There are reported scans from inside MY.NET going outward to IL-ISDNNET-990517, the Watchlist subject network. In particular MY.NET.217.250 is scanning regularly to other addresses as well as to IL-ISDNNET-990517 and should be looked at for possible compromise or misuse.

Below is shown UDP port 28800 traffic associated with MY.NET.217.250. It looks as if it is used in gaming applications. In http://archives.neohapsis.com/archives/incidents/2000-08/0256.html, it is alleged that "28000 udp is used in the popular online game Starsiege Tribes" … I don't want to draw conclusions about 28800 from that information alone. At http://edge.fireplug.net/disc1/000003df.htm there is "What I'm trying to get working is some on-line gaming for my flight sim "habit". According to the Zone's tech personnel, what I need to do is the following: Allow an initial outbound connection to TCP port 6667 and subsequent connections on TCP ports 28800-29000". Further, at http://www.withgate.com/help/Tested%20Hardware%20and%20Software.htm , there is yet another reference to UDP 28800 as a gaming port. (As an aside, there is a collegiate CIS web server on port 28800 at http://www.victor.cc.ca.us:28800/, but that is at 207.233.102.2, and it would be TCP.)

Also shown is UDP traffic to various addresses at high port numbers. This looks like a scan but for what I can't tell at this point.

This shows in which scan files MY.NET.217.250 shows up:

```
> sh grepscript MY.NET.217.250
../SCAN1/SnortS11.txt:28
../SCAN1/SnortS13.txt:28
../SCAN1/SnortS14.txt:28
../SCAN1/SnortS17.txt:111
../SCAN1/SnortS18.txt:1
../SCAN1/SnortS21.txt:1
../SCAN1/SnortS22.txt:34
../SCAN1/SnortS25.txt:1
../SCAN2/SnortS34.txt:1
../SCAN2/SnortS42.txt:960
>
```

And a couple of examples of the scans reported:

```
> grep MY.NET.217.250 ../SCAN2/SnortS42.txt | more
Jan  8 17:16:54 MY.NET.217.250:1138 -> 207.46.172.63:28845 SYN **S*****
Jan  8 17:16:54 MY.NET.217.250:28800 -> 208.61.176.121:28800 UDP
Jan  8 17:16:55 MY.NET.217.250:28800 -> 213.134.10.184:28800 UDP
Jan  8 17:16:55 MY.NET.217.250:28800 -> 172.134.255.125:28800 UDP
Jan  8 17:16:55 MY.NET.217.250:28800 -> 63.17.39.173:28800 UDP
Jan  8 17:16:56 MY.NET.217.250:28800 -> 24.43.129.55:1060 UDP
Jan  8 17:16:56 MY.NET.217.250:28800 -> 216.23.50.133:28800 UDP
Jan  8 17:16:57 MY.NET.217.250:28800 -> 213.30.47.43:28800 UDP

> grep MY.NET.217.250 ../SCAN1/SnortS14.txt | more
```

```
Dec 21 00:05:59 MY.NET.217.250:4659 -> 216.23.151.2:61824 UDP
Dec 21 00:05:59 MY.NET.217.250:4688 -> 212.162.240.49:37016 UDP
Dec 21 00:05:59 MY.NET.217.250:4690 -> 212.162.240.23:27031 UDP
Dec 21 00:05:59 MY.NET.217.250:4691 -> 212.162.240.10:27026 UDP
Dec 21 00:05:59 MY.NET.217.250:4696 -> 212.122.148.112:27020 UDP
Dec 21 00:05:59 MY.NET.217.250:4712 -> 209.249.117.95:27101 UDP
Dec 21 00:05:59 MY.NET.217.250:4682 -> 213.207.20.11:27020 UDP
```

Another interesting pair of hits was

```
../SCAN1/SnortS17.txt:Dec 20 03:14:34 MY.NET.98.130:0 ->
212.179.163.1:0 UDP
../SCAN1/SnortS17.txt:Dec 20 03:25:45 MY.NET.98.130:21090 ->
212.179.163.1:2000 SYN **S*****
```

I notice the source and destination ports of 0 on the first packet, and then the attempted TCP connection to port 2000 on the second. Port 0 attracts attention because it would not be used in any normally created packets. I happen to be aware that a certain manufacturer's routers and LAN equipment uses telnet port 2000 for management, and the host address of "1" is often used for routers or gateways, so I wonder whether this was some type of mapping/connection attempt to a networking device?

In another case of a "collateral" hit,

```
> grep MY.NET.98.130 ../SCAN1/SnortS22.txt | more
Dec 31 02:02:17 24.3.0.36:53 -> MY.NET.98.130:1693 UDP
Dec 31 02:02:17 24.3.0.36:53 -> MY.NET.98.130:1694 UDP
Dec 31 02:02:18 24.3.0.36:53 -> MY.NET.98.130:1695 UDP
Dec 31 02:02:18 24.3.0.36:53 -> MY.NET.98.130:1697 UDP
Dec 31 02:02:18 24.3.0.36:53 -> MY.NET.98.130:1698 UDP
-snip-
```

On Dec. 31 MY.NET.98.130 may have stimulated this response by hitting 24.3.0.36 with a port 53 exploit, or possibly this is a case of MY.NET.98.130 having been spoofed by someone else in a dns ddos event. On Dec. 20, however, MY.NET.98.130 was doing some scanning of it's own, much of it to dest port 2000 at many different addresses.

Getting back to the Watchlist 000220, here's a case - Alert log 12 and Scan log 10, from the same day, both have 212.179 represented, but not the same traffic. In the alert file, we could be seeing a port scan agains MY.NET.203.46, and so would want to check that host for compromise, but they are separated in time, and did not show up in the scan files, so we don't know that this was not some legitimate connection.

```
> grep 000220 ../ALERT/SnortA12.txt
12/17-04:08:12.637287  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.79.2:27171 -> MY.NET.203.46:4913
12/17-05:53:15.249672  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.8.164:4691 -> MY.NET.203.46:1068
12/17-06:18:46.209017  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.8.164:4691 -> MY.NET.203.46:1190
12/17-06:29:32.323774  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.8.164:4691 -> MY.NET.203.46:1284
```

```
> grep 212.179 ../SCAN1/SnortS10.txt
Dec 17 22:10:50 MY.NET.206.186:1191 -> 212.179.145.29:27960 UDP
>
```

Here's another case, where we can see traffic between the same two hosts. At 1:48:56 a packet is sent from the Watchlist host to the MY.NET host and port that sent a SYN during the same second .... we do not see any evidence that this resulted in a completed connection, however.

```
> grep 212.179 ../SCAN2/SnortS35.txt
Jan 11 01:48:56 MY.NET.217.78:2493 -> 212.179.37.93:4598 SYN **S*****
> grep 212.179.37.93 ../ALERT/SnortA34.txt
01/11-00:19:34.025374  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.37.93:4598 -> MY.NET.217.78:1284
01/11-00:19:34.025707  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.37.93:4598 -> MY.NET.217.78:1284
01/11-00:59:45.260515  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.37.93:4598 -> MY.NET.217.78:1956
01/11-01:10:30.144564  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.37.93:4598 -> MY.NET.217.78:2139
01/11-01:29:03.756867  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.37.93:4598 -> MY.NET.217.78:2318
01/11-01:29:03.760638  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.37.93:4598 -> MY.NET.217.78:2318
01/11-01:48:56.937340  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.37.93:4598 -> MY.NET.217.78:2493
01/11-02:09:23.394070  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.37.93:4598 -> MY.NET.217.78:2691
```

And another example, where the numbers 212.179 show up in the host portion of the ip address, causing false matches, but also there is outgoing traffic from MY.NET to 212.179.x.y in the scan file, involving different hosts and ports than the ones in the alert log. Do we know whether MY.NET.253.43 is a mail server (port 25)?

```
> grep 212.179. ../ALERT/SnortA43.txt (excerpts)
01/08-02:58:40.659468  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.45.241:3958 -> MY.NET.217.138:4852
01/08-02:58:40.870185  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.45.241:3958 -> MY.NET.217.138:4852
01/08-14:08:15.717182  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.7.36:1128 -> MY.NET.253.43:25
01/08-14:08:21.501728  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.7.36:1128 -> MY.NET.253.43:25
> grep 212.179 ../SCAN2/SnortS42.txt (excepts)
Jan  8 06:44:49 MY.NET.217.94:4197 -> 195.29.212.179:7778 UDP
Jan  8 07:09:39 MY.NET.217.94:4657 -> 195.29.212.179:7778 UDP
Jan  8 14:44:27 MY.NET.97.34:6112 -> 212.179.187.127:6112 UDP
>
```

In the excerpt below we see repeated packets to MY.NET.253.43:25, and elsewhere, in file S24 for instance, we see evidence of MY.NET.253.43 behaving like a possible mail server, sending port 25 traffic to other

addresses and sending auth requests to other addresses, so we don't know that this represents a problem.

```
01/08-14:08:40.177445  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.7.36:1128 -> MY.NET.253.43:25
01/08-14:08:41.250630  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.7.36:1128 -> MY.NET.253.43:25
01/08-14:08:44.776320  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.7.36:1128 -> MY.NET.253.43:25
01/08-14:08:46.401897  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.7.36:1128 -> MY.NET.253.43:25
-snip-
```

Attempt at more internal correlation reveals the difficulty of having gaps in the log files, where data is not available from the same day among the 3 types of logs.

```
-Watchlist 000220 appears in /Snortalerts/sansAlert2.txt on Nov 29
> cat 000220.srcips.sorted
  36  212.179.63.10
 403  212.179.79.2
neither address is found in the scan files ...
> grep 212.179.63.10 ../../SCAN2/*.*
> grep 212.179.79.2 ../../SCAN2/*.*
>   grep 212.179.63.10 ../../SCAN1/*.*
> grep 212.179.79.2 ../../SCAN1/*.*
nor in OOS directory
-Nov 29 does not appear in any of the SCAN files either
```

Note: The only SYNFIN found with both 212.179 and SYNFIN was a reflexive attack on MY.NET.212.179:

```
> grep 212.179 *.delim | grep  SYNFIN
SnortS34.txt.delim:Dec 25
15:33:22,133.1.36.184,21,MY.NET.212.179,21,SYNFIN,**SF****,
```

Whether we have collaterally captured part of an ftp scan, or whether MY.NET.212.179 is actually an ftp server, is something to look at.

```
> grep MY.NET.212.179 ../SCAN1/*.txt
../SCAN1/SnortS11.txt:Dec 21 01:13:49 62.227.243.120:2558 ->
MY.NET.212.179:21 SYN **S*****
../SCAN1/SnortS13.txt:Dec 21 01:13:49 62.227.243.120:2558 ->
MY.NET.212.179:21 SYN **S*****
../SCAN1/SnortS14.txt:Dec 21 01:13:49 62.227.243.120:2558 ->
MY.NET.212.179:21 SYN **S*****
../SCAN1/SnortS18.txt:Jan  3 21:18:45 212.64.74.169:3045 ->
MY.NET.212.179:21 SYN **S*****
../SCAN1/SnortS18.txt:Jan  3 21:18:48 212.64.74.169:3045 ->
MY.NET.212.179:21 SYN **S*****
```

Because Watchlist 000220 was the most frequent alert listed in the alert files, I looked at them closely. I would have to consider carefully the usefulness of the "Watchlist" idea, when applied to an entire address range. This could constitute one of those DOS attacks against the analyst's time. We should have alerts on suspicious activity, but not necessarily broad-brush alerts on large blocks of addresses.

CORRELLATION OUTSIDE

-Correlation: Bayerkohler practical: "The most noteworthy incident was an incoming ftp session from Israel, a country that triggered the Watchlist 000220 alert, on 9/3" - from
http://www.sans.org/y2k/practical/Marc Bayerkohler GCIA.html

-how about correlation at SANS? There are reports from earlier months of the same alert, for instance in
FROM *http://www.sans.org/y2k/032200-1700.htm*
FROM *http://www.sans.org/y2k/032500-2200.htm*
(see also Watchlist 000222 entries)
FROM *http://www.sans.org/y2k/052000.htm*

NOTE: FROM: *http://www.sans.org/y2k/practical/Robert Currie.doc*
"Napster itself is known to have some security issues (CAN-2000-0281, CAN-2000-0412 at http://cve.mitre.org/)."

SYN-FIN

TIME/DATE

All of the Scan and OOS log files supplied, and most of the Alert files, showed evidence of SYN-FIN activity.

GENERAL DESCRIPTION

A SYN-FIN scan is the sending of traffic to hosts using TCP packets which have both SYN and FIN flags set in the TCP header. In properly assembled TCP headers, SYN is used to initiate the 3-way handshake that begins a TCP session, and FIN is used to initiate the 2-way teardown of that session. SYN-FIN packets can be used, for instance, for mapping a network's hosts, by taking note of how any given host responds (or not) to the SYN-FIN packet. By the type of response, the mapper may be able to guess something about the operating system running on the host. Judging from the use of certain well known ports in the SYN-FIN packets, it may be possible to learn more information about a host based on how particular services respond to these malformed packets.

DESCRIPTION

SOURCE ADDRESS

No SYN-FINs were sent *from* MY.NET

```
> grep -c MY.NET SYN-FIN.srcips.sorted
        0
```

Here is a partial listing of the source addresses associated with SYN-FINs.

```
> cat SYN-FIN.srcips.sorted | sort -rn
hits    address
17604  211.34.40.1
9878  195.56.182.206
8565  194.234.48.26
4096  147.8.182.157
3052  194.204.224.131
1951  139.130.61.206
1790  200.194.102.99
1580  194.197.170.7
1242  63.204.152.253
 706  193.253.202.9
 630  132.68.37.141
  44  64.161.240.254
  25  63.229.92.11
   4  63.11.25.117
```

```
   2  63.252.94.211
   2  63.252.92.239
   2  24.68.49.13
   1  64.196.23.118
   1  64.196.112.164
```

SOURCE PORTS

Here is a partial listing of source ports in SYN-FIN packets.

```
>cat SYN-FIN.srcports.sorted | sort -rn
hits  port
21579 21   ftp
18863 53   dns
9099 109   pop2
1580 9055
  18 32808
  11 25    smtp
   6 110   pop3
   4 4
   2 6688 Napster
   1 90
   1 64190
   1 64159
```

DESTINATION ADDRESS

27067 of the 65025 possible MY.NET addresses were at some time during the period the target of at least one SYN-FIN:

```
> grep -c MY.NET SYN-FIN.dstips.sorted
              27067
```

There was no one, or few, addresses that attracted a large block of them, but rather the destination address list was rather diffuse.

```
> cat SYN-FIN.dstips.sorted | sort -rn | head
hits  address
  19  MY.NET.253.112
   8  MY.NET.21.15
   7  MY.NET.5.125
   7  MY.NET.11.212
   6  MY.NET.7.184
   -snip-
```

DESTINATION PORTS

The favorite scans seem to be for FTP, DNS, and POP2 services.

```
> cat SYN-FIN.dstports.sorted | sort -rn
21604 21
18863 53
9099 109
1580 9055
  18 259
  11 25
   6 110
   3 80
   2 3713
   1 6970
-snip-
```

Looking closer at examples of actual SYN-FIN traffic...using 211.34.40.1, we find that address was missed by the OOS files that were supplied us, and by the scan files as well.

This address is from a high school in Korea.

```
> whois -h whois.nic.or.kr 211.34.40.1
# ENGLISH

IP Address         : 211.34.40.0-211.34.40.127
Connect ISP Name   : PUBNET
Connect Date       : 19991002
Registration Date  : 19991022
Network Name       : YOUSUBOOYOUNG-GHS

[ Organization Information ]
Orgnization ID     : ORG83057
Name               : YousuBooyoungGirl`sHighSchool
State              : CHONNAM
Address            : 657-1 Ansan-Dong Yousu-City
Zip Code           : 555-050

[ Admin Contact Information]
Name               : Hajin Choi
Org Name           : YousuBooyoungGorl`sHighSchool
State              : CHONNAM
Address            : 657-1 Ansan-Dong Yousu-City
Zip Code           : 555-050
Phone              : 062-606-0322
E-Mail             : jeonnam3@soback.kornet.net

[ Technical Contact Information ]
Name               : Hajin Choi
Org Name           : YousuBooyoungGorl`sHighSchool
Address            : 657-1 Ansan-Dong Yousu-City
Zip Code           : 555-050
Phone              : 062-606-0322
E-Mail             : jeonnam3@soback.kornet.net
```

In the alert files we see that a DNS scan was going on on Jan 7, 2001. With the DNS BIND exploits that were common at the beginning of the year, this may have been a case of scanning for hosts with BIND running. The attacker might expect from these packets either a RST or

RST-ACK, or possibly a "service not available" message, or no response at all, any of which could reveal the presence or absence of an interesting target.

```
-snip-
../ALERT/SnortA45.txt:01/07-03:47:16.287373  [**] SYN-FIN scan! [**]
211.34.40.1:53 -> MY.NET.1.254:53
../ALERT/SnortA45.txt:01/07-03:47:16.305149  [**] SYN-FIN scan! [**]
211.34.40.1:53 -> MY.NET.1.255:53
../ALERT/SnortA45.txt:01/07-03:47:16.322211  [**] SYN-FIN scan! [**]
211.34.40.1:53 -> MY.NET.2.1:53
../ALERT/SnortA45.txt:01/07-03:47:16.341990  [**] SYN-FIN scan! [**]
211.34.40.1:53 -> MY.NET.2.2:53
-snip-
```

Now looking at 195.56.182.206 for representation in the logs:

```
../OOS/OOSche37.txt:14919 Jan 10
nothing in the scan files
../ALERT/SnortA38.txt:10157 Jan 10
```

The address is represented nearly 15,000 times in the Out of Spec logs, not at all in the scan logs, but over 10,000 alerts on January 10.

The address is located in Hungary.

```
route:        195.56.0.0/16
descr:        DataNet Telecommunications Ltd.
descr:        Public Internet Access Provider
descr:        Hungary
origin:       AS3340
notify:       rzsolt@datanet.hu
mnt-by:       AS3340-MNT
changed:      kajtar@datanet.hu 20000225
source:       RIPE
```

From the OOSche37.txt Out-Of-Spec file:
We can see in these three identical port-reflexive frames, going to 3 different destination addresses from the Hungarian source, that the sequence numbers, ack numbers, ID numbers are all fixed, which are characteristics of crafted packets (as is of course the SF flag itself.) The point here, as above, would be to elicit some sort of response, or none, to the SF scan. The character of the response would tell the scanner something about the target host.

```
[date-time] [source ip:port] -> [destination ip:port]
[protocol] [time-to-live] [type of service] [frame IP header ID number]
[tcp flags] [tcp sequence number] [tcp ack number] [tcp window size]
[payload]
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/10-12:00:58.038135 195.56.182.206:21 -> MY.NET.1.2:21
TCP TTL:28 TOS:0x0 ID:39426
```

```
**SF**** Seq: 0x42616ED   Ack: 0x1294489   Win: 0x404
00 00 00 00 00 00                                ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/10-12:00:58.038200 195.56.182.206:21 -> MY.NET.1.3:21
TCP TTL:28 TOS:0x0 ID:39426
**SF**** Seq: 0x42616ED   Ack: 0x1294489   Win: 0x404
00 00 00 00 00 00                                ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/10-12:00:58.038268 195.56.182.206:21 -> MY.NET.1.4:21
TCP TTL:28 TOS:0x0 ID:39426
**SF**** Seq: 0x42616ED   Ack: 0x1294489   Win: 0x404
00 00 00 00 00 00                                ......

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

It looks as if 195.56.182.206 “stealthily” scanned its way through over 10,000 MY.NET addresses looking for vulnerabilities and/or mapping information at port 21, FTP. Here is a tiny sample of the log entries.

```
>   grep -c 195.56.182.206 ../ALERT/*38.txt
10157
> grep 195.56.182.206 ../ALERT/*38.txt | head
01/10-12:17:10.289740  [**] spp_portscan: PORTSCAN DETECTED from
195.56.182.206 (STEALTH) [**]
01/10-12:00:54.263825  [**] SYN-FIN scan! [**] 195.56.182.206:21 ->
MY.NET.1.2:21
01/10-12:00:54.263882  [**] SYN-FIN scan! [**] 195.56.182.206:21 ->
MY.NET.1.3:21
```

NORMAL OR CRAFTED PACKETS?

SYN-FIN packets are by definition crafted, or the result of some technical error. They would never be found in legitimate traffic.

CORRELATION?

internal

Could these logs have come from different hosts, accounting for the time difference, or might one have been lagging behind the other in processing the information? The source and destination addresses and ports are the same, but the times are about 4 seconds apart.

```
> grep 195.56.182.206 ../ALERT/*38.txt | grep "MY.NET.1\.2:21"
01/10-12:00:54.263825  [**] SYN-FIN scan! [**] 195.56.182.206:21 ->
MY.NET.1.2:21
> grep 195.56.182.206 ../OOS/*37.txt | grep "MY.NET.1\.2:21"
01/10-12:00:58.038135 195.56.182.206:21 -> MY.NET.1.2:21
>
```

SANS CORELLATION:

Two days later the 211.34.40.1:53 address was reported scanning another network:
FROM: http://www.sans.org/y2k/011701-1500.htm

```
"(Security@auckland)
On Tue 09 Jan 2001 at 08:18 (UTC) we detected a scan of tcp-53 ports in
part of our network. This incident appears to have originated from
211.34.40.1. Either some third party has compromised 211.34.40.1 and is
now using it to attack others sites or a legitimate users of
211.34.40.1 are engaging in practices that are not condoned under most
company or ISP acceptable use policies.
Sample logs, times are UTC + 1300, GPS synchronized:

 09 Jan 01 21:17:52      tcp     211.34.40.1.53     ?>
130.216.2.35.53    F
 09 Jan 01 21:17:52      tcp     211.34.40.1.53     ?>
130.216.2.36.53    F
 09 Jan 01 21:17:52      tcp     211.34.40.1.53     ?>
130.216.2.37.53    F
 09 Jan 01 21:17:52      tcp     211.34.40.1.53     ?>
130.216.2.38.53    F
 09 Jan 01 21:17:52      tcp     211.34.40.1.53     ?>
130.216.2.39.53    F
 09 Jan 01 21:17:52      tcp     211.34.40.1.53     ?>
130.216.2.40.53    F
 09 Jan 01 21:17:52      tcp     211.34.40.1.53     ?>
130.216.2.41.53    F
 09 Jan 01 21:17:52      tcp     211.34.40.1.53     ?>
130.216.2.42.53    F
 09 Jan 01 21:17:52      tcp     211.34.40.1.53     ?>
130.216.2.43.53    F
 09 Jan 01 21:17:52      tcp     211.34.40.1.53     ?>
130.216.2.44.53    F

 Source: 211.34.40.1
 Ports: tcp-53
 Incident type: Network_scan
 re-distribute: yes
 timezone: UTC + 1300
 reply: no
 Time: Tue 09 Jan 2001 at 08:18 (UTC) "
```

ACTIVE TARGETING?

We can say that these events represent moderately active targeting in that the traffic was sent to these addresses, with the hope of gaining some information or opportunity as a result. MY.NET was

not alone in receiving the attack from 211.34.40.1, for instance, so I cannot say that MY.NET has been exclusively targeted by these events.

INTENT

Reconnaissance, Compromising hosts, finding an opening to exploit.

METHOD

Scanning the network address space with SYNFIN packets to FTP, DNS, POP2 and other services or ports.

SEVERITY?

We do not know whether protections are in place on MY.NET to stop such traffic at the perimeter, but there certainly should be. Successful reconnaissance is designed to yield information which can later be used in a more serious compromise.

More Information for SYN-FIN
REFERENCES:
FROM: *http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=BIND*
FROM: *http://lists.sourceforge.net/archives//snort-users/2000-July/000062.html*

By the way, while looking for SYN-FIN packets I also found many packets like these illegally flagged packets including RESETS. These, like other crafted packets, could be an attempt to either reconnoiter or to bring certain hosts down:

```
> grep R ../OOS/OOSche20.txt
**SFR*AU Seq: 0x489B002   Ack: 0x1030D   Win: 0x5018
**SFR*AU Seq: 0x49A9002   Ack: 0xBF030D   Win: 0x5018
21SFRP*U Seq: 0x477   Ack: 0x96899DEF   Win: 0x5010
21SFR*** Seq: 0x104E9   Ack: 0x847187DA   Win: 0x5010
2*SFRP** Seq: 0x4E9A3BF   Ack: 0x90A0   Win: 0x5010
2*SFR*A* Seq: 0x7704E9   Ack: 0xBF5F9875   Win: 0x5010
*1SFRP** Seq: 0xD004E9   Ack: 0xE56CA33B   Win: 0x5010
21SFRPAU Seq: 0x1D271EAA   Ack: 0x116ED387   Win: 0x5018
*1SFR*** Seq: 0x1D70971E   Ack: 0x6116E   Win: 0x5010
21SFRPA* Seq: 0x4EA1839   Ack: 0xCB1A1   Win: 0x5010
```

DNS UDP DOS

TIME/DATE
This incident is reflected in alert file SnortA47.txt and was logged on Jan 6, 2001. It lasted from 18:30:02 to 20:00:01.

GENERAL DESCRIPTION

DNS ddos from unisog: Apparently unisog is a discussion list hosted by SANS and there were one or more DNS DDOS incidents which were the subject of some discussion, and so a Snort alert was designed to respond to that. There was a notable one on January 11, '01 which is referenced in SANS discussions, but these detects are prior to that

date. The general idea is that spoofed dns queries with the source address of the intended victim are sent to nameservers, so that dns responses, far too many of them, are sent to the victim address. In this case MY.NET.1.3, 4, and 5 appear to be used as reflectors or amplifiers, if you will, to send traffic to the address from which the requests appear to be coming, at about 3 frames per second. Entries *in* *http://www.sans.org/y2k/010801-1900.htm* and following, show that others are also involved, so that the victim host receives many more than 3 frames per second as a result of this ddos.

DESCRIPTION

Looking at the source ips, we see all of the noted traffic was from 209.67.50.0/24; mostly from 209.67.50.203.

SOURCE ADDRESS

```
> cat ../../DNS/DNS.srcips.sorted
   1  209.67.50.220
   1  209.67.50.246
   1  209.67.50.33
   2  209.67.50.209
   2  209.67.50.241
   3  209.67.50.85
   4  209.67.50.253
16132  209.67.50.203  here's the big one.
```

Are any or all of the following actually dns servers? A check of their appearances in the scan files, at least for the last 3, below, suggests that they are, due to the number of outgoing messages from them at source port 53 to other hosts.

```
> cat ../../DNS/DNS.dstips.sorted
   2  MY.NET.1.9
   6  MY.NET.1.10
   6  MY.NET.1.8
5331  MY.NET.1.5
5390  MY.NET.1.4
5411  MY.NET.1.3
```

There are over 16 thousand hits to port 53 reported in this category.

```
> cat ../../DNS/DNS.dstports.sorted
   2 42394
   6 50936
   6 58191
16132 53
```

Just a sample of sourceports detected: no great large counts of any one port. There are 14 source port 53 instances, and 14 destination ports that are not 53.

```
> head ../../DNS/DNS.srcports.sorted
```

```
   1 10005
   1 10006
   1 10007
   1 10009
   1 10012
   1 10015
   1 10016
   1 10020
   1 10026
   1 10030
> tail ../../DNS/DNS.srcports.sorted
   5 12247
   5 1494
   5 15252
   5 17306
   5 18282
   5 22611
   5 6195
   5 8546
   6 17932
  14 53
>
```

Here is just a sample of the log entries

```
grep MY.NET.1.3 ../ALERT/*A47.txt
01/06-18:30:02.600073  [**] DNS udp DoS attack described on unisog [**]
209.67.50.203:9247 -> MY.NET.1.3:53
01/06-18:30:05.030330  [**] DNS udp DoS attack described on unisog [**]
209.67.50.203:10165 -> MY.NET.1.3:53
-snip-
01/06-18:30:12.854623  [**] DNS udp DoS attack described on unisog [**]
209.67.50.203:15929 -> MY.NET.1.5:53
01/06-18:30:14.735326  [**] DNS udp DoS attack described on unisog [**]
209.67.50.203:10319 -> MY.NET.1.5:53
```

In the meantime MY.NET.1.10 is receiving dns responses from Exodus.Net (just these four.)

```
> grep 209.67.50.253  ../../DNS/DNS.grep
01/06-18:46:33.109670  & DNS udp DoS attack described on unisog &
209.67.50.253:53 & MY.NET.1.10:58191
01/06-19:10:45.735923  & DNS udp DoS attack described on unisog &
209.67.50.253:53 & MY.NET.1.10:58191
01/06-19:10:45.737415  & DNS udp DoS attack described on unisog &
209.67.50.253:53 & MY.NET.1.10:58191
01/06-19:56:12.023216  & DNS udp DoS attack described on unisog &
209.67.50.253:53 & MY.NET.1.10:58191
>
```

CORRELATION?

internal:

There were 16,000 hits to this alert on January 6.

address

```
seeking 209.67.50
no same day entries in scan and alert files for this

no entries in OOS files for this address
../ALERT/SnortA47.txt:16146 jan 6
../SCAN1/SnortS12.txt:5 jan 2
../SCAN1/SnortS27.txt:3 jan 12
```

All this next display really tells us is that there is traffic to hosts on the Exodus.Net:

```
> grep 209.67.50 ../SCAN1/SnortS12.txt ../SCAN1/SnortS27.txt
../SCAN1/SnortS12.txt:Jan  2 02:26:21 MY.NET.100.230:32780 ->
209.67.50.241:53 UDP
../SCAN1/SnortS12.txt:Jan  2 02:26:21 MY.NET.100.230:32780 ->
209.67.50.220:53 UDP
../SCAN1/SnortS12.txt:Jan  2 07:59:23 MY.NET.100.230:32780 ->
209.67.50.86:53 UDP
../SCAN1/SnortS12.txt:Jan  2 07:59:23 MY.NET.100.230:32780 ->
209.67.50.85:53 UDP
../SCAN1/SnortS12.txt:Jan  2 19:12:45 MY.NET.100.230:32780 ->
209.67.50.254:53 UDP
../SCAN1/SnortS27.txt:Jan 12 07:20:30 MY.NET.253.52:56187 ->
209.67.50.203:25 SYN **S*****
../SCAN1/SnortS27.txt:Jan 12 07:21:21 MY.NET.253.52:56187 ->
209.67.50.203:25 SYN **S*****
```

SANS

FROM: http://www.sans.org/y2k/010801-1900.htm

```
"...I have received quite a few emails about a some DNS queries. I dont
have a lot of information to go on, but the ip address is
209.67.50.203. If you are seeing anything please let me know."
```

FROM: http://www.sans.org/y2k/010901.htm

```
"(Gene Runion)

I too have been curious. Here is the run down from my end.
I still have network 209.67.50.0 blocked at our three routers with
different internet access. I am still seeing about the same number of
denies. (I am no longer logging the denies so I am assuming that they
are still all coming from 209.67.50.203 (or spoofed) and that they are
DNS requests).

We first discovered this because we have one DNS server with the newer
bind that is configured not resolve names for hosts that are not in our
domain when the request comes from the internet. We were logging such
requests which resulted in an abnormally large log file which got our
attention. Then I noticed a steady stream of DNS requests from
209.67.50.203 to our five DNS servers. At that point I decided
something was wrong, other than someone trying to use our DNS server,
and blocked that network. We then sent an email to abuse@exodus.net.
```

Then I received a telephone call from from them who said they were not the source but the victim and they, for the last 72 hours or so, have been trying to put an end to it. This all took place on 4 Jan from ~3-9pm est. We have had no further correspondence with them.

Late Friday afternoon I checked with a sister organization who, after checking their logs, saw the same behavior. Saturday, after checking to see if this traffic was still present, I sent a message to you.

That's it from my end."

other?

While I was unable to find any more information about the following web page, it is referred to as a "banlist" and so it was interesting to find the address responsible for the largest number of hits in this alert category:

FROM:
http://www.google.com/search?hl=en&lr=&safe=off&q=209.67.50&btnG=Google+

Search
Untitled
... hub 974764008 0!* hub 974764008 *!194.20.201.151 hub 974764008 *!**209.67.50.*** hub
974764008 blah!* hub 974764008 *!216.218.134.* hub 974764008 *!212.25.168.129 ...
imperialfleet.com/opennap-banlist.php - 10k - Cached - Similar pages

REFERENCES for DDOS

http://www.sans.org/y2k/011101.htm
http://www.theorygroup.com/Archive/Unisog/2001/msg00055.html

> *whois -h whois.arin.net 209.67.50.0*

Exodus Communications Inc. (NETBLK-ECI-5)
1605 Wyatt Dr.
Santa Clara, CA 95054
US

Netname: ECI-5
Netblock: 209.67.0.0 - 209.67.255.255
Maintainer: ECI

Coordinator:
Center, Network Control (NOC44-ARIN) CompServ@Exodus.net
(888) 239-6387 (FAX) (888) 239-6387

```
   Domain System inverse mapping provided by:

   NS.EXODUS.NET                206.79.230.10
   NS2.EXODUS.NET               207.82.198.150

   * Rwhois reassignment information for this block is available at:
   * rwhois.exodus.net 4321

   ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
```

http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=DNS+DDOS provides a list of potential exposures that could victimize DNS servers.

PORT 515

TIME The first occurrence in alert files - Nov 24 '00 02:45:xx (4 frames in 1 sec)
Last - Jan 18 '01 -14:28:22 - 23:28:06 - 134 frames
Max occurrences - Dec 15 '00 - 00:24:36 - 0:55:52 - clusters (of several frames at once), several seconds apart -
18 days in all

GENERAL DESCRIPTION

Traffic to destination port 515. The largest 515 scan attacking MY.NET. came on December 15 possibly from a host at the University of Michigan. Without a tcpdump or other representation of the packets, it is not possible to distinguish what shape the packet headers or payload might be in. There were no examples of port 515 traffic among the OOS out-of-specification files. There are 4951 instances of incoming events, and 159 outgoing. Some of the traffic is aimed at one destination address at a time, indicating what may be a pure scan for printer services or vulnerabilities. Other parts of the evidence shows outside hosts making multiple attempts to connect to specific target addresses. It is unknown whether they have gotten responses in the past from these hosts, or are just trying different techniques of scanning.

DESCRIPTION

-INFORMATION FROM ALERT FILES BY WAY OF ALERTSCRIPT

SOURCE ADDRESS

The following are source addresses associated with port 515 crossing the Snort detector, and their frequencies. As you can see some of the source addresses are on MY.NET indicating that there may be suspicious activity taking place locally. In fact the address MY.NET.70.38 shows some interesting activity.

```
   1  128.61.36.117
   1  172.161.186.125
   1  207.173.179.18
```

```
   1  24.160.143.196
   1  24.4.196.167
   1  MY.NET.163.17
   1  MY.NET.179.78
   1  MY.NET.219.122
   1  MY.NET.219.194
   1  MY.NET.60.16
   2  MY.NET.99.244
   3  MY.NET.253.12
   3  MY.NET.60.38
   4  62.46.70.175
   7  192.118.36.9
   9  MY.NET.98.151
 137  MY.NET.70.38 - 1/18/01
1273  216.119.15.88 - 12/20/2000
1426  209.217.166.69 - 12/16/2000 in part
2236  141.211.176.99 - 12/15/2000
```

SOURCE PORTS

Various ports from the 1000's to the 4000's generally incrementing upwards but with some exceptions (frames arriving out of order?)

DESTINATION ADDRESS

Destinations are both on and off MY.NET.

Below is just a sample of the destination addresses. Many entries are associated with just one or a few instances.

```
9  216.181.129.185
210  MY.NET.214.166
259  MY.NET.130.86
403  MY.NET.99.104
405  MY.NET.100.209
2  129.155.192.99
3  128.8.3.106
3  212.187.65.135
3  64.23.4.67
```

DESTINATION PORTS

There were 5110 packets to port 515.

Port 515 is used for unix printer services.

NORMAL OR CRAFTED PACKETS?

Without higher fidelity information, it is difficult to reach any firm conclusions about this issue. Whereas some of the related exploits have the goal of gaining root access to affected servers, it would be a reasonable guess that the source addresses are correct and that the

packets are "normal" enough to be functional, at least. The scan samples of the packets from 209.217.166.69 the TCP flags are SYN's ... suggesting that these are may be normally formed SYN packets meant to elicit a response back to the source, or even to open a session to the printer and cause a denial of service or other attack by printing!

KNOWN EXPLOIT

There are a number of known exploits involving port 515. Weaknesses in various platforms' printing processes can be exploited. From GIAC: "Alert: Increased probes to TCP port 515 Posted: 14:00 November 20, 2000, ... Local and remote users can send string-formatting operators to the printer daemon to corrupt the daemon's execution, potentially gaining root access." Another reference listed in http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0839 would cause a buffer overflow by including a large number of lpd options send to the lpd port. Another, mentioned in http://www.cert.org/current/current activity.html#LPRng - *Vulnerability Note VU#382365*

```
            Printer  515/tcp
            IN-2001-01, Widespread Compromises via "ramen" Toolkit
   Vulnerability Note VU#382365, LPRng can pass user-supplied input as
a
   format string parameter to syslog() calls
```

FROM: http://www.kb.cert.org/vuls/id/382365

```
"A popular replacement software package to the BSD lpd printing service
called LPRng contains at least one software defect
 known as a "format string vulnerability" which may allow remote users
to execute arbitrary code on vulnerable systems. The
 privileges of such code will probably be root-level."
```

CORRELATION?

The address 209.217.166.69 was found in two Scan files:

Search for 209.217.166.69 in scan files: SnortS3.txt:689
SnortS10.txt:9

Search for 141.211.176.99 in scan files: none found.

```
Sample: > grep 209.217.166.69 ...SnortS3.txt
Dec 16 21:09:41 209.217.166.69:3105 -> MY.NET.60.129:515 SYN **S*****
Dec 16 21:09:41 209.217.166.69:3135 -> MY.NET.60.159:515 SYN **S*****
```

```
Dec 16 21:09:47 209.217.166.69:1116 -> MY.NET.68.73:515 SYN **S*****
Dec 16 21:09:47 209.217.166.69:1133 -> MY.NET.68.90:515 SYN **S*****
Dec 16 21:09:47 209.217.166.69:1136 -> MY.NET.68.93:515 SYN **S*****
 Sample: > grep 209.217.166.69 ...SnortA11.txt | grep 21:09
12/16-21:09:41.155108  [**] connect to 515 from outside [**]
209.217.166.69:3105 -> MY.NET.60.129:515
12/16-21:09:41.167007  [**] connect to 515 from outside [**]
209.217.166.69:3135 -> MY.NET.60.159:515
12/16-21:09:47.133724  [**] connect to 515 from outside [**]
209.217.166.69:1116 -> MY.NET.68.73:515
12/16-21:09:47.148808  [**] connect to 515 from outside [**]
209.217.166.69:1133 -> MY.NET.68.90:515
12/16-21:09:47.150408  [**] connect to 515 from outside [**]
209.217.166.69:1136 -> MY.NET.68.93:515
```

ACTIVE TARGETING?

I would shy away from active targeting in this case, because we know that 209.217.166.69, at least, was scanning more than just MY.NET:

On *http://www.freerepublic.com/forum/a3a3d939c4d23.htm*
the question was asked on Dec. 17, 2000,

> "Has anyone else been scanned while on FreeRepublic. I just get scanned from IP 209.217.166.69 on the Verio,Inc network. This is the 7th time in a week that this has happened while on the FreeRepublic web site. I have never been scanned while on any other site. If you don't have firewall software I suggest you get some. I use ZoneLabs ZoneAlarm. It's free and it works. If you don't know much about computers or TCP/IP just install it with the defaults. There web site is zonelabs.com"

INTENT

Possibly to gain root access to the server, or to cause a denial of service. Port 515 scans have specific known exploits that can be taken advantage of if found, but the scanning is still preliminary, to try and map where these services might be, by eliciting the sorts of responses mentioned above regarding dns and ftp.

METHOD

Scanning for responsive servers on TCP well known port 515, possibly with the intent to exploit those services if found.

SEVERITY

The potential for harm is great, if the perpetrator achieves root access to servers, and can then install programs, access passwords, and so forth. There is unlikely to be any legitimate reason for those outside the institution to need to connect to printing services, and so it would be advisable to restrict traffic to destination port 515 at the perimeter.

Correlation:
The OOS files have nothing to offer in this case, there are no ":515" in OOS files.

There are many 515 references in the scan files. These are from 209.217.166.69, which is seen to be performing an ftp scan the next day.

```
> tail scan515.tmp
sample
...
Dec 16 21:12:04 209.217.166.69:4852 -> MY.NET.253.125:515 SYN **S*****
Dec 16 21:12:04 209.217.166.69:4853 -> MY.NET.253.126:515 SYN **S*****
Dec 16 21:12:04 209.217.166.69:4857 -> MY.NET.253.130:515 SYN **S*****
Dec 16 21:12:04 209.217.166.69:4859 -> MY.NET.253.132:515 SYN **S*****
...snip...
Dec 16 21:12:05 209.217.166.69:1250 -> MY.NET.254.243:515 SYN **S*****
...
The day after it's 515 scan on Dec 16:
> grep  209.217.166.69 ../SCAN1/SnortS10.txt
Dec 17 03:24:08 209.217.166.69:1608 -> MY.NET.139.136:21 SYN **S*****
Dec 17 03:24:08 209.217.166.69:1612 -> MY.NET.139.231:21 SYN **S*****
Dec 17 03:24:08 209.217.166.69:1614 -> MY.NET.140.29:21 SYN **S*****
Dec 17 03:24:09 209.217.166.69:1625 -> MY.NET.145.153:21 SYN **S*****
Dec 17 03:24:09 209.217.166.69:1635 -> MY.NET.145.174:21 SYN **S*****
Dec 17 03:24:09 209.217.166.69:1637 -> MY.NET.145.178:21 SYN **S*****
Dec 17 03:24:09 209.217.166.69:1648 -> MY.NET.156.29:21 SYN **S*****
Dec 17 03:24:10 209.217.166.69:1666 -> MY.NET.181.112:21 SYN **S*****
Dec 17 03:24:10 209.217.166.69:1670 -> MY.NET.214.166:21 SYN **S*****
(Looking for an open ftp and/or reconnaissance scanning)
```

REFERENCES

```
From Solaris:
> grep 515 /etc/services
printer         515/tcp         spooler         # line printer spooler
>
```

```
NT / Windows 2000 TCP/IP Printing Service DoS
Vulnerability

 credit
     Posted to Bugtraq on March 30, 2000 by USSR
     Labs <labs@ussrback.com>.

 reference
      advisory:
```

```
                MS00-021: Malformed TCP/IP Print
                Request Vulnerability
                (MS)
        advisory:
                USSR-2000037: Remote DoS Attack in
                Windows 2000/NT 4.0 TCP/IP Print
                Request Server Vulnerability
                (USSR)
        web page:
                Frequently Asked Questions: Microsoft
                Security Bulletin (MS00-021)
                (Microsoft)
        web page:
                Q257870: Malformed Print Request May
                Stop Windows 2000 TCP/IP Printing
                Service
                (Microsoft)

                 http://www.securityfocus.com/bid/1082
>
                                                    Disclaimer
                                          About The Vulnerability Database
```

The source address 141.211.176.99 is associated with:

```
> University of Michigan (NET-UMNET1)
   Information Technology Division (ITD)
   535 West William Street
   Ann Arbor, MI 48103-4943
   US

   Netname: UMNET1
   Netblock: 141.211.0.0 - 141.211.255.255

   Coordinator:
      University of Michigan Hostmaster  (UM17-ORG-ARIN)
hostmaster@UMICH.EDU
      +1 313 647-4267
Fax- +1 313 764-5140

   Domain System inverse mapping provided by:

   DNS.ITD.UMICH.EDU            141.211.144.15
   DNS2.ITD.UMICH.EDU           141.211.125.15
   DNS.CS.WISC.EDU              128.105.2.10

   Record last updated on 18-Dec-1997.
   Database last updated on 28-Mar-2001 22:46:19 EDT.
```

The address destination 216.181.129.185:

```
Integrated Technology Solutions (NETBLK-ITS3-DS)
   1450 S. Rolling Road
   Baltimore, MD 21227
```

```
   US

   Netname: ITS3-DS
   Netblock: 216.181.129.160 - 216.181.129.191

   Coordinator:
      Administrator, Operations  (OA20-ARIN)
opsadmin@DIGITALSELECT.NET
      703-435-0400

   Record last updated on 15-Oct-1999.
   Database last updated on 28-Mar-2001 22:46:19 EDT.
```

FROM: http://www.sans.org/newlook/alerts/port515.htm

```
"...on October 4, 2000 there were advisories released regarding
vulnerabilities for the
              LPR service, for many distributions of Linux and for the
BSD variants. We believe that
              the increase in probes to port 515 is for attackers
looking for this vulnerability."
http://www.kb.cert.org/vuls/id/382365 (lots o good stuff here)
```

MY.NET.70.38 – COLLATERAL DETECT

While looking at the Port 515 events I incidentally happened to notice this scan ... victim or scanner?; Destination port range suggests traceroute or load balancer? Coming *from* MY.NET.

```
> grep MY.NET.70.38 ../SCAN2/SnortS39.txt
Jan  9 16:43:21 MY.NET.70.38:59427 -> 128.183.104.105:33438 UDP
Jan  9 16:43:22 MY.NET.70.38:59427 -> 128.183.104.105:33445 UDP
Jan  9 16:43:22 MY.NET.70.38:59427 -> 128.183.104.105:33446 UDP
Jan  9 16:43:22 MY.NET.70.38:59427 -> 128.183.104.105:33448 UDP
Jan  9 16:43:22 MY.NET.70.38:59427 -> 128.183.104.105:33450 UDP
Jan  9 16:43:22 MY.NET.70.38:59427 -> 128.183.104.105:33451 UDP
Jan  9 16:43:22 MY.NET.70.38:59427 -> 128.183.104.105:33452 UDP
Jan  9 16:43:23 MY.NET.70.38:59427 -> 128.183.104.105:33453 UDP
Jan  9 16:43:23 MY.NET.70.38:59427 -> 128.183.104.105:33456 UDP
Jan  9 16:43:23 MY.NET.70.38:59427 -> 128.183.104.105:33457 UDP
Jan  9 16:43:23 MY.NET.70.38:59427 -> 128.183.104.105:33460 UDP
Jan  9 16:43:23 MY.NET.70.38:59427 -> 128.183.104.105:33461 UDP
Jan  9 16:43:23 MY.NET.70.38:59427 -> 128.183.104.105:33462 UDP
Jan  9 16:43:23 MY.NET.70.38:59427 -> 128.183.104.105:33463 UDP
>
```

Is MY.NET.70.38 an FTP server?

```
> grep MY.NET.70.38 ../SCAN1/SnortS26.txt
Dec 27 09:44:11 62.158.93.109:4766 -> MY.NET.70.38:21 SYN **S*****
Dec 27 09:44:13 62.158.93.109:4766 -> MY.NET.70.38:21 SYN **S*****
>  grep MY.NET.70.38 ../SCAN2/SnortS32.txt
Jan  1 16:29:39 217.80.182.182:2660 -> MY.NET.70.38:21 SYN **S*****
```

```
> grep MY.NET.70.38 ../SCAN1/SnortS29.txt
Jan  1 16:29:39 217.80.182.182:2660 -> MY.NET.70.38:21 SYN **S*****
> grep MY.NET.70.38 ../SCAN1/SnortS24.txt
Dec 29 16:26:00 62.226.88.105:1576 -> MY.NET.70.38:21 SYN **S*****
```

MY.NET.70.38
In looking for more data regarding MY.NET.70.38 I ran across this excerpt, which includes a logged reference to NMAP, the network mapping tool, which was being deployed in this example against another MY.NET host. This is an internal host scanning the internal network. If this is not being done by the security staff then this host has either been compromised or is being misused and should be shut down and investigated.

```
...
../ALERT/SnortA48.txt:01/18-15:48:12.745888  [**] spp_portscan:
portscan status from MY.NET.70.38: 2 connections across 1 hosts: TC]
../ALERT/SnortA48.txt:01/18-15:48:14.440647  [**] spp_portscan: End of
portscan from MY.NET.70.38 (TOTAL HOSTS:1 TCP:2 UDP:0) [**]
../ALERT/SnortA48.txt:01/18-15:33:14.404615  [**] NMAP TCP ping! [**]
MY.NET.70.38:52342 -> MY.NET.0.29:31844
../ALERT/SnortA48.txt:01/18-15:34:55.150396  [**] connect to 515 from
inside [**] MY.NET.70.38:3806 -> MY.NET.0.30:515
../ALERT/SnortA48.txt:01/18-15:35:05.274417  [**] connect to 515 from
inside [**] MY.NET.70.38:3812 -> MY.NET.0.30:515
../ALERT/SnortA48.txt:01/18-15:35:08.270626  [**] connect to 515 from
inside [**] MY.NET.70.38:3812 -> MY.NET.0.30:515
../ALERT/SnortA48.txt:01/18-15:49:53.881534  [**] spp_portscan:
PORTSCAN DETECTED from MY.NET.70.38 (STEALTH) [**]
../ALERT/SnortA48.txt:01/18-15:49:55.357900  [**] spp_portscan:
portscan status from MY.NET.70.38: 3 connections across 1 hosts: TC]
...
```

MY.NET.70.38 shows up as a source or destination in these files:

```
../ALERT/SnortA40.txt:3 - Jan 9
../ALERT/SnortA48.txt:1159 - Jan 18
../ALERT/SnortA51.txt:45 - Jan 4
../SCAN1/SnortS18.txt:1 - Jan 3
../SCAN1/SnortS21.txt:1 - Dec 9
../SCAN1/SnortS24.txt:1 - Dec 29
../SCAN1/SnortS26.txt:2 - Dec 27
../SCAN1/SnortS29.txt:1 - Jan 1
../SCAN2/SnortS39.txt:14 - Jan 9
```

The following hits from Jan 9 could have been a traceroute, from the range of the destination ports, and the closeness in time of successive packets. The discrepancy in time between the alert log and the scan log is curious, if they came from the same Snort host, as the 14 UDP connections in the alert log and the 14 entries in the scan log make it appear that they refer to the same incident.

```
>  grep  MY.NET.70.38 ../ALERT/*40.txt
01/09-16:56:44.614243  [**] spp_portscan: PORTSCAN DETECTED from
MY.NET.70.38 (THRESHOLD 7 connections in 2 seconds) [**]
```

```
01/09-16:56:47.576265  [**] spp_portscan: portscan status from
MY.NET.70.38: 14 connections across 1 hosts: TCP(0), UDP(14) [**]
01/09-16:56:51.345168  [**] spp_portscan: End of portscan from
MY.NET.70.38 (TOTAL HOSTS:1 TCP:0 UDP:14) [**]
> grep MY.NET.70.38 ../SCAN2/*S39.txt
```

```
Jan  9 16:43:21 MY.NET.70.38:59427 -> 128.183.104.105:33438 UDP
Jan  9 16:43:22 MY.NET.70.38:59427 -> 128.183.104.105:33445 UDP
Jan  9 16:43:22 MY.NET.70.38:59427 -> 128.183.104.105:33446 UDP
Jan  9 16:43:22 MY.NET.70.38:59427 -> 128.183.104.105:33448 UDP
Jan  9 16:43:22 MY.NET.70.38:59427 -> 128.183.104.105:33450 UDP
Jan  9 16:43:22 MY.NET.70.38:59427 -> 128.183.104.105:33451 UDP
Jan  9 16:43:22 MY.NET.70.38:59427 -> 128.183.104.105:33452 UDP
Jan  9 16:43:23 MY.NET.70.38:59427 -> 128.183.104.105:33453 UDP
Jan  9 16:43:23 MY.NET.70.38:59427 -> 128.183.104.105:33456 UDP
Jan  9 16:43:23 MY.NET.70.38:59427 -> 128.183.104.105:33457 UDP
Jan  9 16:43:23 MY.NET.70.38:59427 -> 128.183.104.105:33460 UDP
Jan  9 16:43:23 MY.NET.70.38:59427 -> 128.183.104.105:33461 UDP
Jan  9 16:43:23 MY.NET.70.38:59427 -> 128.183.104.105:33462 UDP
Jan  9 16:43:23 MY.NET.70.38:59427 -> 128.183.104.105:33463 UDP
```

There was one hit on MY.NET.70.38 among the OOS files; this one is an illegal SYN-FIN-PUSH-URG frame to service Kerberos authentication port on 203.202.20.66 in Australia on Jan 4 at 12:33. At 12:37 on the same day MY.NET.70.38 began logging portscans in alert file SnortA51.txt.

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/04-12:33:02.700945 MY.NET.70.38:52576 -> 203.202.20.66:88
TCP TTL:42 TOS:0x0 ID:63272
**SF*P*U Seq: 0xAEF17506   Ack: 0x0   Win: 0x1000
TCP Options => WS: 10 NOP MSS: 265 TS: 1061109567 0 EOL EOL


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

References

FROM: *ftp://ftp.isi.edu/in-notes/iana/assignments/port-numbers*

```
kerberos         88/tcp    Kerberos
kerberos         88/udp    Kerberos
```

```
> whois -h whois.apnic.net 203.202.20.66
...
inetnum:      203.202.20.0 - 203.202.20.127
netname:      SPOTWIRE
descr:        Spotwire Pty Ltd
country:      AU
admin-c:      OA3-AP
tech-c:       OA3-AP
mnt-by:       MAINT-OPTUSCOM-AP
changed:      ipadmin@cwo.net.au 20000914
```

COMMENTS AND RECOMMENDATIONS

There is plenty to be concerned about here, both internally and from outside. Establish Ingress and Egress filtering if not done, in order to prevent traffic with illegally spoofed MY.NET addresses to enter the network from the outside, and as well to prevent any illegally spoofed traffic hiding it's origin as being within the MY.NET.
Take a closer look at host MY.NET.70.38. Try to achieve more consistent logging so that suspicious activity can be analyzed from various angles. If the scan and alert logging is being done on separate hosts, get them to synchronize their time stamps.

EndNotes

I found the Snortsnarf tool to be of limited usefulness, possibly due to my inexperience with it and with Perl, but it seemed only effective in parsing out the different alert types. I was obliged to use other tools to process addresses and ports, and so forth. Perhaps Snort's methods of writing rules and logs has changed from time to time, as has Snortsnarf. I look forward to becoming more proficient with these tools. Following the advice of some of the prior practical writers, I tried to keep the tools relatively simple.

http://www.sans.org/y2k/practical/Marc Bayerkohler GCIA.doc

Also from Bayerkohler regarding analysis methods: "This made the files easier to work with for me. I started to browse through the html SnortSnarf had created, but it turns out the data
was still too big! All of the files had not been processed, because the html filled my 900 megabyte partition. So rather than using SnortSnarf any more, I fell back to the useful tools of the command line, most especially grep. "

LESSON LEARNED, for a person such as myself who is a network tech but not an experience sysadmin: Time spent learning perl, grep, sort, uniq -c, etcetera is critical.

Here is an example of a shell script I used to help retrieve address:port pairs associated with particular alerts, from delimited alert logs, with thanks to
http://www.sans.org/y2k/practical/Teri Bidwell GCIA.doc

```
> cat alertscript
#!/bin/bash
#from Bidwell
#edit type for whatever alert type is of concern at the moment.
# or use command arguments
#type=spp_portscan
#type=000220
type=$2
#grep -i $type *.txt > $type.grep
#grep -i $type SnortA6.txt > $type.grep
#grep -i $type ../sansAlert35.txt.delim > $type.grep
grep -i $type $1 > $type.grep

#get the source-port pairs
cat $type.grep | awk -F"&" '{print $3}' > $type.src-p.grep
```

```
#get the dest-port pairs
cat $type.grep | awk -F"&" '{print $4}' > $type.dst-p.grep

#get the src ips
cat $type.src-p.grep | awk -F":" '{print $1}' | sort | uniq -c | sort >
$type.srcips.sorted

#get the src ports
cat $type.src-p.grep | awk -F":" '{print $2}' | sort | uniq -c | sort >
$type.srcports.sorted

#get the dst ips
cat $type.dst-p.grep | awk -F":" '{print $1}' | sort | uniq -c | sort >
$type.dstips.sorted

#get the dst ports
cat $type.dst-p.grep | awk -F":" '{print $2}' | sort | uniq -c | sort >
$type.dstports.sorted
```

I Edited *alertscript* to take command line arguements using $1, $2, like this:
>sh alertscript ../sansAlert27.txt.delim SYN-FIN

Here is a perl script that was helpful to delimit the Alert logs.

```
> cat myscan3.pl
#!/opt/optivity/nms/lnms/perl/bin/perl
#originally from McGlaughlin dana_mclaughlin_gcia.doc
#(http://www.sans.org/y2k/analysts.htm)
#this file is to '&' delimit Short Alert logs for
#SANS Jan 2001 IDS practicum.
#11/24-22:38:26.337001  [**] spp_portscan: End of portscan from
144.51.17.1 (TOTAL HOSTS:1 TCP:0 UDP:10) [**]
#11/24-22:26:50.430596  [**] WinGate 1080 Attempt [**]
205.136.57.121:2374 -> MY.NET.97.242:1080
use warnings;
use strict;

if (open (ATTACK, $ARGV[0])) {
        }
        else {
                die ("Cannot open input file!");
        }

# Initialize $line
my $line = "";
my $delim = "$ARGV[0].delim";

# by default OUTFILE goes in the same directory as ARGV[]
#open (OUTFILE, "> sansScan.txt");
open (OUTFILE, "> $delim");

while ($line = <ATTACK>)  {
```

```
# substitute & for [**]
$line =~ s/\[\*\*\]/&/g;

# substitute & for ->
$line =~ s/->/&/;

# substitute hosts:= for hosts:
$line =~ s/hosts:/hosts:=/;

# substitute spp_portscan for spp_portscan:
$line =~ s/spp_portscan:/spp_portscan/;

# substitute from& for from
$line =~ s/from/from&/;

# substitute &( for (
$line =~ s/\(/&\(/;

# substitute & for : space
$line =~ s/:\s/&/;

#for testing:
#print "the first match was $1\n";
#print "the output file is $delim\n";



#send altered line to output
print OUTFILE "$line";

#for testing:
#print "$line";
}
close OUTFILE;
>
```

This perl script helped to parse out delimited scan logs.:

```
> cat myscan2.pl
#!/opt/optivity/nms/lnms/perl/bin/perl
#from McGlaughlin: "I used a pattern like this to
#separate the concatenated SnortS* file information
#into separate files for the different
#type of scans:"

#if (open (ATTACK, "sfiles.log")) {
if (open (ATTACK, $ARGV[0])) {
        }
        else {
                die ("Cannot open input file!");
        }
#Initialize $parse and $attack to set name of file
my $parse = "$ARGV[0].parse";
my $attack = "SYNFIN";
```

```
open (OUTFILE, "> myscan2.output/$attack$parse");

while ($line = <ATTACK>)  {
#       if ($line =~ /$attack/ && $line !~ /SYNFIN/) {
        if ($line =~  /$attack/) {
                @words1 = split (/,/, $line);

                $source = @words1[1];
                $sport = @words1[2];
                $dest = @words1[3];
                $dport = @words1[4];
                $stype = @words1[5];
                $flags = @words1[6];
                $rbits = @words1[7];

                print OUTFILE "$source ";
                print OUTFILE "$sport ";
                print OUTFILE "$dest ";
                print OUTFILE "$dport ";
                print OUTFILE "$stype ";
                print OUTFILE "$flags ";
                print OUTFILE "$rbits\n";
        }
}
close OUTFILE;
>
```

On "to 515 from" alerts, the delimit process and the alertscript process doesn't work right. The source shows as "outside" while the dst shows as the actual source address! I needed to do some additional massaging of the logs using vi to adjust the delimiters.

Other Resources

TCP/IP Architecture, Protocols, and Implementation; Feit, Sidnie; McGraw Hill 1993

A Practical Guide to Solaris; Sobell, Mark G.; Addison Wesley 1999

Beginning Perl; Cozens, Simon; Wrox Press Ltd. 2000

IDS Signatures and Analysis; Northcutt, Stephen; SANS Institute Course Book for SANS New Orleans 2001

Intrusion Signatures and Analysis; Northcutt, Sephen et.al.; New Riders 2001