Global Information Assurance Certification Paper

SANS GIAC Practical Assignment for GCIA
JoanneTreurniet
Submitted September 22, 2000

Assignment 1: Network Detects

All of the following are network traces, collected using the SHADOW IDS from August 26 to August 30, 2000. The output format is tcpdump 3.4-19 (Red Hat) format.

Trace #1: Unsolicited SYN-ACKs

1. **The trace:**August 26, 2000.

```
14:25:50.601342 P 205.188.146.23.80 > xxx.www.15.97.3072: S 3671365637:3671365637(0) ack 2134707042 win 8760  (DF)
14:25:50.703916 P 205.188.146.23.80 > xxx.yyy.132.11.3072: S 1490583475:1490583475(0) ack 1707889013 win 8760  (DF)
14:25:50.779169 P 205.188.146.23.80 > xxx.zzz.96.25.1024: S 519011171:519011171(0) ack 1559479582 win 8760  (DF)
14:25:50.930877 P 205.188.146.23.80 > xxx.zzz.203.96.1024: S 3697088346:3697088346(0) ack 838145105 win 8760  (DF)
14:26:08.143181 P 205.188.146.23.80 > xxx.zzz.69.111.1024: S 2105300557:2105300557(0) ack 358331415 win 8760  (DF)
14:26:11.171607 P 205.188.146.23.80 > xxx.www.239.97.1024: S 985835859:985835859(0) ack 1224972921 win 8760  (DF)
14:26:16.386441 P 205.188.146.23.80 > xxx.www.183.90.1024: S 1028965770:1028965770(0) ack 3481083729 win 8760  (DF)
14:26:22.543507 P 205.188.146.23.80 > xxx.zzz.248.88.1024: S 961402307:961402307(0) ack 810377019 win 8760  (DF)
14:26:27.739501 P 205.188.146.23.80 > xxx.zzz.199.97.1024: S 3993142508:3993142508(0) ack 2036921356 win 8760  (DF)
14:26:33.028251 P 205.188.146.23.80 > xxx.www.152.14.3072: S 1381805934:1381805934(0) ack 660372241 win 8760  (DF)
14:26:34.142812 P 205.188.146.23.80 > xxx.www.94.68.1024: S 1408875300:1408875300(0) ack 3909878529 win 8760  (DF)
14:26:34.940319 P 205.188.146.23.80 > xxx.www.118.86.3072: S 2599360045:2599360045(0) ack 3743101225 win 8760  (DF)
14:26:40.134523 P 205.188.146.23.80 > xxx.yyy.147.55.3072: S 476227787:476227787(0) ack 2233570319 win 8760  (DF)
14:26:49.741032 P 205.188.146.23.80 > xxx.www.16.33.3072: S 1786084840:1786084840(0) ack 77419538 win 8760  (DF)
14:26:55.342197 P 205.188.146.23.80 > xxx.yyy.102.73.1024: S 1951445764:1951445764(0) ack 309417226 win 8760  (DF)
14:27:14.735120 P 205.188.146.23.80 > xxx.yyy.12.32.1024: S 2308950040:2308950040(0) ack 1555109689 win 8760  (DF)
14:27:27.233030 P 205.188.146.23.80 > xxx.zzz.111.53.1024: S 2517473810:2517473810(0) ack 609822243 win 8760  (DF)
14:27:35.235707 P 205.188.146.23.80 > xxx.zzz.250.126.1024: S 2479901254:2479901254(0) ack 2754211123 win 8760  (DF)
14:27:35.541196 P 205.188.146.23.80 > xxx.zzz.154.60.3072: S 1639899616:1639899616(0) ack 2757170010 win 8760  (DF)
14:27:44.520124 P 205.188.146.23.80 > xxx.yyy.138.93.1024: S 3389981171:3389981171(0) ack 2790402316 win 8760  (DF)
14:27:52.829212 P 205.188.146.23.80 > xxx.zzz.109.108.3072: S 3149504162:3149504162(0) ack 2778994733 win 8760  (DF)
```

2. **Source:** The data was collected from my network using the SHADOW IDS. The event was found using a "badhosts" filter based on previous bad behaviour from the source host (a SYN-ACK scan).
3. **Probability of spoofed source:** The source address is unlikely to have been spoofed. This trace shows what is most likely a third party effect. The source resolves to www2.aol.com, so it is a legitimate web server. Maybe an attacker attempted a SYN flood denial of service and used our addresses as the spoofed source. The earlier "SYN-ACK scan" (at 2AM the same day from the same source) was likely the same effect.
4. **Description of attack:** This is a third party effect of a SYN flood denial of service attack on 205.188.146.23.
5. **Attack mechanism:** An attacker performs a SYN flood on 205.188.146.23 using crafted packets. The victim replies to each SYN with a SYN-ACK, but the source addresses are spoofed to be those on our network. Hence we get what looks like a SYN-ACK scan of random machines on our network.
6. **Correlations:** This is well-documented. As an example, see http://www.sans.org/y2k/122299.htm.
7. **Evidence of active targeting:** There is no active targeting in this case; the source was the intended victim.
8. **Severity:**
   Criticality = 3 (random machines)
   Lethality = 0 (third party effect, no worry at all)
   System countermeasures = 3 (our systems are mostly patched)
   Network Countermeasures = 5 (our border is strongly protected)
   Severity = (3+0)-(3+5) = -5
9. **Defensive recommendation:** Take this address off the badhosts list; it shouldn't be there. Ensure that the firewall is stateful.
10. **Multiple choice question:**

```
If you see odd traffic coming from a host, you should
a) shun them
b) retaliate
c) phone the ISP
d) keep an eye on them by filtering for the IP
```

    The answer is d. The first 3 are totally reactionary: (a) would have led to complaints since the source is a legitimate web host, (b) would have given them even more trouble than they already have, (c) would have been okay but wouldn't you feel silly when you found out what the real cause was.

Trace #2: Traffic to ephemeral ports

1. **The trace:** August 26, 2000.

```
13:33:25.856988 P 24.69.204.123.3868 > xxx.yyy.196.3.21823: S 183776848:183776848(0) win 8192  (DF)
13:33:25.862010 P xxx.yyy.196.3.21823 > 24.69.204.123.3868: S 856277196:856277196(0) ack 183776849 win 8760  (DF)
13:33:25.947441 P 24.69.204.123.3868 > xxx.yyy.196.3.21823: . 1:1(0) ack 1 win 8760 (DF)
13:33:25.960034 P 24.69.204.123.3867 > xxx.yyy.196.3.21: P 56:62(6) ack 447 win 8314 (DF)
13:33:26.011255 P xxx.yyy.196.3.21 > 24.69.204.123.3867: . 447:447(0) ack 62 win 8760 (DF) [tos 0x10]
13:33:26.019639 P xxx.yyy.196.3.21 > 24.69.204.123.3867: P 447:500(53) ack 62 win 8760 (DF) [tos 0x10]
13:33:26.075538 P xxx.yyy.196.3.21823 > 24.69.204.123.3868: P 1:279(278) ack 1 win 8760 (DF) [tos 0x10]
13:33:26.075583 P xxx.yyy.196.3.21823 > 24.69.204.123.3868: F 279:279(0) ack 1 win 8760 (DF) [tos 0x10]
13:33:26.163376 P 24.69.204.123.3868 > xxx.yyy.196.3.21823: . 1:1(0) ack 280 win 8482 (DF)
13:33:26.175596 P 24.69.204.123.3868 > xxx.yyy.196.3.21823: F 1:1(0) ack 280 win 8482 (DF)
```

```
13:33:26.180083 P xxx.yyy.196.3.21823 > 24.69.204.123.3868: . 280:280(0) ack 2 win 8760 (DF) [tos 0x10]
:::::
13:33:31.414660 P xxx.yyy.196.3.21 > 24.69.204.123.3869: P 773:803(30) ack 194 win 8760 (DF) [tos 0x10]
13:33:31.502731 P 24.69.204.123.3869 > xxx.yyy.196.3.21: P 194:200(6) ack 803 win 7958 (DF)
13:33:31.550969 P xxx.yyy.196.3.21 > 24.69.204.123.3869: . 803:803(0) ack 200 win 8760 (DF) [tos 0x10]
13:33:31.553125 P xxx.yyy.196.3.20 > 24.69.204.123.3871: S 226179111:226179111(0) win 8760  (DF) [tos 0x8]
13:33:31.640744 P 24.69.204.123.3871 > xxx.yyy.196.3.20: S 183782574:183782574(0) ack 226179112 win 8760  (DF)
13:33:31.645194 P xxx.yyy.196.3.20 > 24.69.204.123.3871: . 1:1(0) ack 1 win 8760 (DF) [tos 0x8]
13:33:31.648711 P xxx.yyy.196.3.21 > 24.69.204.123.3869: P 803:856(53) ack 200 win 8760 (DF) [tos 0x10]
13:33:31.660044 P xxx.yyy.196.3.20 > 24.69.204.123.3871: P 1:839(838) ack 1 win 8760 (DF) [tos 0x8]
13:33:31.660046 P xxx.yyy.196.3.20 > 24.69.204.123.3871: F 839:839(0) ack 1 win 8760 (DF) [tos 0x8]
13:33:31.754786 P 24.69.204.123.3871 > xxx.yyy.196.3.20: . 1:1(0) ack 840 win 7922 (DF)
13:33:31.766747 P 24.69.204.123.3871 > xxx.yyy.196.3.20: F 1:1(0) ack 840 win 7922 (DF)
13:33:31.771173 P xxx.yyy.196.3.20 > 24.69.204.123.3871: . 840:840(0) ack 2 win 8760 (DF) [tos 0x8]
13:33:31.928870 P 24.69.204.123.3869 > xxx.yyy.196.3.21: . 200:200(0) ack 856 win 7905 (DF)
```

2. **Source:** The data was collected from my network using the SHADOW IDS. The event was found using a filter which logs based on a SYN connection to an ephemeral port on our network.
3. **Probability of spoofed source:**The source address is not spoofed.
4. **Description of attack:** This actually isn't an attack. It's a false positive. The filter triggered on SYN connections to ephemeral ports on machines on our network. The immediate assumption was a trojan of some kind, but on further investigation I found that early versions of proftpd used ephemeral ports instead of ftp-data (20). In this particular trace the user appears to have both regular ftp and proftpd, because port 20 is used later in the conversation.
5. **Attack mechanism:** Proftpd connects to port 21 on the ftp server. When data is to be transferred, the server connects to the client from an ephemeral port instead of the usual port 20.
6. **Correlations:**There are no correlations since this isn't actually an attack.
7. **Evidence of active targeting:**None.
8. **Severity:** N/A.
9. **Defensive recommendation:** None.
10. **Multiple choice question:**

```
A SYN attempt to an ephemeral port on your network means (choose all that may apply):
a) a trojan on your network
b) someone on your network is using proftpd
c) someone on another network is using proftpd to download from you
d) someone is port-scanning you
```

The answer is a, c and d.

Trace #3: POP3 scan

1. **The Trace:**August 27,2000.

```
03:56:01.341211 P 209.51.87.70.3622 > xxx.yyy.244.234.110: S 2484379651:2484379651(0) win 16060  (DF)
03:56:01.344023 P xxx.yyy.244.234.110 > 209.51.87.70.3622: S 2278945217:2278945217(0) ack 2484379652 win 10136  (DF)
03:56:01.739776 P 209.51.87.70.3622 > xxx.yyy.244.234.110: . 1:1(0) ack 1 win 16060  (DF)
03:56:01.744764 P xxx.yyy.244.234.110 > 209.51.87.70.3622: F 1:1(0) ack 1 win 10136  (DF)
03:56:02.283259 P 209.51.87.70.3622 > xxx.yyy.244.234.110: . 1:1(0) ack 2 win 16059  (DF)
03:56:02.284511 P 209.51.87.70.3622 > xxx.yyy.244.234.110: F 1:1(0) ack 2 win 16060  (DF)
03:56:02.285365 P xxx.yyy.244.234.110 > 209.51.87.70.3622: . 2:2(0) ack 2 win 10136  (DF)
03:56:02.723072 P 209.51.87.70.3633 > xxx.yyy.244.235.110: S 2492962244:2492962244(0) win 16060  (DF)
03:56:02.726303 P xxx.yyy.244.235.110 > 209.51.87.70.3633: S 2279185206:2279185206(0) ack 2492962245 win 10136  (DF)
03:56:03.279037 P 209.51.87.70.3633 > xxx.yyy.244.235.110: . 1:1(0) ack 1 win 16060  (DF)
03:56:03.284162 P xxx.yyy.244.235.110 > 209.51.87.70.3633: F 1:1(0) ack 1 win 10136  (DF)
03:56:03.847548 P 209.51.87.70.3633 > xxx.yyy.244.235.110: . 1:1(0) ack 2 win 16059  (DF)
03:56:03.848142 P 209.51.87.70.3633 > xxx.yyy.244.235.110: F 1:1(0) ack 2 win 16060  (DF)
03:56:03.849067 P xxx.yyy.244.235.110 > 209.51.87.70.3633: . 2:2(0) ack 2 win 10136  (DF)
03:58:22.132178 P 209.51.87.70.3939 > xxx.yyy.196.2.110: S 2622385037:2622385037(0) win 16060  (DF)
03:58:22.139967 P my.router > 209.51.87.70: icmp: host xxx.yyy.196.2 unreachable - admin prohibited filter
03:58:22.396799 P 209.51.87.70.3940 > xxx.yyy.196.8.110: S 2616514801:2616514801(0) win 16060  (DF)
03:58:22.651207 P 209.51.87.70.3941 > xxx.yyy.196.7.110: S 2616688000:2616688000(0) win 16060  (DF)
03:58:22.652179 P 209.51.87.70.3942 > xxx.yyy.196.1.110: S 2621985636:2621985636(0) win 16060  (DF)
03:58:22.657691 P my.router > 209.51.87.70: icmp: host xxx.yyy.196.7 unreachable - admin prohibited filter
03:58:23.568708 P 209.51.87.70.3944 > xxx.yyy.196.129.110: S 2624041172:2624041172(0) win 16060  (DF)
03:58:23.576443 P my.router > 209.51.87.70: icmp: host xxx.yyy.196.129 unreachable - admin prohibited filter
03:58:23.662119 P 209.51.87.70.3945 > xxx.yyy.196.114.110: S 2619881498:2619881498(0) win 16060  (DF)
03:58:23.823830 P 209.51.87.70.3946 > xxx.yyy.196.3.110: S 2627821816:2627821816(0) win 16060  (DF)
03:58:24.046695 P 209.51.87.70.3947 > xxx.yyy.196.14.110: S 2625142009:2625142009(0) win 16060  (DF)
03:58:24.342253 P 209.51.87.70.3948 > xxx.yyy.196.113.110: S 2628827089:2628827089(0) win 16060  (DF)
03:58:24.349991 P my.router > 209.51.87.70: icmp: host xxx.yyy.196.113 unreachable - admin prohibited filter
03:58:24.982664 P 209.51.87.70.3949 > xxx.yyy.196.10.110: S 2619091809:2619091809(0) win 16060  (DF)
03:58:24.990509 P my.router > 209.51.87.70: icmp: host xxx.yyy.196.10 unreachable - admin prohibited filter
03:58:25.349696 P 209.51.87.70.3950 > xxx.yyy.196.194.110: S 2630143693:2630143693(0) win 16060  (DF)
03:58:25.632978 P my.router > 209.51.87.70: icmp: host xxx.yyy.196.7 unreachable - admin prohibited filter
03:58:26.575756 P my.router > 209.51.87.70: icmp: host xxx.yyy.196.129 unreachable - admin prohibited filter
03:58:27.081592 P my.router > 209.51.87.70: icmp: host xxx.yyy.196.3 unreachable - admin prohibited filter
03:58:28.071142 P my.router > 209.51.87.70: icmp: host xxx.yyy.196.10 unreachable - admin prohibited filter
03:58:31.009452 P my.router > 209.51.87.70: icmp: host xxx.yyy.196.2 unreachable - admin prohibited filter
03:58:31.604826 P my.router > 209.51.87.70: icmp: host xxx.yyy.196.7 unreachable - admin prohibited filter
03:58:32.567896 P my.router > 209.51.87.70: icmp: host xxx.yyy.196.129 unreachable - admin prohibited filter
03:58:33.378230 P my.router > 209.51.87.70: icmp: host xxx.yyy.196.113 unreachable - admin prohibited filter
03:58:33.997671 P my.router > 209.51.87.70: icmp: host xxx.yyy.196.10 unreachable - admin prohibited filter
```

2. **Source:** The data was collected from my network using the SHADOW IDS. The event was found using a filter which logs based on any SYN connection to a machine which doesn't allow the service to the port. Further investigation produced the above trace.
3. **Probability the source address was spoofed:** Since this was a scan, it is unlikely that the source was spoofed.
4. **Description of attack:** This was a scan for pop3 on port 110. A plethora of POP3 vulnerabilities exist. The attacker was scanning for open ports and did not attempt to exploit in this session. The CVE numbers are: CVE-1999-0006, CVE-1999-0042, CVE-1999-0272, CVE-1999-0494, CVE-1999-0920, CVE-2000-0034, CVE-2000-0091, CVE-2000-0139, CVE-2000-0140, CVE-2000-0399, CVE-2000-0442.
5. **Attack Mechanism:** SYNs were sent to port 110 on various machines on xxx.yyy.196. If a SYN-ACK response was obtained, the handshake was immediately

completed and gracefully closed. For those machines which do not provide pop3 services, the router dropped the packets and was kind enough to pass on the message. A breakdown of the ICMP packets shows these type 3 code 13 (admin prohibited) packets:

```
4500 0038 d17f 0000 fd01 4689 802b fd16
d133 5746 030d 9fe0 0000 0000 4500 003c
0b64 4000 2c06 d2d4 d133 5746 8388 c481
0f68 006e 9c67 b0d4
```

6. **Correlations:** This scan can be correlated with previous activity, 1 hour earlier, a similar scan for ftp port 21 on the same set of hosts:

```
02:22:47.526997 P 209.51.87.70.3008 > xxx.yyy.244.234.21: S 872339412:872339412(0) win 16060  (DF)
02:22:47.531800 P xxx.yyy.244.234.21 > 209.51.87.70.3008: S 750886910:750886910(0) ack 872339413 win 10136  (DF)
02:22:47.907092 P 209.51.87.70.3008 > xxx.yyy.244.234.21: . 1:1(0) ack 1 win 16060  (DF)
02:22:47.912788 P xxx.yyy.244.234.21 > 209.51.87.70.3008: F 1:1(0) ack 1 win 10136  (DF)
02:22:48.404518 P 209.51.87.70.3008 > xxx.yyy.244.234.21: . 1:1(0) ack 2 win 16059  (DF)
02:22:48.439593 P 209.51.87.70.3008 > xxx.yyy.244.234.21: F 1:1(0) ack 2 win 16060  (DF)
02:22:48.440347 P xxx.yyy.244.234.21 > 209.51.87.70.3008: . 2:2(0) ack 2 win 10136  (DF)
02:22:48.727794 P 209.51.87.70.3019 > xxx.yyy.244.235.21: S 867771204:867771204(0) win 16060  (DF)
02:22:48.730999 P xxx.yyy.244.235.21 > 209.51.87.70.3019: S 751188551:751188551(0) ack 867771205 win 10136  (DF)
02:22:49.031758 P 209.51.87.70.3019 > xxx.yyy.244.235.21: . 1:1(0) ack 1 win 16060  (DF)
02:22:49.036816 P xxx.yyy.244.235.21 > 209.51.87.70.3019: F 1:1(0) ack 1 win 10136  (DF)
02:22:49.356806 P 209.51.87.70.3019 > xxx.yyy.244.235.21: . 1:1(0) ack 2 win 16059  (DF)
02:22:49.358589 P 209.51.87.70.3019 > xxx.yyy.244.235.21: F 1:1(0) ack 2 win 16060  (DF)
02:22:49.359339 P xxx.yyy.244.235.21 > 209.51.87.70.3019: . 2:2(0) ack 2 win 10136  (DF)
02:25:04.154938 P 209.51.87.70.3325 > xxx.yyy.196.2.21: S 994615552:994615552(0) win 16060  (DF)
02:25:04.160932 P xxx.yyy.196.2.21 > 209.51.87.70.3325: R 0:0(0) ack 994615553 win 0 (DF)
02:25:04.229836 P 209.51.87.70.3326 > xxx.yyy.196.8.21: S 1005889501:1005889501(0) win 16060  (DF)
02:25:04.239076 P xxx.yyy.196.8.21 > 209.51.87.70.3326: S 1391542906:1391542906(0) ack 1005889502 win 10136  (DF)
02:25:04.604497 P 209.51.87.70.3327 > xxx.yyy.196.7.21: S 1007085225:1007085225(0) win 16060  (DF)
02:25:04.606380 P 209.51.87.70.3326 > xxx.yyy.196.8.21: . 1:1(0) ack 1 win 16060  (DF)
02:25:04.609641 P xxx.yyy.196.7.21 > 209.51.87.70.3327: S 2129122218:2129122218(0) ack 1007085226 win 10136  (DF)
02:25:04.612450 P xxx.yyy.196.8.21 > 209.51.87.70.3326: F 1:1(0) ack 1 win 10136  (DF)
02:25:05.010499 P 209.51.87.70.3327 > xxx.yyy.196.7.21: . 1:1(0) ack 1 win 16060  (DF)
02:25:05.012569 P 209.51.87.70.3326 > xxx.yyy.196.8.21: . 1:1(0) ack 2 win 16059  (DF)
02:25:05.013249 P 209.51.87.70.3326 > xxx.yyy.196.8.21: F 1:1(0) ack 2 win 16060  (DF)
02:25:05.013421 P 209.51.87.70.3328 > xxx.yyy.196.1.21: S 998318105:998318105(0) win 16060  (DF)
02:25:05.016350 P xxx.yyy.196.7.21 > 209.51.87.70.3327: F 1:1(0) ack 1 win 10136  (DF)
02:25:05.017587 P xxx.yyy.196.8.21 > 209.51.87.70.3326: . 2:2(0) ack 2 win 10136  (DF)
02:25:05.020428 P xxx.yyy.196.1.21 > 209.51.87.70.3328: R 0:0(0) ack 998318106 win 0
02:25:05.104080 P 209.51.87.70.3330 > xxx.yyy.196.129.21: S 998247160:998247160(0) win 16060  (DF)
02:25:05.112277 P xxx.yyy.196.129.21 > 209.51.87.70.3330: S 2129286622:2129286622(0) ack 998247161 win 10136  (DF)
02:25:05.305359 P 209.51.87.70.3331 > xxx.yyy.196.114.21: S 1012153529:1012153529(0) win 16060  (DF)
02:25:05.460554 P 209.51.87.70.3327 > xxx.yyy.196.7.21: . 1:1(0) ack 2 win 16059  (DF)
02:25:05.461690 P 209.51.87.70.3327 > xxx.yyy.196.7.21: F 1:1(0) ack 2 win 16060  (DF)
02:25:05.462044 P 209.51.87.70.3332 > xxx.yyy.196.3.21: S 1001198980:1001198980(0) win 16060  (DF)
02:25:05.466191 P xxx.yyy.196.7.21 > 209.51.87.70.3327: . 2:2(0) ack 2 win 10136  (DF)
02:25:05.469343 P xxx.yyy.196.3.21 > 209.51.87.70.3332: S 3196190773:3196190773(0) ack 1001198981 win 10136  (DF)
02:25:05.492826 P 209.51.87.70.3333 > xxx.yyy.196.14.21: S 1005385350:1005385350(0) win 16060  (DF)
02:25:05.494520 P 209.51.87.70.3330 > xxx.yyy.196.129.21: . 1:1(0) ack 1 win 16060  (DF)
02:25:05.500388 P xxx.yyy.196.129.21 > 209.51.87.70.3330: F 1:1(0) ack 1 win 10136  (DF)
02:25:05.502210 P xxx.yyy.196.14.21 > 209.51.87.70.3333: R 0:0(0) ack 1005385351 win 0
02:25:05.784329 P 209.51.87.70.3332 > xxx.yyy.196.3.21: . 1:1(0) ack 1 win 16060  (DF)
02:25:05.824767 P 209.51.87.70.3330 > xxx.yyy.196.129.21: . 1:1(0) ack 2 win 16059  (DF)
02:25:05.842086 P 209.51.87.70.3330 > xxx.yyy.196.129.21: F 1:1(0) ack 2 win 16060  (DF)
02:25:05.846618 P xxx.yyy.196.129.21 > 209.51.87.70.3330: . 2:2(0) ack 2 win 10136  (DF)
02:25:05.898376 P 209.51.87.70.3334 > xxx.yyy.196.113.21: S 996872635:996872635(0) win 16060  (DF)
02:25:05.900141 P 209.51.87.70.3335 > xxx.yyy.196.10.21: S 1011322476:1011322476(0) win 16060  (DF)
02:25:05.911666 P xxx.yyy.196.113.21 > 209.51.87.70.3334: R 0:0(0) ack 996872636 win 0
02:25:05.913047 P xxx.yyy.196.10.21 > 209.51.87.70.3335: R 0:0(0) ack 1011322477 win 0
02:25:06.201994 P 209.51.87.70.3336 > xxx.yyy.196.194.21: S 1010935387:1010935387(0) win 16060  (DF)
02:25:06.209875 P xxx.yyy.196.194.21 > 209.51.87.70.3336: S 2129454342:2129454342(0) ack 1010935388 win 10136  (DF)
02:25:06.301243 P xxx.yyy.196.3.21 > 209.51.87.70.3332: P 1:27(26) ack 1 win 10136  (DF) [tos 0x10]
02:25:06.525569 P 209.51.87.70.3336 > xxx.yyy.196.194.21: . 1:1(0) ack 1 win 16060  (DF)
02:25:06.531518 P xxx.yyy.196.194.21 > 209.51.87.70.3336: F 1:1(0) ack 1 win 10136  (DF)
02:25:06.689686 P 209.51.87.70.3332 > xxx.yyy.196.3.21: . 1:1(0) ack 27 win 16034  (DF)
02:25:06.706980 P 209.51.87.70.3332 > xxx.yyy.196.3.21: F 1:1(0) ack 27 win 16060  (DF)
02:25:06.711794 P xxx.yyy.196.3.21 > 209.51.87.70.3332: . 27:27(0) ack 2 win 10136  (DF) [tos 0x10]
02:25:06.712308 P xxx.yyy.196.3.21 > 209.51.87.70.3332: P 27:64(37) ack 2 win 10136  (DF) [tos 0x10]
02:25:06.714921 P xxx.yyy.196.3.21 > 209.51.87.70.3332: F 64:64(0) ack 2 win 10136  (DF) [tos 0x10]
02:25:06.894540 P 209.51.87.70.3336 > xxx.yyy.196.194.21: . 1:1(0) ack 2 win 16059  (DF)
02:25:06.925191 P 209.51.87.70.3336 > xxx.yyy.196.194.21: F 1:1(0) ack 2 win 16060  (DF)
02:25:06.929481 P xxx.yyy.196.194.21 > 209.51.87.70.3336: . 2:2(0) ack 2 win 10136  (DF)
02:25:07.143210 P 209.51.87.70.3332 > xxx.yyy.196.3.21: R 1001198982:1001198982(0) win 0 [tos 0x10]
02:25:07.143442 P 209.51.87.70.3332 > xxx.yyy.196.3.21: R 1001198982:1001198982(0) win 0 [tos 0x10]
02:25:08.409116 P 209.51.87.70.3331 > xxx.yyy.196.114.21: S 1012153529:1012153529(0) win 16060  (DF)
02:25:14.303210 P 209.51.87.70.3331 > xxx.yyy.196.114.21: S 1012153529:1012153529(0) win 16060  (DF)
```

7. **Evidence of active targeting:** The IPs targeted are a small subset of our network and are the same in both cases. This is an indication of previous recon activity. I postulate that s/he is looking for machines vulnerable to specific exploits.
8. **Severity:** The bad news is s/he got a reply from one of our machines, so s/he may use this to exploit. So far, I haven't seen this.

   Criticality = 3 (random machines)
   Lethality = 5 (recon with unknown final intent, but probably bad)
   System countermeasures = 2 (our systems are mostly patched)
   Network Countermeasures = 4 (our border is strongly protected, but allows in pop3 to 2 of the machines)
   Severity = (3+4)-(2+4) = 2

9. **Defensive recommendation:**Make sure we have the latest patches applied to the hosts which are targeted, especially those which replied. Keep a close eye on the attacker's behaviour by putting the address into the "badhosts" filter.
10. **Multiple choice question:**

    Given the following trace:

```
02:22:48.727794 P 209.51.87.70.3019 > xxx.yyy.244.235.21: S 867771204:867771204(0) win 16060  (DF)
02:22:48.730999 P xxx.yyy.244.235.21 > 209.51.87.70.3019: S 751188551:751188551(0) ack 867771205 win 10136  (DF)
02:22:49.031758 P 209.51.87.70.3019 > xxx.yyy.244.235.21: . 1:1(0) ack 1 win 16060  (DF)
02:22:49.036816 P xxx.yyy.244.235.21 > 209.51.87.70.3019: F 1:1(0) ack 1 win 10136  (DF)
02:22:49.356806 P 209.51.87.70.3019 > xxx.yyy.244.235.21: . 1:1(0) ack 2 win 16059  (DF)
02:22:49.358589 P 209.51.87.70.3019 > xxx.yyy.244.235.21: F 1:1(0) ack 2 win 16060  (DF)
02:22:49.359339 P xxx.yyy.244.235.21 > 209.51.87.70.3019: . 2:2(0) ack 2 win 10136  (DF)
What are possible causes (choose all that apply)?
a) A scanning tool
b) An ftp session ended before any attempt to authenticate
c) A refused connection
```

The answer is a and b.

Trace #4: TFTP Vulnerability

1. **The Trace:** August 30, 2000.

```
09:18:13.313928 P 207.107.223.113.2949 > xxx.www.48.103.69: 44 WRQ "/tmp/CyberCop.tftp.vulnerability" (ttl 115, id 43889)
09:18:13.537206 P 207.107.223.113.2955 > xxx.www.48.103.69: 23 RRQ "/etc/passwd" (ttl 115, id 48497)
09:18:18.188034 P 207.107.223.113.2941 > xxx.www.48.103.69: 18 RRQ "passwd" (ttl 115, id 56434)
09:18:18.569053 P 207.107.223.113.2955 > xxx.www.48.103.69: 23 RRQ "/etc/passwd" (ttl 115, id 57202)
09:18:23.098436 P 207.107.223.113.2941 > xxx.www.48.103.69: 18 RRQ "passwd" (ttl 115, id 9844)
09:18:23.562522 P 207.107.223.113.2955 > xxx.www.48.103.69: 23 RRQ "/etc/passwd" (ttl 115, id 12660)
09:18:28.120861 P 207.107.223.113.2941 > xxx.www.48.103.69: 18 RRQ "passwd" (ttl 115, id 58996)
09:18:28.462457 P 207.107.223.113.2955 > xxx.www.48.103.69: 23 RRQ "/etc/passwd" (ttl 115, id 61300)
09:18:31.288791 P 207.107.223.113.35550 > xxx.www.48.103.69: S 7051:7051(0) win 4096 (ttl 51, id 19264)
09:18:33.148486 P 207.107.223.113.2941 > xxx.www.48.103.69: 18 RRQ "passwd" (ttl 115, id 55669)
09:18:33.522420 P 207.107.223.113.2955 > xxx.www.48.103.69: 23 RRQ "/etc/passwd" (ttl 115, id 57973)
09:18:38.041912 P 207.107.223.113.3044 > xxx.www.48.103.69: 23 RRQ "/etc/passwd" (ttl 115, id 59254)
09:18:38.481779 P 207.107.223.113.3046 > xxx.www.48.103.69: 25 RRQ "../etc/passwd" (ttl 115, id 61558)
09:18:43.075772 P 207.107.223.113.3044 > xxx.www.48.103.69: 23 RRQ "/etc/passwd" (ttl 115, id 54391)
09:18:43.504890 P 207.107.223.113.3046 > xxx.www.48.103.69: 25 RRQ "../etc/passwd" (ttl 115, id 56695)
09:18:48.065351 P 207.107.223.113.3044 > xxx.www.48.103.69: 23 RRQ "/etc/passwd" (ttl 115, id 50808)
09:18:48.385736 P 207.107.223.113.3046 > xxx.www.48.103.69: 25 RRQ "../etc/passwd" (ttl 115, id 53112)
09:18:53.133343 P 207.107.223.113.3044 > xxx.www.48.103.69: 23 RRQ "/etc/passwd" (ttl 115, id 46201)
09:18:53.416092 P 207.107.223.113.3046 > xxx.www.48.103.69: 25 RRQ "../etc/passwd" (ttl 115, id 51833)
09:18:58.137104 P 207.107.223.113.3044 > xxx.www.48.103.69: 23 RRQ "/etc/passwd" (ttl 115, id 54138)
09:18:58.940319 P 207.107.223.113.3046 > xxx.www.48.103.69: 25 RRQ "../etc/passwd" (ttl 115, id 58490)
```

2. **Source:** The data was collected from my network using the SHADOW IDS. The event was found using a filter which collects based on UDP activity to port 69. The WRQ is a write request and RRQ is a read request.
3. **Probability the source address was spoofed:** The source was likely not spoofed since the attacker wants the response of /etc/passwd.
4. **Description of attack:** The trivial file transfer protocol is an unreliable version of ftp, using the UDP protocol. If not configured properly, it may be vulnerable. CVE-1999-0183 is the major CVE item describing this vulnerability. From xforce.iss.net,

```
Linux TFTP didn't restrict users to tftpboot directory, allowing remote
retrieval of files.

Description:
Some older Linux distributions did not correctly limit Trivial File
Transfer Protocol (TFTP) access to the /tftpboot directory, allowing
attackers to retrieve the passwd file as ../etc/passwd.
```

   A candidate CVE also exists, CAN-1999-0498 (under review).
5. **Attack Mechanism:** The attacker runs tftp and types "get /etc/passwd".
6. **Correlations:** The mention of Cybercop in the write request made me suspect that the attacker might be running a tool to scan the machine in question. The attacker also tried everything under the sun on this host. Together, this means s/he probably used Cybercop Scanner to scan the host.
7. **Evidence of active targeting:** Only the one host is targeted; the attacker has chosen it specifically but knows nothing of it.
8. **Severity:**
   Criticality = 2 (the host is a desktop workstation)
   Lethality = 3 (recon)
   System countermeasures = 3 (our systems are mostly patched)
   Network Countermeasures = 5 (our border is strongly protected)
   Severity = (2+3)-(3+5) = -1
9. **Defensive recommendation:** Make sure the host is patched up and watch for future activity from the attacker.
10. **Multiple choice question:**

```
In the following tcpdump trace:
09:18:58.940319 P 207.107.223.113.3046 > xxx.www.48.103.69: 25 RRQ "../etc/passwd" (ttl 115, id 58490
what does RRQ stand for?
a) Road Runner Query
b) Read Request
c) Recon Requisition
d) Relay Response Quest
```

    The answer is b.

Assignment 2: Evaluate an Attack

The Hping2 Reconnaissance Tool

I chose to analyze the hping2 recon tool for this section, available at http://developers.of.pl/antirez/hping/. With any luck, this will help me to identify when this tool is likely being used. I used this on my home computers, both running Red Hat Linux 6.2, with Snort running on the victim, "poit". The attacker is called "narf".

Are you up?

The most basic hping command, invoked by *hping poit*, generates the following output Snort log on poit:

```
09/16-09:39:59.655143 192.168.1.1:2136 -> 192.168.1.2:0
TCP TTL:64 TOS:0x0 ID:30473
******** Seq: 0x2FFB51EC   Ack: 0x0   Win: 0x200

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/16-09:39:59.655205 192.168.1.2:0 -> 192.168.1.1:2136
TCP TTL:255 TOS:0x0 ID:276
***R**A* Seq: 0x0   Ack: 0x2FFB51EC   Win: 0x0

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/16-09:40:00.647649 192.168.1.1:2137 -> 192.168.1.2:0
TCP TTL:64 TOS:0x0 ID:18065
******** Seq: 0x6997970C   Ack: 0x0   Win: 0x200

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/16-09:40:00.647715 192.168.1.2:0 -> 192.168.1.1:2137
TCP TTL:255 TOS:0x0 ID:277
***R**A* Seq: 0x0   Ack: 0x6997970C   Win: 0x0

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/16-09:40:01.647586 192.168.1.1:2138 -> 192.168.1.2:0
TCP TTL:64 TOS:0x0 ID:16519
******** Seq: 0x5DD45B3C   Ack: 0x0   Win: 0x200

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

It is aimed at port 0, which is generally closed. The packet sent has no flags set (null packet), which elicits a RST-ACK response from poit. This tells the attacker that the machine is up. This is useful if ICMP is filtered at the firewall.

Testing a Port

The command *hping poit -p 23* sends a null packet to port 23 on poit. Since the port is listening but doesn't know what to do with the null packet, it sends no response.

```
09/16-09:40:15.655489 192.168.1.1:2980 -> 192.168.1.2:23
TCP TTL:64 TOS:0x0 ID:64327
******** Seq: 0x3E2673B4   Ack: 0x0   Win: 0x200

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/16-09:40:16.647179 192.168.1.1:2981 -> 192.168.1.2:23
TCP TTL:64 TOS:0x0 ID:43455
******** Seq: 0x3980D9FF   Ack: 0x0   Win: 0x200

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/16-09:40:17.647017 192.168.1.1:2982 -> 192.168.1.2:23
TCP TTL:64 TOS:0x0 ID:34822
******** Seq: 0x227C16F5   Ack: 0x0   Win: 0x200

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/16-09:40:18.647022 192.168.1.1:2983 -> 192.168.1.2:23
TCP TTL:64 TOS:0x0 ID:42378
******** Seq: 0x24002708   Ack: 0x0   Win: 0x200

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/16-09:40:19.646945 192.168.1.1:2984 -> 192.168.1.2:23
TCP TTL:64 TOS:0x0 ID:11557
******** Seq: 0x73C4B9F6   Ack: 0x0   Win: 0x200
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

If the port had been closed, a RST-ACK would have been sent in reply.

Testing for firewalls

The command *hping poit -p 23 -A* sends an ACK packet to port 23 on poit:

```
09/16-09:40:31.655022 192.168.1.1:1283 -> 192.168.1.2:23
TCP TTL:64 TOS:0x0 ID:30225
******A* Seq: 0x4BACBA71   Ack: 0x0   Win: 0x200

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/16-09:40:31.655055 192.168.1.2:23 -> 192.168.1.1:1283
TCP TTL:255 TOS:0x0 ID:287
***R**** Seq: 0x0   Ack: 0x0   Win: 0x0

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/16-09:40:32.646493 192.168.1.1:1284 -> 192.168.1.2:23
TCP TTL:64 TOS:0x0 ID:13022
******A* Seq: 0x86C4BE9   Ack: 0x0   Win: 0x200

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/16-09:40:32.646525 192.168.1.2:23 -> 192.168.1.1:1284
TCP TTL:255 TOS:0x0 ID:288
***R**** Seq: 0x0   Ack: 0x0   Win: 0x0

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

Poit responds with a reset since the connection wasn't initiated with a SYN.

How can this be used to see if a port is firewalled? If the port is firewalled, the response to a NULL packet will be either ICMP destination unreachable or nothing. If we get either of these, the follow logic can be applied:

1. If we then send the port an ACK and get a RST, we know that a stateless firewall is being used because it let an ACK in with no connection.
2. If we then send the port an ACK and get no response again, the port is either filtered or they are using a stateful firewall.

TCP Spoofed Scanning using IP ID

The command *hping poit -p 23 -S* sends SYN packets to poit on port 23.

```
09/16-09:40:47.654348 192.168.1.1:2233 -> 192.168.1.2:23
TCP TTL:64 TOS:0x0 ID:45778
**S***** Seq: 0x5A318CC9   Ack: 0x0   Win: 0x200

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/16-09:40:47.654388 192.168.1.2:23 -> 192.168.1.1:2233
TCP TTL:64 TOS:0x0 ID:295  DF
**S***A* Seq: 0x8724527C   Ack: 0x5A318CCA   Win: 0x7FB8
TCP Options => MSS: 536

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/16-09:40:47.654658 192.168.1.1:2233 -> 192.168.1.2:23
TCP TTL:255 TOS:0x0 ID:215
***R**** Seq: 0x5A318CCA   Ack: 0x0   Win: 0x0

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

Poit replies with a SYN-ACK and narf sends a RST to terminate the handshake. This definitively proves that port 23 is listening. Here's what narf sees from the hping output:

```
default routing not present
HPING poit (eth0 192.168.1.2): S set, 40 headers + 0 data bytes
46 bytes from 192.168.1.2: flags=SA seq=1 ttl=64 id=296 win=32696 rtt=0.3 ms
46 bytes from 192.168.1.2: flags=SA seq=2 ttl=64 id=297 win=32696 rtt=0.3 ms
46 bytes from 192.168.1.2: flags=SA seq=3 ttl=64 id=298 win=32696 rtt=0.3 ms
46 bytes from 192.168.1.2: flags=SA seq=4 ttl=64 id=299 win=32696 rtt=0.3 ms

--- poit hping statistic ---
5 packets tramitted, 4 packets received, 20% packet loss
round-trip min/avg/max = 0.3/0.3/0.3 ms
```

The IP ID is incremented by 1 with each packet sent. If we use the relative IP ID (-r) option of hping, we can see how many packets poit sent out between responses to our SYN request: *hping poit -p 23 -S -r*

```
default routing not present
HPING poit (eth0 192.168.1.2): S set, 40 headers + 0 data bytes
46 bytes from 192.168.1.2: flags=SA seq=1 ttl=64 id=304 win=32696 rtt=0.3 ms
46 bytes from 192.168.1.2: flags=SA seq=2 ttl=64 id=+1 win=32696 rtt=0.4 ms
46 bytes from 192.168.1.2: flags=SA seq=3 ttl=64 id=+1 win=32696 rtt=0.4 ms
46 bytes from 192.168.1.2: flags=SA seq=4 ttl=64 id=+1 win=32696 rtt=0.3 ms

--- poit hping statistic ---
5 packets tramitted, 4 packets received, 20% packet loss
round-trip min/avg/max = 0.3/0.4/0.4 ms
```

Since this is on my home network and I'm not doing anything, the relative change in IP ID is obviously +1. The author of hping (Antirez) included this example in the HPING-HOWTO:

> # hping www.yahoo.com -P 80 -A -r
> ppp0 default routing interface selected (according to /proc)
> HPING www.yahoo.com (ppp0 204.71.200.68): A set, 40 headers + 0 data bytes
> 40 bytes from 204.71.200.68: flags=R seq=0 ttl=53 id=65179 win=0 rtt=327.1 ms
> 40 bytes from 204.71.200.68: flags=R seq=1 ttl=53 id=+1936 win=0 rtt=360.0 ms
> 40 bytes from 204.71.200.68: flags=R seq=2 ttl=53 id=+1880 win=0 rtt=340.0 ms
> 40 bytes from 204.71.200.68: flags=R seq=3 ttl=53 id=+1993 win=0 rtt=330.0 ms
> 40 bytes from 204.71.200.68: flags=R seq=4 ttl=53 id=+1871 win=0 rtt=350.0 ms
> 40 bytes from 204.71.200.68: flags=R seq=5 ttl=53 id=+1932 win=0 rtt=340.0 ms
> 40 bytes from 204.71.200.68: flags=R seq=6 ttl=53 id=+1776 win=0 rtt=330.0 ms
> 40 bytes from 204.71.200.68: flags=R seq=7 ttl=53 id=+1749 win=0 rtt=320.0 ms
> 40 bytes from 204.71.200.68: flags=R seq=8 ttl=53 id=+1888 win=0 rtt=340.0 ms
> 40 bytes from 204.71.200.68: flags=R seq=9 ttl=53 id=+1907 win=0 rtt=330.0 ms

As you can see, this gives an estimate of the amount of traffic going to www.yahoo.com. The tool includes an option to vary the amount of time between sending packets. On my network, this would again yield a relative IP ID of +1, so I'll quote the example of yahoo.com again:

> # hping www.yahoo.com -P 80 -A -r -i u 500000
> ppp0 default routing interface selected (according to /proc)
> HPING www.yahoo.com (ppp0 204.71.200.68): A set, 40 headers + 0 data bytes
> 40 bytes from 204.71.200.68: flags=R seq=0 ttl=53 id=35713 win=0 rtt=327.0 ms
> 40 bytes from 204.71.200.68: flags=R seq=1 ttl=53 id=+806 win=0 rtt=310.0 ms
> 40 bytes from 204.71.200.68: flags=R seq=2 ttl=53 id=+992 win=0 rtt=320.0 ms
> 40 bytes from 204.71.200.68: flags=R seq=3 ttl=53 id=+936 win=0 rtt=330.0 ms
> 40 bytes from 204.71.200.68: flags=R seq=4 ttl=53 id=+987 win=0 rtt=310.0 ms
> 40 bytes from 204.71.200.68: flags=R seq=5 ttl=53 id=+952 win=0 rtt=320.0 ms
> 40 bytes from 204.71.200.68: flags=R seq=6 ttl=53 id=+918 win=0 rtt=330.0 ms
> 40 bytes from 204.71.200.68: flags=R seq=7 ttl=53 id=+809 win=0 rtt=320.0 ms

40 bytes from 204.71.200.68: flags=R seq=8 ttl=53 id=+881 win=0 rtt=320.0 ms

Hping allows you to spoof the source as well with the -a option:
hping poit -p 23 -S -a poit
Here is the trace taken from poit:

```
09/16-09:41:49.652491 192.168.1.2:1178 -> 192.168.1.2:23
TCP TTL:64 TOS:0x0 ID:53878
**S***** Seq: 0x681D1F26   Ack: 0x0   Win: 0x200

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/16-09:41:50.643736 192.168.1.2:1179 -> 192.168.1.2:23
TCP TTL:64 TOS:0x0 ID:34444
**S***** Seq: 0x4BF4C29   Ack: 0x0   Win: 0x200

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

Putting it all together, we have the necessary elements for the spoofed SYN scan. Let's say you know of one computer that isn't used very much and you have a target. You can establish a "heartbeat" on the 3rd party computer by running *hping <othermachine> -S -r* in one window. Then you can send SYNs to the computer you want to scan on the port you want to scan with <othermachine> as the spoofed source.

- If the port was open, the SYN-ACKs would go to <othermachine>, who would respond "RESET", since it never initiated a connection. You would see the IP IDs in the "heartbeat" increase as it responds to the SYN-ACKs, thereby showing that the port is open.
- If the port was closed, the target would send <othermachine> a RST, to which <othermachine> would not reply at all. Hence if you didn't see an increase, you'd know the port was closed.

Note that this technique won't work for proxy or NATting firewalls since the relative IP ID is unpredictable with the large amount of traffic.

OS determination using hping

Quoting the HOWTO again,

> hpinging windows box without using --winid option you will see as increments are 256 multiple because different id byteordering. This can be really usefull for OS fingerprinting:

Since the byte ordering in different for windows or unix machines, the command *hping poit -p 23 -A -r --winid* will give a relative IP ID always equal to +256 if this option is set to the wrong value:

```
default routing not present
HPING poit (eth0 192.168.1.2): A set, 40 headers + 0 data bytes
46 bytes from 192.168.1.2: flags=R seq=1 ttl=255 id=16385 win=0 rtt=0.3 ms
46 bytes from 192.168.1.2: flags=R seq=2 ttl=255 id=+256 win=0 rtt=0.3 ms
46 bytes from 192.168.1.2: flags=R seq=3 ttl=255 id=+256 win=0 rtt=0.3 ms
46 bytes from 192.168.1.2: flags=R seq=4 ttl=255 id=+256 win=0 rtt=0.3 ms

--- poit hping statistic ---
5 packets tramitted, 4 packets received, 20% packet loss
round-trip min/avg/max = 0.3/0.3/0.3 ms
```

This is what poit's analyst would see:

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/16-09:41:33.652889 192.168.1.1:2978 -> 192.168.1.2:23
TCP TTL:64 TOS:0x0 ID:9398
******A* Seq: 0x5A0522EA   Ack: 0x0   Win: 0x200

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/16-09:41:33.652921 192.168.1.2:23 -> 192.168.1.1:2978
TCP TTL:255 TOS:0x0 ID:319
***R**** Seq: 0x0   Ack: 0x0   Win: 0x0

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/16-09:41:34.644299 192.168.1.1:2979 -> 192.168.1.2:23
TCP TTL:64 TOS:0x0 ID:1208
******A* Seq: 0x350E311A   Ack: 0x0   Win: 0x200

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/16-09:41:34.644329 192.168.1.2:23 -> 192.168.1.1:2979
TCP TTL:255 TOS:0x0 ID:320
***R**** Seq: 0x0   Ack: 0x0   Win: 0x0

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

Assignment 3: "Analyze This" Scenario

In this section we are asked to analyze data from a mystery network in an effort to make a bid for services to the owner of that network.

Executive Summary

Between June 27 and August 8, 2000, we were given the opportunity to perform an analysis of network traffic on MY.NET. This section of our report summarizes our findings. For further technical information, please refer to the Analysis of Alerts section.

Compromised Systems

MY.NET.6.34, 97.112, 100.230, 130.94, 253.24, 253.42, 253.51, 253.52 and 253.53 may be infected with the winoo (Windows Trinoo) trojan program. It may be worthwhile to check these hosts for anomalous programs, and/or keep watching their traffic for future activity. The following were sent mail containing the signature of the Happy 99 virus and should be checked: MY.NET.6.34, MY.NET.6.47, MY.NET.110.150, and MY.NET.253.42.

Napster activity

MY.NET.130.65, MY.NET.201.2, MY.NET.97.229 and MY.NET.97.230 are partaking in Napster activities. If you have a security policy set up, this activity is allowed. Otherwise, a security policy should be developed for your organization and this type of traffic stopped at your boundary protection device accordingly.

Scans

We have performed a statistical analysis of the scans, separating the scans into host scans (those scanning multiple hosts) and port scans (those scanning multiple ports on a single machine). Table 1 shows the top 5 host scanners against your network. Table 2 shows the top 5 port scanners against your network. Table 3 shows the top 5 most scanned machines on your network.

1	p3E9E2D79.dip0.t-ipconnect.de	41172
2	212.170.19.199 (Telefonica Data Espana)	32375
3	211.60.222.33 (DACOM Seoul, Korea)	23589
4	ci196729-a.wllmsn1.tn.home.com	22974
5	gatnoxs.com	22113

Table 1: Top 5 host scanners.

1	165.138.228.4 (State of Indiana, Department of Education)	1509
2	cc731098-a.hwrd1.md.home.com	1376
3	www.dslreports.com	1048
4	ABoulogne-102-1-2-190.abo.wanadoo.fr	865
5	MY.NET.1.3	789

Table 2: Top 5 port scanners.

You may notice that a port scan is reported to be originating from your own network; this traffic is UDP, originating from port 53 and going to an ephemeral port on the destination. This makes it look as though it should be normal DNS traffic (a reply to a DNS request), but the transmission rate is far too fast and happens in fits and starts. Further investigation is required to determine whether this is a network misconfiguration or an exploit of some kind.

1	MY.NET.70.123	1376
2	MY.NET.97.83	1224
3	MY.NET.253.114	1140
4	MY.NET.98.118	1048
5	MY.NET.181.88	893

Table 3: Top 5 most scanned machines.

Conclusions

It is our conclusion from this analysis that there are possibly compromised systems on your network and possibly some network misconfigurations. We would like to offer our services in setting up a security policy, optimizing your network and protecting your network from attacks.

Analysis of Alerts

The sections to follow summarize the alerts reported in the snort logs and analyze them. Each section pertains to a particular rule that triggered the alert. For each, a description of what is happening is given, whether the intent was malicious or benign.

ICMP Destination Unreachable

The machine MY.NET.70.121 was attacked denial-of-service-style on August 5 at 18:30. Here are the facts to consider:

1. ICMP 3 - destination unreachable (from the ICMP Protocol section at http://www.networksorcery.com/enp/):

   > "This message is not generated in response to a datagram destined for a multicast address."

   Also from networksorcery:

   > "If, in the destination host, the IP module cannot deliver the datagram because the indicated protocol module or process port is not active, the destination host may send a destination unreachable message to the source host."

2. MY.NET.70.121 receives over 22600 ICMP destination unreachable messages within 12 minutes. The likelihood that 70.121 requested connections at a rate of 30 requests per second in normal use is quite low, so we conclude that this was either an attack by 70.121, on 70.121, or a third party effect of a decoy scan.
   - A: It is possible that 70.121 sent out a SYN scan and got many destination unreachable packets regarding the ports being closed (which we see) and many SYN-ACK replies (which we don't see). Once the machine had been flooded, the ones that replied with a SYN-ACK couldn't get through and 70.121 had to send it out a destination unreachable message of its own.
   - B: If the packets in A were crafted with 70.121 as the source, the same effect would be seen as in A. This would happen if an attacker was doing SYN scans with 70.121 crafted as the source host.
   - C: Third party effect of a decoy scan: This is a reasonable candidate as well, except for the sheer volume of traffic. I believe that B is a more likely

description of events.

3. The host MY.NET.140.9 appears to be under attack as well (see ICMP Time Exceeded). There were destination unreachable reports seen in this section as a result of that activity.

Table 4 below shows the statistics of the amount of traffic aimed at MY.NET.70.121 from each host. We conclude that this activity is most likely due to SYN scanning by an attacker who set the source address to be 70.121.

141.208.208.81:	2	204.143.112.136:	2
212.204.188.51:	2244	AC8EE25E.ipt.aol.com:	184
AC904DD0.ipt.aol.com:	1608	PPPa87-ResaleMillersville1-4R7151.saturn.bbn.com:	88
adsl-63-205-40-169.dsl.lsan03.pacbell.net:	1158	c197926-a.carneg1.pa.home.com:	9846
c633759-a.grlnd1.tx.home.com:	4692	dialup220.comesurfthe.net:	134
m900-mp1-cvx1c.not.ntl.com:	10	modem-98.fermium.dialup.pol.co.uk:	2
mu-216-68-192-50.fuse.net:	2448	onyx-dial-81.ras.ind.net:	4
pool0713.cvx6-bradley.dialup.earthlink.net:	50	ppp-037.max1.fdl.dyn.dotnet.com:	48
user-38ld9b9.dsl.mindspring.com:	89		

Table 4: ICMP destination unreachable traffic to MY.NET.70.121 from 18:30:02 to 18:42:17 on Aug 5, 2000.

128.109.100.100:	6	128.192.234.130:	10
128.8.7.3:	8	129.130.167.114:	10
129.137.231.235:	8	129.174.44.9:	6
129.237.15.1:	8	129.59.1.201:	6
131.215.48.15:	10	131.247.254.97:	10
132.198.101.254:	10	137.229.71.34:	6
141.161.61.81:	10	146.229.127.200:	8
149.169.254.30:	6	157.182.254.66:	6
170.140.127.97:	6	198.17.101.21:	10
198.83.5.10:	6	216.79.60.26:	6
AMP-MIT.MIT.EDU:	10	NLANR-AMP.imt.uwm.edu:	8
amp-33test.nlanr.net:	6	amp-ballston.ncsa.uiuc.edu:	6
amp-nlanr.net.msstate.edu:	10	amp-uiowa.nlanr.net:	6
amp-umass.oit.umass.edu:	6	amp.atmos.uiuc.edu:	6
amp.hpc.utexas.edu:	6	amp.its.yale.edu:	10
amp.nss.udel.edu:	6	amp.psc.edu:	6
amp.wustl.edu:	6	ampmon.memphis.edu:	6
cgtest-14.cns.vt.edu:	6	iubamp.uits.indiana.edu:	6
nai-a-ariz.nlanr.net:	12	nlanr-amp.itc.Virginia.EDU:	6
nlanr-amp.nws.orst.edu:	8	nlanr-amp.ua.edu:	6
nlanr-fermi.fnal.gov:	10	nlanr-monitor.ncsu.edu:	8
nlanr-monitor.utd.rochester.edu:	10	nlanr-ou.backbone.ou.edu:	8
nlanr.cac.psu.edu:	12	nlanr.net.okstate.edu:	6
nlanr.netcom.duke.edu:	10	nlanr.nts.uci.edu:	6
nlanrtrgt.harvard.edu:	6	stolen121-7.ncsa.uiuc.edu:	6
thor.item.ntnu.no:	6	uic-nlanr.adn.uic.edu:	10

Table 5: ICMP destination unreachable traffic to MY.NET.140.9 from 18:30:02 to 18:42:17 on Aug 5, 2000. Only those sites which sent more than 5 destination unreachable messages are shown here.

ICMP Time Exceeded

MY.NET.140.9 was attacked on August 5 at 18:30 until 19:14. There were 5830 ICMP time exceeded packets directed at 140.9 in 44 minutes, or just over 2 per second.

This flurry of activity coincided with 542 destination unreachable messages (see Table 5), almost entirely from .edu sites with "nlanr" or "amp" in their names. A web search resulted in the following from http://moat.nlanr.net/:

> NLANR [National Laboratory for Applied Network Research] is supporting and extending a Network Analysis Infrastructure (NAI) to derive a better understanding of systemic service models and metrics of the Internet, with a specific focus on NSF's High Performance networking community. This includes passive measurements based on analysis of packet header traces to, e.g., derive workload profiles, active measurements which probe service properties, SNMP information from participating servers, and Internet routing related information based on BGP data.

AMP is an acronym for Active Monitoring Program. It can be concluded that this traffic is benign so long as 140.9 is involved in this project.

The addresses from whence the time exceeded packets came appear to be routers, judging from their resolved names. Hence the time exceeded message has nothing to do with fragment reassembly. Either this is due to traceroute activity from 140.9, or there is a routing loop somewhere. Since it only is coming to 140.9 and few others, we conclude it's tracerouting. Since these are mostly gateway and atm-like names and many of them sent just a few, one can postulate that this is likely just a wide sweep of traceroutes, which would correspond to what one might expect from the NLANR AMP program, which involves mapping the Internet. Table 6 shows a sample of the traffic due to time exceeded involving MY.NET.140.9, for those addresses sending more than 15 packets.

0car-0gw.oregon-gigapop.net:	16	192.5.89.45:	19
198.32.249.61:	30	198.48.91.77:	29
204.147.130.98:	52	Abilene-NYCM.maxgigapop.net:	646
BERK--SUNV.POS.calren2.net:	17	QAnh--USC.POS.calren2.net:	19
UCSD--USC.POS.calren2.net:	19	USC--abilene.ATM.calren2.net:	51
abilene-fiuup.net.fiu.edu:	16	abilene-gp.psc.net:	16
abilene-gw.ncni.net:	23	abilene-sox-rtr.abilene.sox.net:	83
atla-wash.abilene.ucaid.edu:	189	balt-core-vbns.maxgigapop.net:	120
clev-nycm.abilene.ucaid.edu:	378	dnvr-kscy.abilene.ucaid.edu:	179
hstn-atla.abilene.ucaid.edu:	29	ipls-clev.abilene.ucaid.edu:	295
jn1-at1-0-0-0.pym.vbns.net:	112	jn1-at1-0-0-13.wor.vbns.net:	49
jn1-at1-0-0-17.cht.vbns.net:	37	jn1-so3-0-0-0.rto.vbns.net:	18
jn1-so7-0-0-0.hsj.vbns.net:	21	jn1-so7-0-0-1.dng.vbns.net:	38
krc3-atm1-0s2.ohio-gigapop.net:	24	ks-2-p00.r.greatplains.net:	17
kscy-ipls.abilene.ucaid.edu:	215	losa-scrm.abilene.ucaid.edu:	41
oc12-pos-gigapop-OKC.onenet.net:	19	pos0-0.michnet8.mich.net:	20
scrm-dnvr.abilene.ucaid.edu:	75	sttl-dnvr.abilene.ucaid.edu:	32
wash-core-a0-0-3.maxgigapop.net:	642	wash-nycm.abilene.ucaid.edu:	174

Table 6: ICMP time exceeded traffic to MY.NET.140.9 on Aug. 5 2000 from 18:30:08 to 19:14:56.

MY.NET.14.2 was also sending out a lot of time exceeded (Table 7). My guess is that 14.2 is a router and and it is sending back these messages for a traceroute, because the packets were only sent for a short time on August 5. The other hosts are probably routers and also probably are responding to traceroutes.

Sending host	# Hits
MY.NET.14.2:	802
MY.NET.5.35:	31
MY.NET.98.199:	27

Table 7: ICMP time exceeded traffic not involving MY.NET.140.9.

GIAC 000218 VA-CIRT port 35555 and GIAC 000218 VA-CIRT port 34555

There was a significant amount of traffic from port 25 of various outside machines to port 35555 of MY.NET.6.34, MY.NET.100.230, MY.NET.253.24, MY.NET.253.41, MY.NET.253.43 and MY.NET.253.51. MY.NET.97.112 and 145.9 were contacted only once. A couple of those came from source port 113 (auth). This occurred once or twice a day from a different source each time, trying about 10 times.

The port numbers point to Windows Trinoo activity (Winoo). Winoo clients listen on 34555 and reply on 35555. See http://www.sans.org/y2k/021800.htm for more details of this alert. If the rule also triggered on content, one would know whether these machines are compromised or it is scanning, or it is just a coincidence.

Happy 99 Virus

On 4 occasions an alert triggered for Happy 99. Assuming the default Snort rules were used as a basis to trigger the alerts, these machines should be check for Happy 99:

```
07/26-07:50:56.700210 208.130.42.17:40221 -> MY.NET.6.34:25
08/05-11:22:48.017066 206.67.51.242:4889 -> MY.NET.6.47:25
07/11-19:28:57.652242 200.223.11.7:4836 -> MY.NET.110.150:25
07/19-04:28:40.867369 203.251.136.2:4985 -> MY.NET.253.42:25
```

The rule is:

```
alert tcp any any -> any 25 (msg:"Virus - Possible Outgoing Happy99 Virus"; content:"X-Spanska\:Yes";)
```

IDS247 - MISC - Large UDP Packet

```
08/05-18:30:03.777730 211.40.176.214:29536 -> MY.NET.98.179:6970
```

This happens 1170 times in 30 minutes from the same source to the same destination. The rule alerts on any UDP packet larger than 1200 bytes. Although 6970 is used by the trojan horse program Gate Crasher, RealAudio uses ports 6970-7170. Since there were a large number of packets sent, we conclude that this is a false positive.

Back Orifice

```
07/12-17:16:32.897041 202.159.46.234:31338 -> MY.NET.100.130:31337
```

The source site is in Indonesia (ip-jkt-234.indo.net.id). If the rule only triggered on a SYN, we can't know if the conversation continued. More data is required.

Napster 7777 and 8888 Data and Napster Client Data

Napster activity was found to have occurred on August 5, 2000 between 1800 and 1900h. If this does not violate your policy, it's not a problem. A description of normal Napster traffic follows.

From http://opennap.sourceforge.net/napster.txt:

```
1.  Client-Server protocol

Napster uses TCP for client to server communication.  Typically the servers
run on ports 8888 and 7777.  Note that this is different from the `metaserver'
(or redirector) which runs on port 8875.
```

Napster servers also listen on ports 4444, 5555 and 6666.

```
4.  Client-Client Protocol

File transfer occur directly between clients without passing through the
server.  There are four transfer modes, upload, download, firewalled upload,
firewalled download.  The normal method of transfer is that the client
wishing to download a file makes a TCP connection to the client holding the
file on their data port.  However, in the case where the client sharing the
file is behind a firewall, it is necessary for them to "push" the data by
making a TCP connection to the downloader's data port.
```

Napster connects to the host to the default port of 6699 when mp3 files are being exchanged between two clients. This is presumably a firewalled network, but we don't know where the sniffer is in relation to it. Access to the servers is apparently allowed (the 208.184 addresses are Napster servers).

The following shows a sample of the flow of traffic for typical Napster activity. MY.NET.130.65, MY.NET.201.2, MY.NET.97.229 and MY.NET.97.230 were involved.

```
08/05-18:30:07.112277  [**] Napster 8888 Data [**] MY.NET.201.2:1463 -> 208.184.216.191:8888
08/05-18:30:07.201812  [**] Napster 8888 Data [**] MY.NET.201.2:1463 -> 208.184.216.191:8888
08/05-18:30:10.125959  [**] Napster 8888 Data [**] 208.184.216.191:8888 -> MY.NET.201.2:1463
08/05-18:30:45.334122  [**] Napster 8888 Data [**] 208.184.216.191:8888 -> MY.NET.201.2:1463
08/05-18:30:45.334201  [**] Napster 8888 Data [**] MY.NET.201.2:1463 -> 208.184.216.191:8888
08/05-18:30:49.580603  [**] Napster 8888 Data [**] 208.184.216.191:8888 -> MY.NET.201.2:1463
08/05-18:30:50.051209  [**] Napster 8888 Data [**] 208.184.216.191:8888 -> MY.NET.201.2:1463
08/05-18:30:50.053254  [**] Napster 8888 Data [**] 208.184.216.191:8888 -> MY.NET.201.2:1463
08/05-18:30:50.056476  [**] Napster 8888 Data [**] 208.184.216.191:8888 -> MY.NET.201.2:1463
08/05-18:30:50.059569  [**] Napster 8888 Data [**] 208.184.216.191:8888 -> MY.NET.201.2:1463
08/05-18:31:07.175452  [**] Napster 8888 Data [**] MY.NET.201.2:1463 -> 208.184.216.191:8888
08/05-18:31:09.130144  [**] Napster 8888 Data [**] 208.184.216.191:8888 -> MY.NET.201.2:1463
08/05-18:31:09.591952  [**] Napster Client Data [**] MY.NET.201.2:1468 -> 128.32.57.190:6699
08/05-18:32:07.367905  [**] Napster Client Data [**] MY.NET.201.2:1469 -> 24.138.6.167:6699
08/05-18:32:31.097752  [**] Napster 8888 Data [**] 208.184.216.191:8888 -> MY.NET.201.2:1463
08/05-18:32:31.098641  [**] Napster 8888 Data [**] 208.184.216.191:8888 -> MY.NET.201.2:1463
:::::
```

Queso fingerprint

Queso is a tool for OS identification through fingerprinting. From the Queso home page,

```
What packets does QueSO send?

               0 SYN           * THIS IS VALID, used to verify LISTEN
               1 SYN+ACK
               2 FIN
               3 FIN+ACK
               4 SYN+FIN
               5 PSH
               6 SYN+XXX+YYY   * XXX & YYY are unused TCP flags

       All packets have a random seq_num and a 0x0 ack_num.
```

It is the 6th packet type that this rule triggers on, S12 flags. The following is a list of the alerts:

```
07/19-09:49:16.702569 212.171.169.46:24122 -> MY.NET.1.3:21
07/19-09:49:22.702119 212.171.169.46:22536 -> MY.NET.1.5:21
06/29-21:44:39.102167 24.3.29.155:1344 -> MY.NET.6.44:110
07/27-10:15:14.440976 210.84.179.196:15398 -> MY.NET.60.8:113
07/17-15:20:37.812781 193.233.7.254:3121 -> MY.NET.99.20:113
07/17-15:37:53.409978 193.233.7.65:3138 -> MY.NET.99.23:113
07/17-15:41:44.730499 193.233.7.65:3139 -> MY.NET.99.23:113
07/17-21:11:17.806127 192.203.80.142:3240 -> MY.NET.99.23:113
07/11-16:23:38.017796 194.159.73.26:27025 -> MY.NET.100.230:27005
06/29-17:19:06.762024 129.21.145.131:5776 -> MY.NET.217.98:113
06/29-17:19:12.466611 129.21.145.131:7470 -> MY.NET.217.98:20
```

Considering that a portscan of some kind came from every one of these hosts at some point in the month, the conclusion is that these hosts are owned by bad people or are compromised (since the scanner can't spoof his/her own address or s/he won't get a response).

For example, on 07/27 at 10:14, 210.84.179.196 initiated a SYN scan on MY.NET.60.8 on ports from 1 to 200 (except 4,11,13,15,17,19), followed by a quick S, SF, S12, F and P on port 113 (auth).

```
Jul 27 10:14:00 210.84.179.196:1054 -> MY.NET.60.8:1 SYN **S*****
Jul 27 10:14:00 210.84.179.196:1055 -> MY.NET.60.8:2 SYN **S*****
:::
:::
Jul 27 10:14:06 210.84.179.196:1253 -> MY.NET.60.8:199 SYN **S*****
```

```
Jul 27 10:15:14 210.84.179.196:15392 -> MY.NET.60.8:113 SYN **S*****
Jul 27 10:15:14 210.84.179.196:15396 -> MY.NET.60.8:113 SYNFIN **SF****
Jul 27 10:15:14 210.84.179.196:15398 -> MY.NET.60.8:113 SYN 21S***** RESERVEDBITS
Jul 27 10:15:14 210.84.179.196:15394 -> MY.NET.60.8:113 FIN ***F*****
Jul 27 10:15:14 210.84.179.196:15397 -> MY.NET.60.8:113 VECNA *****P**
```

Sun RPC high port access - attempted and successful

These alerts are triggered by any destination port 32771 activity. From http://advice.networkice.com/advice/Exploits/Ports/32771/default.htm:

```
Ghost Portmapper. Some SunOS machines listen at this port for
portmapper. Since firewalls frequently don't filter at high ports, it
can allow the attacker access to portmapper even when port 111 is blocked.
```

Attempted Sun RPC high port access

These alerts are mostly ICQ activity to fes-d024.icq.aol.com, an ICQ server. The alert triggered on a coincidence (false positive).

```
06/28-14:33:18.376906 fes-d024.icq.aol.com:4000 -> MY.NET.105.2:32771
:::
```

cc362592-a.hwrd1.md.home.com is going to port 32771 from ports 407 and 1419, alternating. Likewise for cc362592-b.hwrd1.md.home.com. The ports 407 and 1419 are associated with the remote administration tool Timbuktu. This may be a system administrator working from home:

```
07/11-09:45:53.237040 cc362592-a.hwrd1.md.home.com:407 -> MY.NET.115.95:32771
07/11-09:45:53.237718 cc362592-a.hwrd1.md.home.com:1419 -> MY.NET.115.95:32771
07/11-09:45:53.237827 cc362592-a.hwrd1.md.home.com:407 -> MY.NET.115.95:32771
07/11-09:45:53.238934 cc362592-a.hwrd1.md.home.com:1419 -> MY.NET.115.95:32771
07/11-09:45:53.239334 cc362592-a.hwrd1.md.home.com:407 -> MY.NET.115.95:32771
07/17-15:35:19.906685 cc362592-b.hwrd1.md.home.com:1419 -> MY.NET.115.91:32771
07/17-15:35:19.906743 cc362592-b.hwrd1.md.home.com:407 -> MY.NET.115.91:32771
07/19-14:26:12.632338 cc362592-b.hwrd1.md.home.com:1419 -> MY.NET.115.91:32771
07/19-14:26:12.632395 cc362592-b.hwrd1.md.home.com:407 -> MY.NET.115.91:32771
07/19-14:26:12.632451 cc362592-b.hwrd1.md.home.com:1419 -> MY.NET.115.91:32771
07/19-14:26:12.632508 cc362592-b.hwrd1.md.home.com:407 -> MY.NET.115.91:32771
07/19-14:26:12.632653 cc362592-b.hwrd1.md.home.com:1419 -> MY.NET.115.91:32771
07/19-14:26:12.634342 cc362592-b.hwrd1.md.home.com:407 -> MY.NET.115.91:32771
```

These remaining connection attempts may or may not have been successful.

```
07/08-07:21:32.145547 speedera-server-0.hisite.com:2385 -> MY.NET.1.8:32771
07/08-07:33:06.203162 207.230.26.34:1295 -> MY.NET.1.8:32771
07/11-16:32:43.189859 212.62.17.145:20969 -> MY.NET.1.10:32771
```

SUNRPC highport access!

This activity is from AOL ICQ servers (default 5190) and are nothing to worry about:

```
     07/29-16:24:35.903923 205.188.3.205:5190 -> MY.NET.98.145:32771
::: 13 times more
     07/29-16:24:53.247324 205.188.3.205:5190 -> MY.NET.98.145:32771
```

These, however, may be something to worry about:

```
     07/12-03:50:48.320005 204.137.237.8:3097 -> MY.NET.97.112:32771
     08/05-02:47:09.846709 192.102.249.3:25 -> MY.NET.130.94:32771
     08/05-02:47:09.972830 192.102.249.3:25 -> MY.NET.130.94:32771
     08/05-02:47:10.049275 192.102.249.3:25 -> MY.NET.130.94:32771
     08/05-13:37:20.335822 209.138.185.157:4067 -> MY.NET.253.114:32771
     08/05-13:37:20.337188 209.138.185.157:4067 -> MY.NET.253.114:32771
```

Nobody should access your high RPC ports, they're accessing services you only want to share with your friends.

WinGate 8080 Attempt

Undernet.Org and any IRC servers will scan for misconfigured Wingate or SOCKS proxy services before allowing a user to connect. Any alerts coming from these sources should be ignored.

The host MY.NET.253.105 is being scanned for proxy services on port 8080 by a wide variety of hosts. What is interesting about this scan is that the same 2 hosts try many times to access this port using a monotonically increasing source port number. This suggests two possibilities: either the attacker is stubborn, stupid or both, or this may be a false positive. It could be Microsoft Internet Explorer visiting a web page and generating connections for each and every image file on the page.

```
     06/27-00:05:15.022896 ppp-127.tnt-1.wdc.smartworld.net:1030 -> MY.NET.253.105:8080
     06/27-00:07:14.860062 ppp-127.tnt-1.wdc.smartworld.net:1106 -> MY.NET.253.105:8080
::::: - every minute or so
     06/27-00:46:15.711723 ppp-127.tnt-1.wdc.smartworld.net:1526 -> MY.NET.253.105:8080
     06/27-00:47:15.669857 ppp-127.tnt-1.wdc.smartworld.net:1531 -> MY.NET.253.105:8080
     06/27-02:29:30.326579 heimdal.iee.lu:61642 -> MY.NET.253.105:8080
     06/27-02:42:35.050074 heimdal.iee.lu:61704 -> MY.NET.253.105:8080
::::: - again every minute or so
     06/27-02:46:12.993203 heimdal.iee.lu:61730 -> MY.NET.253.105:8080
     06/27-03:45:41.043418 heimdal.iee.lu:61994 -> MY.NET.253.105:8080
     06/27-03:45:49.321507 heimdal.iee.lu:61995 -> MY.NET.253.105:8080
     06/27-05:43:19.586523 heimdal.iee.lu:62236 -> MY.NET.253.105:8080
     06/27-05:43:22.760965 heimdal.iee.lu:62236 -> MY.NET.253.105:8080
```

Every hour at the same time, this source tries again to connect to 8080. Without more data it is difficult to determine the point of it.

```
06/27-00:57:47.874602 h-205-217-233-122.netscape.com:45611 -> MY.NET.99.85:8080
06/27-01:57:47.564098 h-205-217-233-122.netscape.com:63323 -> MY.NET.99.85:8080
06/27-03:57:49.587272 h-205-217-233-122.netscape.com:56981 -> MY.NET.99.85:8080
06/27-04:57:49.731194 h-205-217-233-122.netscape.com:65234 -> MY.NET.99.85:8080
06/27-05:57:49.897979 h-205-217-233-122.netscape.com:46307 -> MY.NET.99.85:8080
06/27-06:57:54.954595 h-205-217-233-122.netscape.com:62755 -> MY.NET.99.85:8080
06/27-07:57:57.151266 h-205-217-233-122.netscape.com:46662 -> MY.NET.99.85:8080
06/27-08:57:58.009044 h-205-217-233-122.netscape.com:62853 -> MY.NET.99.85:8080
```

The remainder are likely scans for anonymous surfing or IRC via Wingate.

```
06/27-02:04:48.644171 port41.tserver.dbcc.ucf.edu:1266 -> MY.NET.20.10:8080
06/27-03:45:48.473124 cc451615-a.catv1.md.home.com:2962 -> MY.NET.100.127:8080
06/27-03:45:51.750095 cc451615-a.catv1.md.home.com:2963 -> MY.NET.100.127:8080
06/27-03:49:11.951578 AC906EF3.ipt.aol.com:3278 -> MY.NET.253.105:8080
06/27-07:37:06.451998 khalao.ncc.up.pt:1130 -> MY.NET.221.186:8080
06/27-07:37:09.441975 khalao.ncc.up.pt:1130 -> MY.NET.221.186:8080
06/27-07:37:15.442184 khalao.ncc.up.pt:1130 -> MY.NET.221.186:8080
06/27-07:37:27.440710 khalao.ncc.up.pt:1130 -> MY.NET.221.186:8080
06/27-09:33:12.147455 ss04.nc.us.ibm.com:60692 -> MY.NET.99.85:8080
06/27-09:33:12.876999 ss04.nc.us.ibm.com:60699 -> MY.NET.99.85:8080
06/27-09:34:20.326637 ss04.nc.us.ibm.com:62716 -> MY.NET.99.85:8080
06/27-09:34:21.014321 ss04.nc.us.ibm.com:62728 -> MY.NET.99.85:8080
```

WinGate 1080 Attempt

As stated in the previous section, Undernet.Org and any IRC servers will scan for misconfigured Wingate or SOCKS proxy services before allowing a user to connect.

```
06/27-00:51:25.609698 ProxyScan.MD.US.Undernet.Org:2546 -> MY.NET.97.235:1080
06/27-00:51:28.611857 ProxyScan.MD.US.Undernet.Org:2546 -> MY.NET.97.235:1080
06/27-12:56:17.094363 ProxyScan.MD.US.Undernet.Org:2465 -> MY.NET.53.214:1080
06/27-13:35:38.073667 irc.au.ac.th:40896 -> MY.NET.97.119:1080
```

>From the RFC,

```
3.  Procedure for TCP-based clients

   When a TCP-based client wishes to establish a connection to an object
   that is reachable only via a firewall (such determination is left up
   to the implementation), it must open a TCP connection to the
   appropriate SOCKS port on the SOCKS server system.  The SOCKS service
   is conventionally located on TCP port 1080.  If the connection
   request succeeds, the client enters a negotiation for the
   authentication method to be used, authenticates with the chosen
   method, then sends a relay request.  The SOCKS server evaluates the
   request, and either establishes the appropriate connection or denies it.
```

The others could be scans for SOCKS or Wingate. Many appear to be TCP retries, based on the source port and time indices.

```
06/27-00:34:31.223648 login1.powertech.no:2423 -> MY.NET.97.154:1080
06/27-03:15:27.232910 61.9.84.35:21153 -> MY.NET.97.105:1080
06/27-03:15:28.449997 61.9.84.35:21153 -> MY.NET.97.105:1080
06/27-03:15:30.447634 61.9.84.35:21153 -> MY.NET.97.105:1080
06/27-03:32:49.023668 213.78.27.204:4546 -> MY.NET.181.88:1080
06/27-03:43:31.521381 212.72.75.236:4819 -> MY.NET.97.162:1080
06/27-04:19:36.364557 212.170.18.67:4095 -> MY.NET.181.88:1080
06/27-06:13:09.506363 212.72.75.236:2372 -> MY.NET.97.211:1080
06/27-06:13:12.496125 212.72.75.236:2372 -> MY.NET.97.211:1080
06/27-06:35:20.691170 a2-1b134.neo.rr.com:1814 -> MY.NET.99.51:1080
06/27-06:35:21.185364 a2-1b134.neo.rr.com:1814 -> MY.NET.99.51:1080
06/27-07:27:48.319397 ip113.newark3.nj.pub-ip.psi.net:1389 -> MY.NET.60.11:1080
06/27-07:27:49.393757 ip113.newark3.nj.pub-ip.psi.net:1389 -> MY.NET.60.11:1080
06/27-07:27:50.177870 ip113.newark3.nj.pub-ip.psi.net:1389 -> MY.NET.60.11:1080
06/27-07:27:51.101313 ip113.newark3.nj.pub-ip.psi.net:1389 -> MY.NET.60.11:1080
06/27-08:27:13.168388 login1.powertech.no:4210 -> MY.NET.97.101:1080
06/27-10:38:29.596097 168.187.220.63:62730 -> MY.NET.60.11:1080
06/27-10:38:32.418083 168.187.220.63:62730 -> MY.NET.60.11:1080
06/27-11:59:38.015888 sgt-66-232.tm.net.my:1523 -> MY.NET.97.159:1080
06/27-11:59:40.876749 sgt-66-232.tm.net.my:1523 -> MY.NET.97.159:1080
06/27-12:00:26.722605 208.223.173.71:11314 -> MY.NET.6.7:1080
06/27-12:00:27.365970 208.223.173.71:11314 -> MY.NET.6.7:1080
06/27-12:00:27.981109 208.223.173.71:11314 -> MY.NET.6.7:1080
06/27-12:00:28.573174 208.223.173.71:11314 -> MY.NET.6.7:1080
06/27-12:01:13.336340 KIRcc-01p58.ppp.odn.ad.jp:3086 -> MY.NET.97.159:1080
06/27-12:01:20.699404 KIRcc-01p58.ppp.odn.ad.jp:3086 -> MY.NET.97.159:1080
06/27-12:16:30.590717 l245ppp151.ksc.net.th:1166 -> MY.NET.97.159:1080
06/27-12:16:31.795642 l245ppp151.ksc.net.th:1166 -> MY.NET.97.159:1080
06/27-12:16:34.149941 l245ppp151.ksc.net.th:1166 -> MY.NET.97.159:1080
06/27-13:20:12.138930 MY.NET.99.51:20 -> MY.NET.101.155:1080
06/27-13:35:36.875360 212.72.75.236:1703 -> MY.NET.97.119:1080
06/27-13:35:41.555027 KIRcc-01p58.ppp.odn.ad.jp:2532 -> MY.NET.97.119:1080
06/27-13:35:42.663208 KIRcc-01p58.ppp.odn.ad.jp:2532 -> MY.NET.97.119:1080
```

Tiny Fragments - Possible Hostile Activity

Tiny fragments indicate a firewall penetration technique, a DoS attack or a scan. Without further data it is difficult to determine the cause of these fragments.

```
06/28-06:35:13.540772 host01.quo.jsv.qwest.net: -> MY.NET.1.8:
06/28-06:35:13.540827 host01.quo.jsv.qwest.net: -> MY.NET.1.8:
06/28-06:35:13.540878 host01.quo.jsv.qwest.net: -> MY.NET.1.8:
06/28-06:37:13.538078 host01.quo.jsv.qwest.net: -> MY.NET.1.8:
06/28-06:37:13.538175 host01.quo.jsv.qwest.net: -> MY.NET.1.8:
```

```
06/28-06:37:13.538272 host01.quo.jsv.qwest.net: -> MY.NET.1.8:
07/11-03:33:54.281367 adsl-61-144-55.mia.bellsouth.net: -> MY.NET.230.241:
07/26-11:05:01.522342 202.76.177.204: -> MY.NET.70.20:
07/26-13:54:29.666358 202.76.177.204: -> MY.NET.70.20:
```

Possible wu-ftpd exploit - GIAC000623

Two separate rules for this attack triggered:

```
06/30-16:33:57.773279 151.164.223.206:4499 -> MY.NET.99.16:21
06/30-16:35:11.406398 151.164.223.206:4500 -> MY.NET.144.59:21
06/30-16:35:13.560305 151.164.223.206:4500 -> MY.NET.144.59:21
07/19-03:53:00.191779 d64.romantis.net:1245 -> MY.NET.100.165:21
07/29-12:07:56.525800 211.38.95.138:3048 -> MY.NET.156.127:21
```

and

```
06/30-16:34:00.037398 151.164.223.206:4499 -> MY.NET.99.16:21
06/30-16:35:13.626498 151.164.223.206:4500 -> MY.NET.144.59:21
```

>From http://www.sans.org/y2k/062300-1430.htm:

```
John suggests a Snort rule:
alert tcp any any -> $INTERNAL 21 (msg: "Possible wu-ftpd exploit"; content:"/usr/bin/id"; flags: PA;)
```

This attack was attempted and may or may not have succeeded on the 3 hosts above.

IDS127 - TELNET - Login Incorrect

Incorrect Telnet logins are probably just users who forgot their passwords. I'd give this a false positive.

```
08/05-18:54:24.387218 MY.NET.60.11:23 -> 24.6.134.169:3452
08/05-18:59:23.821523 MY.NET.6.7:23 -> 38.30.171.95:1223
08/05-18:56:11.376893 MY.NET.60.8:23 -> 63.24.126.127:1197
08/05-18:53:09.934812 MY.NET.60.8:23 -> 151.198.144.196:1026
08/05-18:47:13.982488 MY.NET.60.8:23 -> 207.172.151.22:1674
08/05-18:47:20.888622 MY.NET.60.8:23 -> 207.172.151.22:1674
08/05-18:37:37.745999 MY.NET.60.11:23 -> 208.198.33.168:1024
```

External RPC call

>From Firewall Forensics:

> Access to portmapper is the first step in scanning a system looking for all the RPC services enabled, such as rpc.mountd, NFS, rpc.statd, rpc.csmd, rpc.ttybd, amd, etc. If the intruder finds the appropriate service enabled, s/he will then run an exploit against the port where the service is running.

This is likely a bad sign and more attention should be paid to it.

```
06/27-01:01:09.457499 207.30.189.91:851 -> MY.NET.6.15:111
06/27-01:01:09.499540 207.30.189.91:851 -> MY.NET.6.15:111
06/30-01:59:39.580221 204.176.11.10:111 -> MY.NET.6.15:111
06/30-01:59:44.793285 204.176.11.10:1556 -> MY.NET.6.15:111
06/30-01:59:44.819258 204.176.11.10:1556 -> MY.NET.6.15:111
06/30-01:59:44.819365 204.176.11.10:1016 -> MY.NET.6.15:111
06/30-01:59:44.848794 204.176.11.10:1016 -> MY.NET.6.15:111
06/30-01:59:44.902365 204.176.11.10:1016 -> MY.NET.6.15:111
```

Watchlist 000222 NET-NCFC

In this section we're just logging everything that comes from NCFC in China because of previous suspicious action (attempted RPC connections). The following is an excerpt of what they're doing now:

```
06/27-02:14:43.603608 aphy.iphy.ac.cn:1237 -> MY.NET.253.43:25
06/27-02:14:44.440409 aphy.iphy.ac.cn:1237 -> MY.NET.253.43:25
06/27-02:14:45.368554 aphy.iphy.ac.cn:113 -> MY.NET.253.43:45197
06/27-02:14:46.165293 aphy.iphy.ac.cn:1237 -> MY.NET.253.43:25
06/27-02:14:46.987142 aphy.iphy.ac.cn:1237 -> MY.NET.253.43:25
06/27-02:14:51.951346 aphy.iphy.ac.cn:1237 -> MY.NET.253.43:25
06/27-02:14:52.768441 aphy.iphy.ac.cn:1237 -> MY.NET.253.43:25
06/27-02:14:54.446714 aphy.iphy.ac.cn:1237 -> MY.NET.253.43:25
06/27-02:14:54.464650 aphy.iphy.ac.cn:1237 -> MY.NET.253.43:25
06/27-02:14:54.466663 aphy.iphy.ac.cn:1237 -> MY.NET.253.43:25
06/27-02:14:55.239825 aphy.iphy.ac.cn:1237 -> MY.NET.253.43:25
06/27-02:14:55.243704 aphy.iphy.ac.cn:1237 -> MY.NET.253.43:25
06/27-02:14:55.916439 aphy.iphy.ac.cn:1237 -> MY.NET.253.43:25
```

They are connecting to 253.43 on SMTP, 253.43 presumably asks them for an auth, and they reply. Then it keeps coming, using the same source port! It may be a retry (I'd have to see the sequence number to know for sure), although it tries a very large number of times. This is also repeated on a variety of machines within the network, suggesting that it is an attempted attack. In this section as well, the attackers try to get to port 80 on MY.NET.100.165.

IDS08 - TELNET - daemon-active

```
08/05-19:03:45.522918 MY.NET.99.51:23 -> nic-25-c111-117.mn.mediaone.net:1029
```

The rule associated with this alert is

```
alert tcp any 23 -> any any (msg:"IDS08 - TELNET - daemon-active";flags:PA; content:"|FF FD 18 FF FD 1F FF FD 23 FF FD 27 FF FD 24|";)
```

Whatever this content is, it's probably a bad sign.

SNMP public access

```
06/30-09:27:45.475626 MY.NET.97.109:1052 -> MY.NET.101.192:161
```

This is one of many coming from our own network to our own network. Depending on the placement of the IDS, this may signify a problem. The main point of this alert is that MY.NET.101.192 may be using the "public" community string, which should be checked and changed if required.

SMB Name Wildcard

This is a very odd scan of 137 from within our own network, always to the same hosts. This warrants some investigation, in particular the placement of the IDS with respect to these hosts. MY.NET.101.192 is also involved in the SNMP public access mystery (above), for which the times interleave with the SMB name wildcard alerts.

```
06/30-09:27:47.890735 MY.NET.101.160:137 -> MY.NET.101.192:137
06/30-09:29:05.193413 MY.NET.101.160:137 -> MY.NET.101.192:137
06/30-09:29:06.691558 MY.NET.101.160:137 -> MY.NET.101.192:137
07/11-14:23:50.181921 MY.NET.70.66:137 -> MY.NET.101.116:137
07/11-15:56:56.690460 MY.NET.101.160:137 -> MY.NET.101.192:137
07/11-15:56:58.079944 MY.NET.101.160:137 -> MY.NET.101.192:137
07/11-16:00:19.379699 MY.NET.101.160:137 -> MY.NET.101.192:137
07/11-16:00:20.866723 MY.NET.101.160:137 -> MY.NET.101.192:137
07/12-16:36:47.188910 MY.NET.70.66:137 -> MY.NET.101.55:137
07/12-17:20:45.412157 MY.NET.70.66:137 -> MY.NET.101.117:137
07/12-18:17:34.633869 MY.NET.101.160:137 -> MY.NET.101.192:137
07/12-18:32:11.853647 MY.NET.101.160:137 -> MY.NET.101.192:137
07/12-18:32:14.856611 MY.NET.101.160:137 -> MY.NET.101.192:137
07/12-18:35:30.556771 MY.NET.101.160:137 -> MY.NET.101.192:137
07/12-18:35:30.571607 MY.NET.101.160:137 -> MY.NET.101.192:137
07/12-18:46:18.501496 MY.NET.101.160:137 -> MY.NET.101.192:137
07/12-18:46:19.960384 MY.NET.101.160:137 -> MY.NET.101.192:137
07/12-19:00:29.220109 MY.NET.101.160:137 -> MY.NET.101.192:137
07/14-08:13:21.447045 MY.NET.101.160:137 -> MY.NET.101.192:137
07/14-08:13:22.972607 MY.NET.101.160:137 -> MY.NET.101.192:137
```

Since they both use port 137 (source and destination), MY.NET.101.160 is a Windows machine (Firewall FAQ). This type of traffic is normal on the Internet, but the odd thing here is that the source and destination are on the same network, and it shouldn't get to an IDS at the border.

IDS246 - MISC - Large ICMP Packet

A large ICMP packet may be used for MTU discovery or ping-of-death. It does not occur normally. These happened at the same time as the ICMP destination unreachable alerts and the target is the same, but the source hosts aren't (although they are also modem-pool-type names).

```
08/05-18:31:47.364699 lu-ppp-1-32.tic.ch: -> MY.NET.70.121:
08/05-18:45:50.534356 h0050e479005f.ne.mediaone.net: -> MY.NET.70.121:
08/05-18:47:38.198198 cx79606-c.vista1.sdca.home.com: -> MY.NET.70.121:
08/05-18:54:22.820323 209.21.59.238: -> MY.NET.70.121:
08/05-19:14:17.790909 user-2ivedlr.dialup.mindspring.com: -> MY.NET.70.121:
```

Watchlist 000220 IL-ISDNNET-990517

Here we're looking for any activity coming from Israel's 212.x.x.x network. Evidently it was a good idea. The following excerpts show a sample of the activity.

```
06/27-06:37:03.434377 212.179.101.218:1219 -> MY.NET.181.88:21
06/27-10:13:49.355998 clnt-5131.bezeqint.net:32852 -> MY.NET.253.24:113
06/27-16:16:49.240815 212.179.30.29:27012 -> MY.NET.181.242:1699
06/28-02:43:30.779113 212.179.23.4:6699 -> MY.NET.179.51:1088
06/28-20:04:04.043157 PT712218.bezeqint.net:21 -> MY.NET.152.10:1964
06/28-20:04:07.870691 PT712218.bezeqint.net:1782 -> MY.NET.152.10:1965
06/29-00:03:44.513085 212.179.58.2:45823 -> MY.NET.253.52:113
06/29-15:37:41.981093 212.179.123.13:6699 -> MY.NET.151.33:1205
06/30-07:28:18.449280 PT712179.bezeqint.net:1349 -> MY.NET.203.190:6346
07/08-13:27:47.451615 212.179.41.218:1032 -> MY.NET.217.114:6688
07/11-11:15:52.590845 212.179.125.114:65046 -> MY.NET.53.28:4807
07/11-12:31:24.535677 212.179.126.8:21 -> MY.NET.110.245:1065
07/17-07:33:57.543265 212.179.27.6:1043 -> MY.NET.53.28:4110
07/17-08:31:47.463838 212.179.29.250:22634 -> MY.NET.179.77:8000
07/17-11:50:58.865132 212.179.4.238:1072 -> MY.NET.53.28:4110
07/26-04:50:11.143506 212.179.54.69:6699 -> MY.NET.182.94:3661
07/27-11:03:39.503471 212.179.126.2:1851 -> MY.NET.6.35:25
07/27-11:03:40.746293 212.179.126.2:113 -> MY.NET.6.35:45712
07/28-12:20:01.792517 clnt-5131.bezeqint.net:57138 -> MY.NET.253.24:113
08/04-19:01:43.812988 212.179.69.68:4362 -> MY.NET.97.225:1432
08/04-19:04:10.301669 212.179.69.68:4362 -> MY.NET.97.225:1448
```

Probable NMAP fingerprint attempt

This packet at least had some strange flags in it (probably SFPU). This particular machine was scanned a lot of the course of the month, but no other scans occurred on the same day.

```
07/12-12:46:34.921774 modemcable045.160-200-24.mtl.mc.videotron.net:1548 -> MY.NET.70.241:8899
```

Scans

Over the course of the month, approximately 129 machines were scanned for vulnerable ports. Host scans covered pretty much the full range of the network. As an example, here are some of the easily-recognizable scans that occurred:

- A Scan for pop3 on every machine on MY.NET.1.255:

```
Jun 27 00:23:42 211.44.13.212:2666 -> MY.NET.1.175:110 SYN **S*****
```

- Host scanning for DNS servers:

```
Jul 17 01:07:05 24.2.123.9:1693 -> MY.NET.1.7:53 UDP
Jul 17 01:07:05 24.2.123.9:1693 -> MY.NET.1.8:53 UDP
Jul 17 01:07:05 24.2.123.9:1693 -> MY.NET.1.9:53 UDP
Jul 17 01:07:05 24.2.123.9:1693 -> MY.NET.1.10:53 UDP
```

- Search for the Subseven trojan, which covered all of MY.NET.

```
Aug  4 05:31:51 24.7.157.43:4409 -> MY.NET.198.1:27374 SYN **S*****
Aug  4 05:31:49 24.7.157.43:4410 -> MY.NET.198.2:27374 SYN **S*****
Aug  4 05:31:49 24.7.157.43:4411 -> MY.NET.198.3:27374 SYN **S*****
Aug  4 05:31:49 24.7.157.43:4412 -> MY.NET.198.4:27374 SYN **S*****
Aug  4 05:31:49 24.7.157.43:4413 -> MY.NET.198.5:27374 SYN **S*****
```

- Scan for anything open on the range 33015 to 39080, one by one:

```
Aug  4 10:55:12 24.23.42.219:1305 -> MY.NET.70.127:33015 SYN **S*****
Aug  4 10:55:12 24.23.42.219:1306 -> MY.NET.70.127:33016 SYN **S*****
Aug  4 10:55:13 24.23.42.219:1307 -> MY.NET.70.127:33017 SYN **S*****
Aug  4 10:55:13 24.23.42.219:1308 -> MY.NET.70.127:33018 SYN **S*****
Aug  4 10:55:13 24.23.42.219:1309 -> MY.NET.70.127:33019 SYN **S*****
Aug  4 10:55:14 24.23.42.219:1310 -> MY.NET.70.127:33020 SYN **S*****
```

- FIN scan on the telnet port to see if it is open. The FIN is used to evade firewalls and the attacker gets a reset if the port is open.

```
Jul 17 12:37:42 213.8.203.144:47850 -> MY.NET.1.1:23 FIN ***F****
Jul 17 12:37:42 213.8.203.144:47850 -> MY.NET.1.2:23 FIN ***F****
Jul 17 12:37:42 213.8.203.144:47850 -> MY.NET.1.3:23 FIN ***F****
Jul 17 12:37:42 213.8.203.144:47850 -> MY.NET.1.4:23 FIN ***F****
Jul 17 12:37:42 213.8.203.144:47850 -> MY.NET.1.5:23 FIN ***F****
Jul 17 12:37:42 213.8.203.144:47850 -> MY.NET.1.7:23 FIN ***F****
Jul 17 12:37:42 213.8.203.144:47850 -> MY.NET.1.11:23 FIN ***F****
```

The snort rules were capable of detecting the scans in the following sections.

Null Scan

Null scans are used for firewall evasion and OS fingerprinting as well as to see if a port is open. There was a noticeable amount of traffic targeting 6699 and 6688 (napster client) ports.

```
06/27-05:17:38.068254 d226-94-105.home.cgocable.net:2584 -> MY.NET.181.88:21
06/27-12:21:36.904991 Sull-15-144.rh.ncsu.edu:6699 -> MY.NET.70.233:1677
06/27-12:21:51.758160 Sull-15-144.rh.ncsu.edu:6699 -> MY.NET.70.233:1677
06/27-12:25:01.004914 Sull-15-144.rh.ncsu.edu:6699 -> MY.NET.70.233:1678
06/27-14:14:36.880429 24.64.178.158.on.wave.home.com:6699 -> MY.NET.162.200:3211
06/27-14:15:08.507108 24.64.178.158.on.wave.home.com:6699 -> MY.NET.162.200:3211
06/27-15:44:30.205306 p219.vub.ac.be:3353 -> MY.NET.110.249:6346
06/27-15:53:07.502643 p219.vub.ac.be:3353 -> MY.NET.110.249:6346
06/28-13:15:18.775587 cr559257-a.crdva1.bc.wave.home.com:1634 -> MY.NET.110.249:6346
06/28-16:16:47.836776 p218.vub.ac.be:1513 -> MY.NET.110.249:6346
06/29-08:47:39.937151 dahgy97.lyan.vxu.se:6699 -> MY.NET.162.200:4160
06/29-12:04:16.039097 24.64.151.88.on.wave.home.com:3781 -> MY.NET.217.62:6699
06/29-20:51:06.451205 cc19041-b.vron1.nj.home.com:6688 -> MY.NET.218.38:1702
06/30-16:55:44.640359 b09253.impsat.com.br:61504 -> MY.NET.97.31:2340
07/08-00:15:11.772935 HSE-Ottawa-ppp84175.sympatico.ca:3763 -> MY.NET.217.178:1264
07/08-11:21:56.262183 cable-195-162-199-244.upc.chello.be:1741 -> MY.NET.130.65:6699
07/08-18:32:13.226827 cap175-215-55.pixi.net:1079 -> MY.NET.110.249:6346
07/08-19:17:15.135129 dhcp044.43.lvcm.com:1130 -> MY.NET.70.241:8899
07/10-03:06:24.370205 isu244070.ilstu.edu:1691 -> MY.NET.217.114:6688
07/10-04:36:15.676792 www.webnl.com:27960 -> MY.NET.20.10:27960
07/10-18:11:11.962424 hellenberg.demon.nl:7788 -> MY.NET.60.14:2218
07/10-18:15:51.778974 hellenberg.demon.nl:27970 -> MY.NET.60.14:27960
07/10-18:19:04.143272 hellenberg.demon.nl:27970 -> MY.NET.60.14:27960
07/11-14:15:45.364250 mkc-31-235-77.kc.rr.com:1092 -> MY.NET.217.174:6699
07/11-14:42:18.045118 www.grcdi.nl:3780 -> MY.NET.20.10:1375
07/11-15:00:26.636634 63.145.43.162:1671 -> MY.NET.110.249:6346
07/11-15:31:18.493853 wh4-422.st.Uni-Magdeburg.DE:3375 -> MY.NET.70.227:6699
07/12-01:23:39.949103 24.65.64.15.on.wave.home.com:6699 -> MY.NET.217.162:2172
07/12-05:16:01.431555 ca-ol-marseille-4-120.abo.wanadoo.fr:1721 -> MY.NET.188.32:8080
07/12-06:46:44.185216 ipcwo227.Chemie.Uni-Mainz.DE:1658 -> MY.NET.110.249:6346
07/12-13:16:01.954025 wh4-422.st.Uni-Magdeburg.DE:1644 -> MY.NET.70.227:6699
07/12-13:36:19.825612 wh4-422.st.Uni-Magdeburg.DE:1870 -> MY.NET.70.227:6699
07/12-18:58:16.425144 cc931590-a.chstfld1.va.home.com:1778 -> MY.NET.106.190:6699
07/12-18:58:24.335726 cc931590-a.chstfld1.va.home.com:128 -> MY.NET.106.190:1778
07/12-18:58:38.347294 cc931590-a.chstfld1.va.home.com:1778 -> MY.NET.106.190:6699
07/12-18:58:38.535136 cc931590-a.chstfld1.va.home.com:1778 -> MY.NET.106.190:6699
07/12-20:01:14.457600 OL49-48.fibertel.com.ar:1172 -> MY.NET.70.227:6699
07/12-20:08:27.345127 OL49-48.fibertel.com.ar:1172 -> MY.NET.70.227:6699
07/12-20:12:44.421116 OL49-48.fibertel.com.ar:1172 -> MY.NET.70.227:6699
07/12-21:10:42.866362 bvh1-9930.twcny.rr.com:1042 -> MY.NET.110.57:6346
07/14-06:25:31.061532 N170P011.adsl.highway.telekom.at:1124 -> MY.NET.53.28:4807
07/14-10:57:04.136606 212.58.186.14:1053 -> MY.NET.110.11:6688
07/14-12:28:25.838842 OL137-51.fibertel.com.ar:1152 -> MY.NET.110.57:6688
07/14-12:28:29.384871 OL137-51.fibertel.com.ar:1152 -> MY.NET.110.57:6688
07/17-17:12:17.516669 cx385471-a.vista1.sdca.home.com:1380 -> MY.NET.217.46:6688
07/17-18:57:44.862726 216.230.130.39:1325 -> MY.NET.70.241:6688
07/19-14:55:58.535261 cx441045-d.ports1.ri.home.com:2340 -> MY.NET.97.116:1371
```

```
07/19-15:07:50.644716 cx441045-d.ports1.ri.home.com:2340 -> MY.NET.97.116:1375
07/19-15:08:46.927923 cx441045-d.ports1.ri.home.com:2340 -> MY.NET.97.116:1375
07/19-17:29:08.539527 bessy.physik.TU-Berlin.DE:1037 -> MY.NET.217.18:994
07/19-17:30:43.860329 bessy.physik.TU-Berlin.DE:1038 -> MY.NET.217.18:994
07/19-18:30:12.847020 bessy.physik.TU-Berlin.DE:1139 -> MY.NET.217.18:994
07/26-14:22:26.445748 cgmd72249.chello.nl:7777 -> MY.NET.70.241:1163
07/26-15:24:39.163260 AC8A25B3.ipt.aol.com:15091 -> MY.NET.253.112:443
07/26-17:04:39.151326 modemcable075.16-200-24.que.mc.videotron.net:1453 -> MY.NET.69.28:6688
07/27-11:15:24.014984 pc237033.sbdc.iastate.edu:6699 -> MY.NET.110.57:4463
07/28-06:45:33.156153 212.4.207.26:1649 -> MY.NET.100.236:6346
07/28-08:50:47.204381 dialin236-28-c5800do2.sonnet.de:1200 -> MY.NET.100.236:6346
07/28-09:01:45.652363 dialin236-28-c5800do2.sonnet.de:1200 -> MY.NET.100.236:6346
07/28-10:05:54.003858 ras10-p230.tlv.netvision.net.il:1440 -> MY.NET.100.236:6346
07/28-10:30:11.805484 A7b0c.pppool.de:6699 -> MY.NET.182.94:3419
07/28-12:39:47.189229 144.41.242.202:1102 -> MY.NET.97.210:6699
07/28-13:12:43.075930 avenger.resnet.tamu.edu:1099 -> MY.NET.110.57:6688
07/28-13:46:39.184433 dialin236-28-c5800do2.sonnet.de:1200 -> MY.NET.100.236:6346
07/28-17:56:03.497735 d6-91.hntnny.optonline.net:6699 -> MY.NET.182.94:4179
07/28-17:57:55.558274 d6-91.hntnny.optonline.net:6699 -> MY.NET.182.94:4180
07/28-18:50:41.884795 cable20-209.gte.net:6699 -> MY.NET.97.235:2942
07/28-23:32:23.368753 216.127.150.136:57878 -> MY.NET.253.114:22
07/29-01:32:39.735686 cr4314-a.rct1.bc.wave.home.com:6699 -> MY.NET.98.166:2757
07/29-02:32:54.058776 AFN-Dynamic-220122.ashlandfiber.net:6688 -> MY.NET.98.166:1055
07/29-03:44:14.119096 cj61321-a.reston1.va.home.com:1963 -> MY.NET.100.236:6346
07/29-06:27:46.015408 modem-13.zinc.dialup.pol.co.uk:1418 -> MY.NET.100.236:6346
07/29-06:47:51.467109 210.121.242.164:2929 -> MY.NET.100.236:6346
07/29-07:08:46.726685 210.121.242.164:2929 -> MY.NET.100.236:6346
07/29-08:28:17.191946 cable-195-162-218-218.upc.chello.be:2494 -> MY.NET.100.236:6346
07/29-08:31:53.342906 cable-195-162-218-218.upc.chello.be:2494 -> MY.NET.100.236:6346
07/29-08:50:53.643598 cable-195-162-218-218.upc.chello.be:2494 -> MY.NET.100.236:6346
07/29-08:52:32.321104 210.121.242.164:2929 -> MY.NET.100.236:6346
07/29-10:11:28.919158 210.121.242.164:2929 -> MY.NET.100.236:6346
07/29-10:12:10.718720 210.121.242.164:2929 -> MY.NET.100.236:6346
07/29-10:14:15.538211 210.121.242.164:2929 -> MY.NET.100.236:6346
07/30-00:09:06.027872 12.78.254.76:49619 -> MY.NET.223.105:23005
07/30-00:10:39.462879 12.78.254.76:37901 -> MY.NET.179.59:55925
07/30-00:11:01.605623 12.78.254.76:26232 -> MY.NET.210.8:2637
07/30-00:12:30.390194 12.78.254.76:43104 -> MY.NET.54.24:26252
07/30-18:00:41.863241 NIC-23-20.RESNET.UPENN.EDU:6699 -> MY.NET.217.38:1855
07/30-22:21:29.543423 spnp46173.spnp.nus.edu.sg:1337 -> MY.NET.217.38:6699
08/03-14:58:43.996407 d141-240-10.home.cgocable.net:4771 -> MY.NET.53.28:4362
08/03-16:47:29.190205 cc290147-a.abdn1.md.home.com:4805 -> MY.NET.253.112:443
08/03-19:59:23.729894 pec-111-69.tnt2.h2.uunet.de:7904 -> MY.NET.60.14:7
08/03-19:59:23.774250 pec-111-69.tnt2.h2.uunet.de:7904 -> MY.NET.60.14:22
08/03-19:59:23.823304 pec-111-69.tnt2.h2.uunet.de:7904 -> MY.NET.60.14:37
08/03-19:59:23.877719 pec-111-69.tnt2.h2.uunet.de:7904 -> MY.NET.60.14:137
08/03-19:59:23.971660 pec-111-69.tnt2.h2.uunet.de:7904 -> MY.NET.60.14:513
08/03-23:13:21.332823 customer-GDL-199-178.megared.net.mx:1449 -> MY.NET.53.28:4362
08/04-13:14:48.162932 xrpc01.quimica.uniovi.es:1064 -> MY.NET.146.68:6699
08/04-23:30:54.234513 m3bhNs1n8.midsouth.rr.com:3676 -> MY.NET.217.214:6699
```

Nmap TCP Ping

We have here a whole bunch of people looking for DNS on MY.NET.1.8 and MY.NET.1.9 (a TCP ping is a TCP ACK packet):

```
06/27-07:39:28.385752 ATHM-209-218-xxx-46.Home.net:80 -> MY.NET.1.8:53
06/27-07:39:28.388448 ATHM-209-218-xxx-46.Home.net:53 -> MY.NET.1.8:53
06/27-07:39:33.390475 ATHM-209-218-xxx-46.Home.net:80 -> MY.NET.1.8:53
06/27-07:39:33.390629 ATHM-209-218-xxx-46.Home.net:53 -> MY.NET.1.8:53
06/27-07:51:18.494768 wsd-vccsc.framfab.se:80 -> MY.NET.1.9:53
06/27-07:51:18.494815 wsd-vccsc.framfab.se:53 -> MY.NET.1.9:53
06/27-07:51:23.472464 wsd-vccsc.framfab.se:80 -> MY.NET.1.9:53
06/27-07:51:23.472859 wsd-vccsc.framfab.se:53 -> MY.NET.1.9:53
06/30-05:34:58.313594 ATHM-209-218-xxx-46.Home.net:80 -> MY.NET.1.8:53
06/30-05:34:58.315836 ATHM-209-218-xxx-46.Home.net:53 -> MY.NET.1.8:53
06/30-06:41:26.631247 ATHM-209-218-xxx-201.Home.net:80 -> MY.NET.1.8:53
06/30-06:41:26.631294 ATHM-209-218-xxx-201.Home.net:53 -> MY.NET.1.8:53
06/30-08:26:02.939759 195.25.86.2:80 -> MY.NET.60.14:80
07/08-20:02:37.444826 ATHM-209-218-xxx-46.Home.net:80 -> MY.NET.1.8:53
07/08-20:02:37.447766 ATHM-209-218-xxx-46.Home.net:53 -> MY.NET.1.8:53
07/11-10:12:48.811332 wsd-vccsc.framfab.se:53 -> MY.NET.1.8:53
07/27-02:54:34.936909 ATHM-209-218-xxx-46.Home.net:80 -> MY.NET.1.8:53
07/27-02:54:39.888327 ATHM-209-218-xxx-46.Home.net:80 -> MY.NET.1.8:53
07/27-02:54:39.888376 ATHM-209-218-xxx-46.Home.net:53 -> MY.NET.1.8:53
07/28-23:32:23.408944 216.127.150.136:57882 -> MY.NET.253.114:1 <== don't forget this SGI fingerprint!
08/04-08:01:02.191197 195.25.86.2:80 -> MY.NET.179.77:80        <== or this web scan!
08/04-10:49:10.811041 205.128.11.157:80 -> MY.NET.1.8:53
08/04-10:49:10.811088 205.128.11.157:53 -> MY.NET.1.8:53
08/04-11:18:28.348261 205.128.11.157:80 -> MY.NET.1.8:53
08/04-11:18:28.348302 205.128.11.157:53 -> MY.NET.1.8:53
08/04-12:43:24.357621 205.128.11.157:80 -> MY.NET.1.8:53
08/04-12:43:24.357665 205.128.11.157:53 -> MY.NET.1.8:53
08/04-16:35:29.378054 205.128.11.157:80 -> MY.NET.1.9:53
08/04-16:35:29.378100 205.128.11.157:53 -> MY.NET.1.9:53
08/04-16:35:34.350598 205.128.11.157:80 -> MY.NET.1.9:53
08/04-16:35:34.350643 205.128.11.157:53 -> MY.NET.1.9:53
08/05-06:19:42.926608 ATHM-209-218-xxx-46.Home.net:80 -> MY.NET.1.8:53
08/05-06:19:42.927973 ATHM-209-218-xxx-46.Home.net:53 -> MY.NET.1.8:53
08/05-08:14:04.746341 205.128.11.157:80 -> MY.NET.1.8:53
08/05-08:14:04.746385 205.128.11.157:53 -> MY.NET.1.8:53
08/05-08:14:09.746384 205.128.11.157:80 -> MY.NET.1.8:53
08/05-08:14:09.746451 205.128.11.157:53 -> MY.NET.1.8:53
08/05-13:21:31.346624 205.128.11.157:80 -> MY.NET.1.8:53
```

```
08/05-21:47:55.033677 205.128.11.157:80 -> MY.NET.1.8:53
08/05-21:47:55.033727 205.128.11.157:53 -> MY.NET.1.8:53
08/05-21:48:00.007043 205.128.11.157:80 -> MY.NET.1.8:53
08/05-21:48:00.007089 205.128.11.157:53 -> MY.NET.1.8:53
08/05-23:25:52.964622 205.128.11.157:80 -> MY.NET.1.8:53
08/05-23:25:52.964687 205.128.11.157:53 -> MY.NET.1.8:53
08/05-23:25:57.962201 205.128.11.157:80 -> MY.NET.1.8:53
08/05-23:25:57.962266 205.128.11.157:53 -> MY.NET.1.8:53
```

SYN-FIN scan!

202.0.178.98 (China Motion Telcom Holdings Ltd.) is looking for a vulnerable OS with domain open, probably wanting to use a specific exploit on it. They scanned much of MY.NET.

```
06/28-06:52:48.291107 202.0.178.98:53 -> MY.NET.1.1:53
06/28-06:52:48.330764 202.0.178.98:53 -> MY.NET.1.3:53
06/28-06:52:48.518996 202.0.178.98:53 -> MY.NET.1.11:53
06/28-06:52:48.522936 202.0.178.98:53 -> MY.NET.1.12:53
06/28-06:52:48.552072 202.0.178.98:53 -> MY.NET.1.14:53
06/28-06:52:48.598539 202.0.178.98:53 -> MY.NET.1.15:53
06/28-06:52:48.647841 202.0.178.98:53 -> MY.NET.1.18:53
06/28-06:52:48.761524 202.0.178.98:53 -> MY.NET.1.23:53
06/28-06:52:48.814427 202.0.178.98:53 -> MY.NET.1.25:53
06/28-06:52:48.852929 202.0.178.98:53 -> MY.NET.1.27:53
06/28-06:52:48.920904 202.0.178.98:53 -> MY.NET.1.30:53
06/28-06:52:48.920953 202.0.178.98:53 -> MY.NET.1.31:53
06/28-06:52:48.929440 202.0.178.98:53 -> MY.NET.1.32:53
06/28-06:52:48.931332 202.0.178.98:53 -> MY.NET.1.33:53
06/28-06:52:48.957975 202.0.178.98:53 -> MY.NET.1.34:53
06/28-06:52:49.089852 202.0.178.98:53 -> MY.NET.1.38:53
06/28-06:52:49.105091 202.0.178.98:53 -> MY.NET.1.40:53
```

Assignment 4: Analysis Process

The SnortA*.txt files were copied into filenames which reflected the date they were collected. These files were concatenated into one large file. The tool snort_sort.pl was used to analyze this large file; it grouped the events by the titles given in the snort ruleset. A minor modification was made to the script: the subroutine by_IP was snarfed from SHADOW's sort_and_resolve script to further sort the data by source address (or by destination address, as needed). In doing so, the data can be viewed separated by event type, time and source or destination IP. I created 3 html files: one sorted by source IP, one by destination IP, and one unsorted but with the IPs resolved to hostnames. These 3 files allowed me to better visualize patterns.

I also wrote a crude perl script to get statistics on the ICMP Destination Unreachable and Time Exceeded data. The time exceeded analysis code follows:

```
#!/usr/bin/perl
#
# Filename:     icmptimexstats.pl

if($ARGV[0] eq undef)
{
   print STDERR "USAGE: icmptimexstats <filename>\n";
   exit;
}

open(INFILE,"< $ARGV[0]") || die "Unable to open file $ARGV[0]\n";

$i=0;
while(<INFILE>) {
  chomp();
  # split the alert apart
  @fields = split(/\s+/ , "$_");
  $date[$i] = $fields[1];
  $src[$i] = $fields[6];
  $dst[$i] = $fields[8];
  $i++;
}
close(INFILE);

$size = @date;
$j=0;
for ( $i = 0 ; $i < $size ; $i++ ) {
    if ($dst[$i] eq "MY.NET.140.9") {
        $srclist[$j] = $src[$i];
        $j++;
    }
}

@sorted = sort @srclist;

$size = @srclist;
$last = $sorted[0];
$bin = 0 ;
$name[$bin] = $last;
for ( $i = 0 ; $i < $size ; $i++ ) {
    if ($sorted[$i] eq $name[$bin]) {
        $n[$bin]++;
    } else {
        $bin++;
        $n[$bin]++;
        $name[$bin] = $sorted[$i];
```

```
    }
    $last = $sorted[$i];
}

# Output data in HTML tabular format
$size = @n;
for ( $i = 0 ; $i < $size ; $i++ ) {
    $j=$i%4;
    if ($j == 0) {
        print "<tr><td> $name[$i] </td><td> $n[$i]</td>\n";
    } elsif ($j == 3) {
        print "<td> $name[$i] </td><td> $n[$i]</td></tr>\n";
    } else {
        print "<td> $name[$i] </td><td> $n[$i]</td>\n";
    }
}
```

I left the scans for last. The SnortA*.txt data tells you there was a "portscan", and where it originated, but it doesn't differentiate between host and port scans. I wrote perl scripts to separate port scans from host scans, and then generated statistics from those. The script for separating the two types of scans follows. The input file is data obtained by *grep -i "end of portscan" concatA.txt > endofportscan.txt*, where concatA.txt is the file of concatenated events.

```
#!/usr/bin/perl
#
# Filename:     portscan.pl

# This one separates portscans from hostscans and other stuff and outputs
# them into 2 different files for later analysis. (uses endofportscan.txt)

if($ARGV[0] eq undef)
{
   print STDERR "USAGE: portscan <filename>\n";
   exit;
}

open(INFILE,"< $ARGV[0]") || die "Unable to open file $ARGV[0]\n";
@stuff = <INFILE>;
close(INFILE);



$size = @stuff;
$i=0;
foreach $line (@stuff) {
    chomp($line);
    @fields = split(/\s+/ , "$line");
    if ($fields[4] eq "->") {
        @temp = split(/:/, $fields[3]);
        $srchost[$i] = $temp[0];
        @temp = split(/:/, $fields[5]);
        $dsthost[$i] = $temp[0];
        $lines[$i] = $line;
        $i++;
    }
}

# separate the hostscans from the portscans

open (PORTSCANS, ">portscans.out");
open (HOSTSCANS, ">hostscans.out");

$size = $i;
for ($i=1; $i<=$size; $i++) {
    if ($srchost[$i] eq $srchost[$i-1]) {
        # it's in the same group
        # compare the destination host
        if ($dsthost[$i] eq $dsthost[$i-1]) {
            # both the source and dst hosts are the same.  Conclude this is a
            # portscan on the destination host.
            print PORTSCANS "$lines[$i-1]\n";
        } else {
            print HOSTSCANS "$lines[$i-1]\n";
        }
    } else {
        print PORTSCANS "$lines[$i-1]\n" if $i eq 1;
    }
}

close (PORTSCANS);
close (HOSTSCANS);
```