SANS GIAC Intrusion Detection Curriculum Practical Assignment for SANS Parliament Hill

Alex Arndt – 22 September 2000

Assignment 1 – Network Detects

Detect 1

```
10:04:10.493001 63.145.81.31.4843 > carhost.my.net.23: S 2314119657:2314119657(0) win
32120  (DF)
10:04:10.514896 63.145.81.31.4844 > narhost.my.net.23: S 2313821603:2313821603(0) win
32120  (DF)
10:04:10.810478 63.145.81.31.4901 > necronomicon.my.net.23: S 2319348191:2319348191(0)
win 32120  (DF)
10:04:10.810556 63.145.81.31.4902 > moira.my.net.23: S 2326587546:2326587546(0) win 32120
(DF)
10:04:10.817847 63.145.81.31.4920 > mail.my.net.23: S 2315396834:2315396834(0) win 32120
(DF)
10:04:10.832466 63.145.81.31.4924 > pure-data1.my.net.23: S 2316889436:2316889436(0) win
32120  (DF)
10:04:10.832543 63.145.81.31.4925 > news.my.net.23: S 2320677097:2320677097(0) win 32120
(DF)
[...]
10:04:16.614802 63.145.81.31.2655 > ts04-p10.ppp.my.net.23: S 2322107479:2322107479(0)
win 32120  (DF)
10:04:16.622052 63.145.81.31.2661 > ts04-p12.ppp.my.net.23: S 2329919233:2329919233(0)
win 32120  (DF)
10:04:16.622130 63.145.81.31.2662 > ts04-p13.ppp.my.net.23: S 2318560253:2318560253(0)
win 32120  (DF)
10:04:16.622209 63.145.81.31.2663 > ts04-p14.ppp.my.net.23: S 2324641439:2324641439(0)
win 32120  (DF)
10:04:16.622289 63.145.81.31.2664 > 10.0.223.255.23: S 2324916423:2324916423(0) win 32120
(DF)
10:04:16.622366 63.145.81.31.2660 > ts04-p11.ppp.my.net.23: S 2319122704:2319122704(0)
win 32120  (DF)
[...TELNET scan ends, IMAP scan begins...]
10:04:16.782577 63.145.81.31.4463 > carchost-gw1.my.net.143: S 2332985184:2332985184(0)
win 32120  (DF)
10:04:16.782653 63.145.81.31.4464 > broadcast.143: S 2325595639:2325595639(0) win 32120
(DF)
10:04:16.819074 63.145.81.31.4504 > testbed1.my.net.143: S 2333649952:2333649952(0) win
32120  (DF)
10:04:16.822738 63.145.81.31.4503 > attila.my.net.143: S 2329509335:2329509335(0) win
32120  (DF)
10:04:16.837340 63.145.81.31.4515 > testbed2.my.net.143: S 2333549196:2333549196(0) win
32120  (DF)
[...]
10:04:21.617782 63.145.81.31.1710 > 10.0.223.255.143: S 2336068679:2336068679(0) win
32120  (DF)
10:04:21.621445 63.145.81.31.1711 > ts04-p13.ppp.my.net.143: S 2331174801:2331174801(0)
win 32120  (DF)
10:04:21.621524 63.145.81.31.1712 > ts04-p14.ppp.my.net.143: S 2321866709:2321866709(0)
win 32120  (DF)
```

```
10:04:21.621601 63.145.81.31.1713 > ts04-p12.ppp.my.net.143: S 2324346777:2324346777(0)
win 32120  (DF)
10:04:21.621680 63.145.81.31.1715 > ts04-p11.ppp.my.net.143: S 2322535423:2322535423(0)
win 32120  (DF)
```

1. Source of Trace:

Class-C network provided to Employer by local ISP. NOTE – All destination IP addresses have been sanitized. The protected addresses are presented as 10.0.223.0/24 or '*hostname*.my.net'.

2. Detect was generated by:

SHADOW

Since this appears to be a port scan, the primary fields of interest in this case would be the source IP, the source port, the destination IP and the destination port. The time field must also be considered, as it is a good indicator of what method is being used to perform the attack (i.e. tool, script, etc.).

Below is a description of the logging fields for the SHADOW extract:

```
|       A      | |    B      | |C |   |     D    | |E| |F|
10:04:18.531749 192.168.81.31.1563 > 10.0.223.215.143: S
|         G        |     |    H   | | I|
2333129333:2333129333(0) win 32120  (DF)

A – Time (HH:MM:SS.ssssss)
B – Source IP
C – Source Port
D - Destination IP
E - Destination Port
F – SYN Flag (May also be R=RESET, F=FIN or P=PUSH), indicates a TCP connection
G – SYN Sequence Number
H – The WIN (Window) Size
I – DF or 'Don't Fragment' Flag (May also be MF=More Fragments if packets are fragmented)
```

3. Probability the source address was spoofed:

The initiator must receive any responses for this port scan to be useful therefore spoofing is unlikely.

4. Description of Attack:

Network port scans against TCP port 23 TELNET and TCP port 143 IMAP.

TELNET has several well-known vulnerabilities, which were noted in the following CVE database entries: CVE-1999-0073, CVE-1999-0192, CVE-1999-0230 and CVE-2000-0268

IMAP also has well-known vulnerabilities, which were noted in the following CVE database entries: CVE-1999-0005, CVE-1999-0042, CVE-1999-0920 and CVE-2000-0053
.

5. Attack Mechanism:

This scan is most likely the result of an automated script and can be considered to be a fairly simple one. The packets generated are not crafted and the source ports and sequence numbers increment in a normal manner. Furthermore the TCP port scan is a simple SYN scan. There is no attempt to hide the scan and no anomalous flags are set in an attempt to evade detection or to provoke responses that may provide information about the targeted host's operating system or the version of the service running on the targeted port (i.e. – OS fingerprinting).

The idea is to send a TCP SYN packet destined for TCP port 23 TELNET or TCP port 143 IMAP in an attempt to determine which hosts in the targeted IP range is offering these services.

All the addresses within the targeted range are first sequentially scanned for the TELNET service first and then scanned again in the same order for the IMAP service. Any system that replies with a SYN/ACK can be assumed to be active and running the one of these services. This reconnaissance information could be used to help facilitate a future attack.

Any system identified during the scan as running TELNET or IMAP could be exploited later on at the attacker's leisure. By using one of the well-known vulnerabilities, the attacker could gain privileged user access or cause a 'denial of service' against the targeted service. This can occur immediately after the reconnaissance effort ends, or whenever the attacker who performed the mapping feels is appropriate. The actual attack on the individual services (TELNET, IMAP) would probably require the use of additional tools or scripts.

6. Correlation:

Both a Cisco SecureIDS sensor and a SHADOW sensor detected this scan. The SHADOW output was used to analyze the attack, while the Cisco SecureIDS initially brought the scan to attention. Finally, TCPdump was used on the raw SHADOW logs to ensure that no system had replied to the scan, and to guarantee that no TCP connections were in fact established.

This particular scan combination of TELNET and IMAP has apparently not been noted before, however there are many well-known vulnerabilities in both TELNET and IMAP. IMAP is even listed as #9 in the "Ten Most Critical Internet Security Threats" list on the SANS web site. This list is available at http://www.sans.org/topten.htm.

7. Evidence of active targeting:

Since this scan involved an entire Class-C address space, it would appear to be quite deliberate. All hosts within the 10.0.223.0/24 range were scanned for both the TELNET and IMAP services.

8. Severity:

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

(4+2) – (4+5) = -3

- This Class-C network contains a web server, a mail server, a news server and several workstations and dial-up connections (4);
- Attack is a network scan (2);
- These systems are a mix of several flavors of Linux, Solaris 2.6 (or higher) and Windows NT 4.0. All of these hosts are reasonably well maintained and are patched with the latest recommended patches, clusters or service packs (4);
- Network is protected by two Argus firewalls and a border router. In addition, a Cisco SecureIDS sensor and SHADOW sensor monitor this network from outside the exterior firewall and border router (5).

9. Defensive recommendation:

Defenses are fine, as the attack was blocked at the firewall and detected by both the SHADOW and Cisco SecureIDS sensors. Because of the layered defence, which was verified using TCPdump during analysis, it is safe to state that no host was connected to via either of the targeted TCP ports.

The web-service servers on this network are outside of the perimeter defenses and are directly reachable from the Internet, therefore these systems are the most vulnerable. Each of these servers is Linux-based and running IPchains, so vulnerable services are properly wrapped to allow only authorized connections. In addition, the mail server is not running IMAP. Due to these considerations, their defenses are also fine.

10. Multiple choice test question:

Given the following trace, determine the purpose of the attack:

```
10:04:10.493001 192.168.81.31.4843 > host.1.my.net.23: S
2314119657:2314119657(0) win 32120  (DF)
[...]
10:04:16.622366 192.168.81.31.2660 > host.254.my.net.23: S
2319122704:2319122704(0) win 32120  (DF)
10:04:16.782577 192.168.81.31.4463 > host.1.my.net.143: S
2332985184:2332985184(0) win 32120  (DF)
[...]
10:04:21.621680 192.168.81.31.1715 > host.254.my.net.143: S
2322535423:2322535423(0) win 32120  (DF)
```

a) TELNET scan
b) IMAP scan
c) TCP SYN scan
d) Attempt to gain privileged user access

Answer: c – Because the scan involves BOTH TELNET and IMAP, the most correct answer is TCP SYN scan

Detect 2

```
[ICMP netsweep begins...)
00:34:01.630996 207.253.109.179 > 10.0.255.1: icmp: echo request (ttl 244, id 16113)
[...]
00:34:09.152663 207.253.109.179 > 10.0.255.9: icmp: echo request (ttl 244, id 16121)
[...ICMP netsweep ends, UDP port 137 scan begins...]
00:34:12.890307 207.253.109.179.1025 > 10.0.255.3.137: udp 50 (ttl 117, id 16125)
00:34:14.379938 207.253.109.179.1025 > 10.0.255.3.137: udp 50 (ttl 117, id 16126)
00:34:15.883046 207.253.109.179.1025 > 10.0.255.3.137: udp 50 (ttl 117, id 16127)
[...]
00:34:55.478148 207.253.109.179.1025 > 10.0.255.9.137: udp 50 (ttl 117, id 16147)
00:34:56.969740 207.253.109.179.1025 > 10.0.255.9.137: udp 50 (ttl 117, id 16148)
00:34:58.472476 207.253.109.179.1025 > 10.0.255.9.137: udp 50 (ttl 117, id 16150)
00:35:00.164826 207.253.109.179.1025 > 10.0.255.9.137: udp 50 (ttl 117, id 16152)
00:35:01.657409 207.253.109.179.1025 > 10.0.255.9.137: udp 50 (ttl 117, id 16153)
00:35:03.159605 207.253.109.179.1025 > 10.0.255.9.137: udp 50 (ttl 117, id 16154)
[...UDP port 137 scan ends, TCP SYN scan begins...]
00:35:07.865911 207.253.109.179.3794 > 10.0.255.1.7: S 500486874:500486874(0) win 8760
<mss 1460,nop,nop,sackOK> (DF) (ttl 117, id 16156)
00:35:07.945225 207.253.109.179.3795 > 10.0.255.1.15: S 500535061:500535061(0) win 8760
<mss 1460,nop,nop,sackOK> (DF) (ttl 117, id 16157)
00:35:07.996126 207.253.109.179.3796 > 10.0.255.1.21: S 500608305:500608305(0) win 8760
<mss 1460,nop,nop,sackOK> (DF) (ttl 117, id 16158)
00:35:08.046142 207.253.109.179.3797 > 10.0.255.1.23: S 500662425:500662425(0) win 8760
<mss 1460,nop,nop,sackOK> (DF) (ttl 117, id 16159)
00:35:08.144886 207.253.109.179.3799 > 10.0.255.1.37: S 500779994:500779994(0) win 8760
<mss 1460,nop,nop,sackOK> (DF) (ttl 117, id 16161)
00:35:08.196504 207.253.109.179.3800 > 10.0.255.1.43: S 500870380:500870380(0) win 8760
<mss 1460,nop,nop,sackOK> (DF) (ttl 117, id 16162)
00:35:08.245003 207.253.109.179.3801 > 10.0.255.1.53: S 500923999:500923999(0) win 8760
<mss 1460,nop,nop,sackOK> (DF) (ttl 117, id 16163)
00:35:08.298571 207.253.109.179.3802 > 10.0.255.1.69: S 501000129:501000129(0) win 8760
<mss 1460,nop,nop,sackOK> (DF) (ttl 117, id 16164)
```

```
00:35:08.346573 207.253.109.179.3803 > 10.0.255.1.70: S 501039010:501039010(0) win 8760
<mss 1460,nop,nop,sackOK> (DF) (ttl 117, id 16165)
00:35:08.395245 207.253.109.179.3804 > 10.0.255.1.79: S 501119844:501119844(0) win 8760
<mss 1460,nop,nop,sackOK> (DF) (ttl 117, id 16166)
00:35:08.446191 207.253.109.179.3805 > 10.0.255.1.80: S 501156559:501156559(0) win 8760
<mss 1460,nop,nop,sackOK> (DF) (ttl 117, id 16167)
00:35:08.447100 10.0.255.1.80 > 207.253.109.179.3805: R 0:0(0) ack 501156560 win 0 (ttl
255, id 64607)
00:35:08.497314 207.253.109.179.3806 > 10.0.255.1.110: S 501216066:501216066(0) win 8760
<mss 1460,nop,nop,sackOK> (DF) (ttl 117, id 16168)
00:35:08.547488 207.253.109.179.3807 > 10.0.255.1.119: S 501263143:501263143(0) win 8760
<mss 1460,nop,nop,sackOK> (DF) (ttl 117, id 16169)
00:35:08.595615 207.253.109.179.3808 > 10.0.255.1.143: S 501335385:501335385(0) win 8760
<mss 1460,nop,nop,sackOK> (DF) (ttl 117, id 16170)
00:35:08.645786 207.253.109.179.3809 > 10.0.255.1.6667: S 501368314:501368314(0) win 8760
<mss 1460,nop,nop,sackOK> (DF) (ttl 117, id 16171)
[...scan repeats x2]
[TCP SYN scan for 10.0.255.1 ends, scan for 10.0.255.2 begins...]
00:36:20.386243 207.253.109.179.3810 > 10.0.255.2.7: S 519335647:519335647(0) win 8760
<mss 1460,nop,nop,sackOK> (DF) (ttl 117, id 16207)
[...]
00:36:30.084403 207.253.109.179.3825 > 10.0.255.2.6667: S 520212374:520212374(0) win 8760
<mss 1460,nop,nop,sackOK> (DF) (ttl 117, id 16255)
00:36:30.089475 207.253.109.179.3824 > 10.0.255.2.143: S 520167579:520167579(0) win 8760
<mss 1460,nop,nop,sackOK> (DF) (ttl 117, id 16256)
[...scan repeats x2]
[TCP SYN scan for 10.0.255.2 ends, scan for 10.0.255.9 begins...]
00:37:09.037133 207.253.109.179.3828 > 10.0.255.9.7: S 532234352:532234352(0) win 8760
<mss 1460,nop,nop,sackOK> (DF) (ttl 117, id 16266)
[...]
00:37:12.746974 207.253.109.179.3843 > 10.0.255.9.6667: S 533198215:533198215(0) win 8760
<mss 1460,nop,nop,sackOK> (DF) (ttl 117, id 16299)
00:37:12.751207 207.253.109.179.3842 > 10.0.255.9.143: S 533145951:533145951(0) win 8760
<mss 1460,nop,nop,sackOK> (DF) (ttl 117, id 16300)
[...scan repeats x2]
[...TCP SYN scan for 10.0.255.9 ends, ICMP network sweep repeats...]
00:37:36.719357 207.253.109.179 > 10.0.255.1: icmp: echo request (ttl 244, id 16306)
[...]
00:37:49.994008 207.253.109.179 > 10.0.255.15: icmp: echo request (ttl 244, id 16322)
[...ICMP network sweep ends, UDP port 137 scan repeats...]
00:37:52.343759 207.253.109.179.1025 > 10.0.255.3.137: udp 50 (ttl 117, id 16323)
00:37:52.344387 10.0.255.130 > 207.253.109.179: icmp: host 10.0.255.3 unreachable - admin
prohibited filter (ttl 255, id 64909)
00:37:53.834906 207.253.109.179.1025 > 10.0.255.3.137: udp 50 (ttl 117, id 16324)
00:37:53.835604 10.0.255.130 > 207.253.109.179: icmp: host 10.0.255.3 unreachable - admin
prohibited filter (ttl 255, id 64911)
00:37:55.336943 207.253.109.179.1025 > 10.0.255.3.137: udp 50 (ttl 117, id 16325)
[...]
00:38:53.496511 207.253.109.179.1025 > 10.0.255.15.137: udp 50 (ttl 117, id 16372)
00:38:54.989870 207.253.109.179.1025 > 10.0.255.15.137: udp 50 (ttl 117, id 16373)
00:38:56.489946 207.253.109.179.1025 > 10.0.255.15.137: udp 50 (ttl 117, id 16374)
[...UDP port 137 scan ends]
```

1. Source of Trace:

Subnet of employer's Class-B address reserved for web servers. NOTE – All destination IP addresses have been sanitized. The protected addresses are presented as 10.0.255.1 through 15.

2. Detect was generated by:

TCPdump – `The date is 0901 (MMDD) timezone Eastern and the filter is -n -v ip and host 207.253.109.179`

Since this appears to be a multi-protocol port scan, the primary fields of interest in this case would be the source IP, the source port, the destination IP and the destination port. The time field must also be considered, as it is a good indicator of what method is being used to perform the attack (i.e. tool, script, etc.).

Below is a description of the logging fields for the TCPdump extract:

```
|      A      | |    B      | |C |   |     D    | |E| |F|
10:04:18.531749 192.168.81.31.1563 > 10.0.223.215.143: S
|          G           |    |   H   | |       I
2333129333:2333129333(0) win 32120 <mss 1460,sackOK,
            I                  |  | J|  |  K  |  |  L  |
timestamp 3739796 0,nop,wscale 0> (DF) (ttl 49, id 51509)

A – Time (HH:MM:SS.ssssss)
B – Source IP
C – Source Port
D - Destination IP
E – Destination Port
F – SYN Flag (May also be R=RESET, F=FIN or P=PUSH), indicates a TCP connection
G – Sequence Number (Relative Sequence Number)
H – The WIN (Window) Size
I – Packet Data Flags
J – DF or ‘Don’t Fragment’ Flag (May also be MF=More Fragments if packets are fragmented)
K - TTL or ‘Time To Live’ for the packet
L – PID or Packet ID Number
```

3. Probability the source address was spoofed:

The initiator must receive any responses for this port scan to be useful therefore spoofing is unlikely.

4. Description of Attack:

This scan is designed to identify active web servers within a specific IP range. ICMP echo, UDP 137 NetBIOS Name Service and various web service related TCP ports were probed. There are many well-known vulnerabilities associated with the scanned ports. These include:

UDP port 137 NetBIOS name service – CVE-1999-0288
TCP port 21 FTP – CVE-1999-0017, CVE-1999-0035, CVE-1999-0082, CVE-1999-0349
TCP port 23 TELNET – CVE-1999-0073, CVE-1999-0192, CVE-1999-0230, CVE-2000-0268
TCP port 53 DNS – CVE-1999-0010, CVE-1999-0048, CVE-1999-0274, CVE-1999-0275
TCP port 69 TFTP – CVE-2000-0015
TCP port 70 Gopher – CVE-1999-0124
TCP port 79 Finger – CVE-1999-0797
TCP port 80 HTTP – CVE-1999-0267, CVE-1999-0415, CVE-1999-0448, CVE-1999-0867
TCP port 110 POP3 – CVE-1999-0006, CVE-1999-0042, CVE-2000-0442
TCP port 143 IMAP – CVE-1999-0005, CVE-1999-0042, CVE-1999-0920, CVE-2000-0053
TCP port 6667 IRC client/server – CVE-2000-183

5. Attack Mechanism:

This attack is the result of an automated script and can be considered to be a fairly simple one. The packets generated are not crafted, as the source ports and sequence numbers increment in a normal manner. Furthermore the TCP port scan is a simple SYN scan. There is no attempt to hide the scan and no anomalous flags are set in an attempt to evade detection or to provoke responses that may provide information about the targeted host’s operating system or the version of the service running on the targeted port.

The initial ICMP echo requests are designed to accomplish two things; determine if the target IP has an associated active host and gauge the likelihood that perimeter defenses are in use (i.e. – router ACLs, presence of firewall, etc.) Additionally, the ICMP echo replies that may be received could also provide some added value – the TTL values of the echo replies may offer some insight into which operating system is in use on the targeted host.

The subsequent port scanning is designed to determine the purpose of the server's existence, or its "raison d'être." By probing UDP 137, the remote attacker can determine if the target host is a Windows-based file server. The probing of the TCP ports is designed to determine what services are being offered. All the ports that were scanned are associated with common web services (i.e. – FTP, TELNET, HTTP, IRC, etc.). Any system that replies with a SYN/ACK can be assumed to be active and running the one of these services. This reconnaissance information could be used to help facilitate a future attack.

Once the information from this port scan is analyzed, the attacker can gauge his list of targets for "easy kills" and proceed to attempt exploiting the vulnerabilities associated with any of the services offered on the active servers. By using one of the well-known vulnerabilities, the attacker could gain privileged user access or cause a 'denial of service' against the targeted service. This can occur immediately after the reconnaissance effort ends, or whenever the attacker who performed the mapping feels is appropriate. The attacking of the individual services (TELNET, IMAP) would probably require the use of additional tools or scripts.

6. Correlation:

A Cisco SecureIDS sensor detected this scan. TCPdump was used to replay raw SHADOW logs from a SHADOW sensor co-located with the Cisco SecureIDS to determine the extent of the scan. This action was necessary as the Cisco SecureIDS had noted only three of the TCP ports targeted, while in reality an ICMP echo request, UDP port 137 and fifteen different TCP ports were actually involved.

This particular scan pattern has apparently not been noted before, however there are many well-known vulnerabilities associated with UDP port 137 and the various TCP ports that were scanned.

7. Evidence of active targeting:

Since this scan involved a subnet used exclusively for connecting web servers to the Internet, it would appear that this port scan was quite deliberate. All hosts within the 10.0.255.1 through 15 range were first scanned using an ICMP echo and then were scanned for both UDP port 137 and many well-known TCP ports.

8. Severity:

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

(4+5) – (5+4) = 0

- This subnet contains a several web servers and a mail server (4);
- Attack is a network scan that will provide focused info about each scanned host by using ICMP echo requests to identify active hosts, UDP port 137 to identify possible Windows-based hosts and attempts to locate several well-known web service related TCP ports available on each IP address scanned (5);
- Network is protected by an access control list (ACL) on the border router and a Cisco SecureIDS sensor (5);
- These systems on the targeted subnet are predominately Windows NT 4.0 based, and those that act as web servers run IIS. All of these hosts are reasonably well maintained and are patched with a recent service pack (4).

9. Defensive recommendation:

Defenses are fine, as the ICMP portion of the attack was blocked using an “Admin Prohibited Filter” within the ACL on the border router and a Cisco SecureIDS sensor detected the UDP and TCP portion of the attack. The border router is also configured to allow only limited access to some of the addresses in the 10.0.255.0/24 subnet. “Destination Unreachable” messages were sent in reply to the scans against many of the hosts.

10. Multiple choice test question:

Given the following network trace, identify the correct answer:

```
00:35:08.446191 207.253.109.179.3805 > 10.0.255.1.80: S
501156559:501156559(0) win 8760 <mss 1460,nop,nop,sackOK> (DF) (ttl
117, id 16167)
00:35:08.447100 10.0.255.1.80 > 207.253.109.179.3805: R 0:0(0) ack
501156560 win 0 (ttl 255, id 64607)
```

a) IP 207.253.109.179 is surfing web pages on IP 10.0.255.1
b) The web server on IP 10.0.255.1 has crashed
c) There is no web server active at IP 10.0.255.1
d) None of the above

Answer: c – IP 10.0.255.1 has sent its reset as a direct result of the fact that is not offering any service on TCP port 80

Detect 3

```
10:19:40.928274 139.175.158.50.80 > 10.0.58.255.40474: S 3075756002:3075756002(0) ack
2652504065 win 16384 <mss 1460> (DF) (ttl 46, id 4190)
10:19:40.938040 139.175.158.50.80 > 10.0.58.255.40474: R 1:1(0) ack 1 win 16384 (DF) (ttl
46, id 4517)
10:19:42.609751 139.175.158.50.80 > 10.0.142.180.19208: S 3934654527:3934654527(0) ack
1258815489 win 16384 <mss 1460> (DF) (ttl 46, id 40487)
10:19:42.619819 139.175.158.50.80 > 10.0.142.180.19208: R 1:1(0) ack 1 win 16384 (DF)
(ttl 46, id 40805)
10:19:51.327206 139.175.158.50.80 > 10.0.220.50.51468: S 3874461458:3874461458(0) ack
3373006849 win 16384 <mss 1460> (DF) (ttl 46, id 22872)
10:19:51.329268 139.175.158.50.80 > 10.0.220.50.51468: R 1:1(0) ack 1 win 16384 (DF) (ttl
46, id 22939)
10:19:54.945638 139.175.158.50.80 > 10.0.248.190.32777: S 1285692721:1285692721(0) ack
2148073473 win 16384 <mss 1460> (DF) (ttl 46, id 29400)
10:19:54.945709 139.175.158.50.80 > 10.0.248.190.32777: R 1:1(0) ack 1 win 16384 (DF)
(ttl 46, id 29439)
10:19:55.492313 139.175.158.50.80 > 10.0.241.124.9478: S 1504980846:1504980846(0) ack
621150209 win 16384 <mss 1460> (DF) (ttl 46, id 38641)
10:19:55.500725 139.175.158.50.80 > 10.0.241.124.9478: R 1:1(0) ack 1 win 16384 (DF) (ttl
46, id 38936)
[...]
09:04:26.297365 139.175.158.50.80 > 10.0.36.31.54272: S 2917146530:2917146530(0) ack
3556769793 win 16384 <mss 1460> (DF) (ttl 45, id 54596)
09:04:26.300590 139.175.158.50.80 > 10.0.36.31.54272: R 1:1(0) ack 1 win 16384 (DF) (ttl
45, id 54737)
09:04:46.514920 139.175.158.50.80 > 10.0.164.246.59919: S 3270474145:3270474145(0) ack
3926851585 win 16384 <mss 1460> (DF) (ttl 45, id 38082)
09:04:46.557661 139.175.158.50.80 > 10.0.164.246.59919: R 1:1(0) ack 1 win 16384 (DF)
(ttl 45, id 38324)
09:04:47.457568 139.175.158.50.80 > 10.0.18.37.15134: S 3690686767:3690686767(0) ack
991821825 win 16384 <mss 1460> (DF) (ttl 45, id 55809)
09:04:47.464690 139.175.158.50.80 > 10.0.18.37.15134: R 1:1(0) ack 1 win 16384 (DF) (ttl
45, id 55994)
09:04:57.031064 139.175.158.50.80 > 10.0.183.228.39430: S 3340091200:3340091200(0) ack
2584084481 win 16384 <mss 1460> (DF) (ttl 45, id 25246)
09:04:57.031445 139.175.158.50.80 > 10.0.183.228.39430: R 1:1(0) ack 1 win 16384 (DF)
(ttl 45, id 25289)
```

```
09:05:00.066022 139.175.158.50.80 > 10.0.149.193.63507: S 386669101:386669101(0) ack
4161994753 win 16384 <mss 1460> (DF) (ttl 45, id 16193)
09:05:00.077711 139.175.158.50.80 > 10.0.149.193.63507: R 1:1(0) ack 1 win 16384 (DF)
(ttl 45, id 16650)
```

1. Source of Trace:

Employer’s Class-B network. NOTE – All destination IP addresses have been sanitized. The protected addresses are presented as 10.0.0.0/24.

2. Detect was generated by:

TCPdump – This event was analyzed over three (3) consecutive days:

1) `The date is 0917 (MMDD) timezone Eastern and the filter is`
`-n -v ip and host 139.175.158.50`

2) `The date is 0917 (MMDD) timezone Eastern and the filter is`
`-n -v ip and host 139.175.158.50`

3) `The date is 0918 (MMDD) timezone Eastern and the filter is`
`-n -v ip and host 139.175.158.50`

Since this activity appears to be the result of an attacker using spoofed source IP addresses that match IP address assigned to the protected network, the primary fields of interest in this case would be the source IP, the source port, the destination IP and the destination port. The time field must also be considered, as it is a good indicator of what method is being used to perform the attack (i.e. tool, script, etc.).

Below is a description of the logging fields for the TCPdump extract:

```
|      A      | |    B      | |C |   |     D    | |E| |F|
10:04:18.531749 192.168.81.31.1563 > 10.0.223.215.143: S
|          G           |    |   H   | |       I
2333129333:2333129333(0) win 32120 <mss 1460,sackOK,
            I                  |  | J|  |  K  |  |  L  |
timestamp 3739796 0,nop,wscale 0> (DF) (ttl 49, id 51509)

A – Time (HH:MM:SS.ssssss)
B – Source IP
C – Source Port
D - Destination IP
E – Destination Port
F – SYN Flag (May also be R=RESET, F=FIN or P=PUSH), indicates a TCP connection
G – Sequence Number (Relative Sequence Number)
H – The WIN (Window) Size
I – Packet Data Flags
J – DF or ‘Don’t Fragment’ Flag (May also be MF=More Fragments if packets are fragmented)
K – TTL or ‘Time To Live’ for the packet
L – PID or Packet ID Number
```

3. Probability the source address was spoofed:

Because of the nature of the traffic detected, this particular issue must be answered in two parts.

1) The original TCP SYN Flood that likely triggered this event does have spoofed source IP addresses involved.
2) The traffic being received on the protected network is in response to the original spoofed packets. The received traffic is from the target of the TCP SYN Flood and is therefore not spoofed.

4. Description of attack:

In this case, the protected network is not really under attack. The traffic received is the result of IP 139.175.158.50.80 being TCP SYN Flooded. Because there is a scanning technique that uses SYN/ACK packets to perform network mapping, the traffic must be confirmed. It becomes obvious that this isn't some sort of scan because the server sending these packets is also sending RST/ACK packets. These RST/ACK packets indicate that that IP 139.175.50.80 is trying to timeout the uncompleted connections before suffering a Denial of Service (DoS).

TCP SYN Flooding is an old attack and is listed in the CVE database. CVE-1999-0116 refers.

5. Attack mechanism:

As described in CERT Advisory CA-96.21, (available at the following URL: http://www.cert.org/advisories/CA-96.21.tcp_syn_flooding.html), the TCP SYN Flood is a DoS attack. It is intended to cause either a performance loss or complete failure of the target by sending it SYN packets from spoofed source IP addresses. When the target receives these SYN packets, it responds with the standard SYN/ACK. This creates a "half-open connection," and the target will track these impending connections. Because there is usually no active hosts at the spoofed source IP addresses, the queue will continue to fill without ever having any of these connections established. Once the resource limit of this queue is exceeded, the server's operating system will usually suffer a fatal crash.

For this attack to be especially effective, the attacker will scan various network segments to try and find ranges of IP addresses that appear to be inactive before the attack is launched. By using spoofed source IP addresses from within these ranges, the attacker will help ensure that no responses with be sent to the target in response to the SYN/ACK packets the target will generate.

Newer TCP/IP stack implementations have attempted to prevent this from occurring by building a timeout mechanism into the connection queue. If the completing ACK is not received in reply to the SYN/ACK, the half-open connection will timeout and a TCP Reset will be sent (which is clearly demonstrated in the network trace for this detect!). Unfortunately, a performance loss is still likely if the volume of spoofed SYN requests is high because of the resources required to track the incoming connection requests and send the TCP resets when these bogus connections timeout.

6. Correlation:

A Cisco SecureIDS deployed on the protected network initially noted this activity. The shear volume of traffic and the number of alarms warranted a more detailed inspection of the activity. A co-located SHADOW sensor was consulted to confirm the activity and TCPdump was run against the raw logs to verify that no anomalous activity or system replies had occurred.

Below is a ConSeal PC Firewall log containing a trace that is similar to the activity detected on the Cisco SecureIDS sensor employed on the 10.0.0.0/16 network.

```
2000/09/18 05:25:37 AM GMT -0400: AcerLAN ALN-325 1..[0000][No matching rule] Blocking
incoming TCP: src=139.175.158.50, dst=192.168.163.101, sport=80, dport=49695.
2000/09/18 05:25:37 AM GMT -0400: AcerLAN ALN-325 1..[0000][No matching rule] Blocking
incoming TCP: src=139.175.158.50, dst=192.168.163.101, sport=80, dport=49695.
```

This ConSeal PC Firewall log is from my own personal computer. I have a cable modem, and because of the permanent nature of this Internet connection, I protect my system from prying eyes with this very capable host-based firewall.

This log clearly demonstrates that the TCP SYN Flood against IP 139.175.158.50 involves a large number of spoofed source IP addresses, all of which were established by the attacker before the attack began as being likely inactive. Compare the ConSeal trace with another excerpt from the TCPdump logs below and it becomes clear that this is a large-scale DoS attack.

```
05:25:35.916545 139.175.158.50.80 > 10.0.235.42.23059: S 293023799:293023799(0) ack
1511194625 win 16384 <mss 1460> (DF) (ttl 45, id 36075)
05:25:35.922851 139.175.158.50.80 > 10.0.235.42.23059: R 1:1(0) ack 1 win 16384 (DF) (ttl
45, id 36375)
05:25:40.291078 139.175.158.50.80 > 10.0.37.5.44314: S 2238743800:2238743800(0) ack
2904162305 win 16384 <mss 1460> (DF) (ttl 45, id 52318)
05:25:40.295233 139.175.158.50.80 > 10.0.37.5.44314: R 1:1(0) ack 1 win 16384 (DF) (ttl
45, id 52529)
```

This log excerpt clearly demonstrates the fact that the activity noted on the ConSeal PC Firewall occurred concurrently with the activity noted by TCPdump. Since these two logs come from two entirely different networks that have no direct relation to one another, it is safe to assume that there will be other spoofed IP addresses being used in this attack as well.

7. Evidence of active targeting:

Because the source IP addresses in use for the attack are spoofed, IP 139.175.158.50 is definitely being deliberately targeted. However, since IP 139.175.158.50 is not associated with the protected network, the traffic it is sending to the addresses within the 10.0.0.0/24 range themselves can be considered “wrong numbers.”

8. Severity:

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

(4+5) – (4+5) = 0

- The attack involves random hosts on a Class-B network (2);
- The traffic is the result of someone performing an attack using spoofed source IP addresses belonging to the same Class-B network and is not the result of active targeting (2);
- The IP addresses involved are quite often unused, so no system is present to be affected (4);
- Network is protected by an access control list (ACL) on the border router, a Cisco SecureIDS sensor and a SHADOW sensor (5).

9. Defensive recommendations:

None – preventing the use of IP addresses within a given address ranges for using in spoofing attacks is quite difficult. This attack is also not directed at the 10.0.0.0/24 network itself but rather IP 139.175.158.50, so is cause for little concern for this site.

10. Multiple choice test question:

Given the following network trace, identify the correct answer:

```
09:05:00.066022 139.175.158.50.80 > 10.0.149.193.63507: S
386669101:386669101(0) ack 4161994753 win 16384 <mss 1460> (DF) (ttl
45, id 16193)
09:05:00.077711 139.175.158.50.80 > 10.0.149.193.63507: R 1:1(0) ack 1
win 16384 (DF) (ttl 45, id 16650)
```

NOTE: There are thousands of similar TCPdump records showing the same source IP address sending SYN/ACK and RST/ACK packets to random hosts within the 10.0.0.0/24 network range.

a) People from the 10.0.0.0/24 network are surfing web pages on IP 139.175.158.50 and it is busy
b) The web server on IP 139.175.158.50 has crashed
c) There is no web server active at IP 139.175.158.50
d) Someone is using spoofed source IP addresses belonging to the 10.0.0.0/24 network to target IP 139.175.158.50

Answer: d – This trace is the result of someone SYN Flooding IP 139.175.158.50 using spoofed source IP addresses belonging to the 10.0.0.0/24 address range

Detect 4

```
2000/08/28 03:53:07 AM GMT -0400: AcerLAN ALN-325 1..[0000][No matching rule] Blocking
incoming TCP: src=211.177.87.231, dst=192.168.163.101, sport=1889, dport=12345.
2000/08/30 10:46:18 AM GMT -0400: AcerLAN ALN-325 1..[0000][No matching rule] Blocking
incoming TCP: src=211.58.162.155, dst=192.168.163.101, sport=4419, dport=12345.
2000/08/30 10:46:21 AM GMT -0400: AcerLAN ALN-325 1..[0000][No matching rule] Blocking
incoming TCP: src=211.58.162.155, dst=192.168.163.101, sport=4419, dport=12345.
2000/08/31 04:49:37 AM GMT -0400: AcerLAN ALN-325 1..[0000][No matching rule] Blocking
incoming TCP: src=211.58.18.220, dst=192.168.163.101, sport=2279, dport=12345.
2000/08/31 04:49:40 AM GMT -0400: AcerLAN ALN-325 1..[0000][No matching rule] Blocking
incoming TCP: src=211.58.18.220, dst=192.168.163.101, sport=2279, dport=12345.
2000/08/31 12:50:49 PM GMT -0400: AcerLAN ALN-325 1..[0000][No matching rule] Blocking
incoming TCP: src=211.177.131.151, dst=192.168.163.101, sport=1854, dport=12345.
2000/08/31 12:50:52 PM GMT -0400: AcerLAN ALN-325 1..[0000][No matching rule] Blocking
incoming TCP: src=211.177.131.151, dst=192.168.163.101, sport=1854, dport=12345.
2000/09/01 03:48:29 AM GMT -0400: AcerLAN ALN-325 1..[0000][No matching rule] Blocking
incoming TCP: src=210.93.10.104, dst=192.168.163.101, sport=2376, dport=12345.
2000/09/01 03:48:32 AM GMT -0400: AcerLAN ALN-325 1..[0000][No matching rule] Blocking
incoming TCP: src=210.93.10.104, dst=192.168.163.101, sport=2376, dport=12345.
2000/09/02 12:29:54 AM GMT -0400: AcerLAN ALN-325 1..[0000][No matching rule] Blocking
incoming TCP: src=211.108.53.3, dst=192.168.163.101, sport=1137, dport=12345.
2000/09/04 06:54:31 AM GMT -0400: AcerLAN ALN-325 1..[0000][No matching rule] Blocking
incoming TCP: src=211.176.104.14, dst=192.168.163.101, sport=2352, dport=12345.
2000/09/04 06:54:34 AM GMT -0400: AcerLAN ALN-325 1..[0000][No matching rule] Blocking
incoming TCP: src=211.176.104.14, dst=192.168.163.101, sport=2352, dport=12345.
2000/09/04 08:25:17 PM GMT -0400: AcerLAN ALN-325 1..[0000][No matching rule] Blocking
incoming TCP: src=211.108.156.192, dst=192.168.163.101, sport=1178, dport=12345.
2000/09/04 09:10:20 PM GMT -0400: AcerLAN ALN-325 1..[0000][No matching rule] Blocking
incoming TCP: src=210.99.222.14, dst=192.168.163.101, sport=2363, dport=12345.
2000/09/04 09:10:23 PM GMT -0400: AcerLAN ALN-325 1..[0000][No matching rule] Blocking
incoming TCP: src=210.99.222.14, dst=192.168.163.101, sport=2363, dport=12345.
2000/09/06 08:40:36 PM GMT -0400: AcerLAN ALN-325 1..[0000][No matching rule] Blocking
incoming TCP: src=210.104.64.75, dst=192.168.163.101, sport=2314, dport=12345.
2000/09/07 02:57:27 AM GMT -0400: AcerLAN ALN-325 1..[0000][No matching rule] Blocking
incoming TCP: src=211.178.102.101, dst=192.168.163.101, sport=1124, dport=12345.
2000/09/07 02:57:30 AM GMT -0400: AcerLAN ALN-325 1..[0000][No matching rule] Blocking
incoming TCP: src=211.178.102.101, dst=192.168.163.101, sport=1124, dport=12345.
2000/09/07 07:22:58 AM GMT -0400: AcerLAN ALN-325 1..[0000][No matching rule] Blocking
incoming TCP: src=211.178.208.214, dst=192.168.163.101, sport=2853, dport=12345.
2000/09/07 07:23:01 AM GMT -0400: AcerLAN ALN-325 1..[0000][No matching rule] Blocking
incoming TCP: src=211.178.208.214, dst=192.168.163.101, sport=2853, dport=12345.
2000/09/08 09:02:09 AM GMT -0400: AcerLAN ALN-325 1..[0000][No matching rule] Blocking
incoming TCP: src=211.44.243.189, dst=192.168.163.101, sport=3909, dport=12345.
2000/09/08 09:02:12 AM GMT -0400: AcerLAN ALN-325 1..[0000][No matching rule] Blocking
incoming TCP: src=211.44.243.189, dst=192.168.163.101, sport=3909, dport=12345
```

1. Source of Trace:

Personal @Home cable modem connection. NOTE – All destination IP addresses have been sanitized. The protected address is presented as 192.168.163.101.

2. Detect was generated by:

ConSeal PC Firewall

Since this activity is the result of scans for the Netbus Trojan, the primary fields of interest in this case would be the source IP, the source port, the destination IP and the destination port. The time field must also be considered, as it is a good indicator of what method is being used to perform the attack (i.e. tool, script, etc.).

Below is a description of the logging fields for the ConSeal PC Firewall extract:

```
|    A   | |    B    | |   C   |  |        D        | | E|
2000/09/18 05:25:37 AM GMT -0400: AcerLAN ALN-325 1..[0000]
 |       F   | |             G         |  |        H       |
[No matching rule] Blocking incoming TCP: src=139.175.158.50,
|        I        |  |   J  |  |    K    |
dst=192.168.163.101, sport=80, dport=49695.

A – Date (YYYY/MM/DD)
B – Time (HH:MM:SS AM/PM)
C – Offset from GMT (+/- HHMM)
D – Network Device Name (Used to differentiate between network adapters (i.e. – NIC,
Dial-Up Networking) in use on the protected host
E – Associated Ruleset (#### - 0000 represents the first network device, 0001 represents
the second network device, etc.)
F – Triggering Rule ("No matching rule" represents the default 'deny all' rule enabled
when firewall is not in 'Learning Mode'
G – Action taken (i.e. – Blocking, Connection Attempt, etc.)
H – Source IP Address
I – Destination IP Address
J – Source Port
K – Destination Port
```

3. Probability that the source address was spoofed:

The initiator must receive any responses for this port scan to be successful therefore spoofing is unlikely.

4. Description of Attack:

This scan is designed to identify hosts compromised by the Netbus Trojan server application that resides on TCP port 12345.

5. Attack Mechanism:

This attack is a simple TCP SYN scan looking for available Netbus servers. A simple script can be easily written or scanning tools like NMAP can be easily configured to scan whole address ranges exclusively for systems with TCP port 12345 open.

Should a system be found reachable on port 12345, the attacker will assume that the Netbus Trojan is present on the host and they will probably attempt to connect using the Netbus client.

The Netbus client software is a simple Graphic User Interface (GUI) that is designed to simplify the control of the compromised host. Peripherals such as the mouse or CD-ROM can be controlled, screenshots or sound files recorded, and files can be viewed, modified or deleted by the attacker on the victim's host by using a few simple mouse clicks.

The server must be installed and active on the intended victim's server for remote access through this Trojan to be successful. The server portion of Netbus is often delivered to its intended victims via e-mail, ICQ or through other means. A social engineering attack is usually used to trick the intended victim to run the executable code to install the Netbus server. One popular method used when Netbus first arrived on the scene was to provide the intended targets with a game called "Whack-a-mole" and encourage them to play it. The game really worked, but was packaged along with the Netbus server in a self-extracting archive. As the game is executed, the server is enabled and opens whatever TCP port the server was configured to bind to.

Because of the age of this attack, the multiple defenses one can employ to protect against it (i.e. – AV software, firewalls) and the many different analyses that have been given, more detailed information on how Netbus works can be obtained at the following URLs:

NAI AVERT Virus Library – http://vil.nai.com/villib/dispVirus.asp?virus_k=98025
F-Secure Computer Virus Information pages – http://www.datafellows.com/v-descs/netbus.htm

6. Correlation:

Multiple sites have been reporting similar activity. The last two editions of NIPC's "Cybernotes" have discussed the increased volume of this particular scan. It has been noted that the scanning from the Netbus Trojan, and well-known Trojans in general, have been increasing steadily.

Excerpt from the NIPC's Cybernotes Issue #2000-18, found under the "Probes/Scans" portion of the "Trends" section – "*Increasing reports of scans to known Trojan ports. System administrators should consult their intrusion detection system and firewall logs for unusual port scans.*"

7. Evidence of active targeting:

Since this scan is intended to located Netbus Trojan compromised machines throughout the entire Internet, it would appear that this activity is quite deliberate. Hosts from all over the world have been scanned and there are many sites confirming these scans.

8. Severity:

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

(5+3) – (5+1) = 2

- This is my computer! Additionally, the scanning is occurring often and against many network other than the one I'm connected to (5);
- Attack is a network scan for the Netbus Trojan (3);
- A firewall and up-to-date AV software protect my computer (5);
- The @Home network is well known for its security short comings… (1).

9. Defensive recommendation:

Defenses are fine, especially when one considers that this Trojan has been detectable by all major distributions of AV software for quite sometime. Because a firewall is also employed, the system (mine) would be protected against exploitation even if the AV software was malfunctioning or uninstalled. No connection would be possible without the firewall also being disabled or uninstalled.

10. Multiple choice test question:

The Netbus Trojan can be delivered to a potential target via:

a) E-mail
b) Napster/Gnutella
c) ICQ
d) All of the above

Answer: d – There have already been reports of people receiving Trojan-infected files through "peer-to-peer file sharing" applications like Napster!

Assignment 2 – Evaluate an Attack

There are many useful applications and tools that have been released in an attempt to gather as many useful network and Internet utilities together in one place as possible. "Genius" is one such software package. For the purposes of this assignment, we'll take a look at version 2.7 of the handy application. The latest version is 3.0 and is available as a time-limited shareware download at http://www.indiesoft.com.

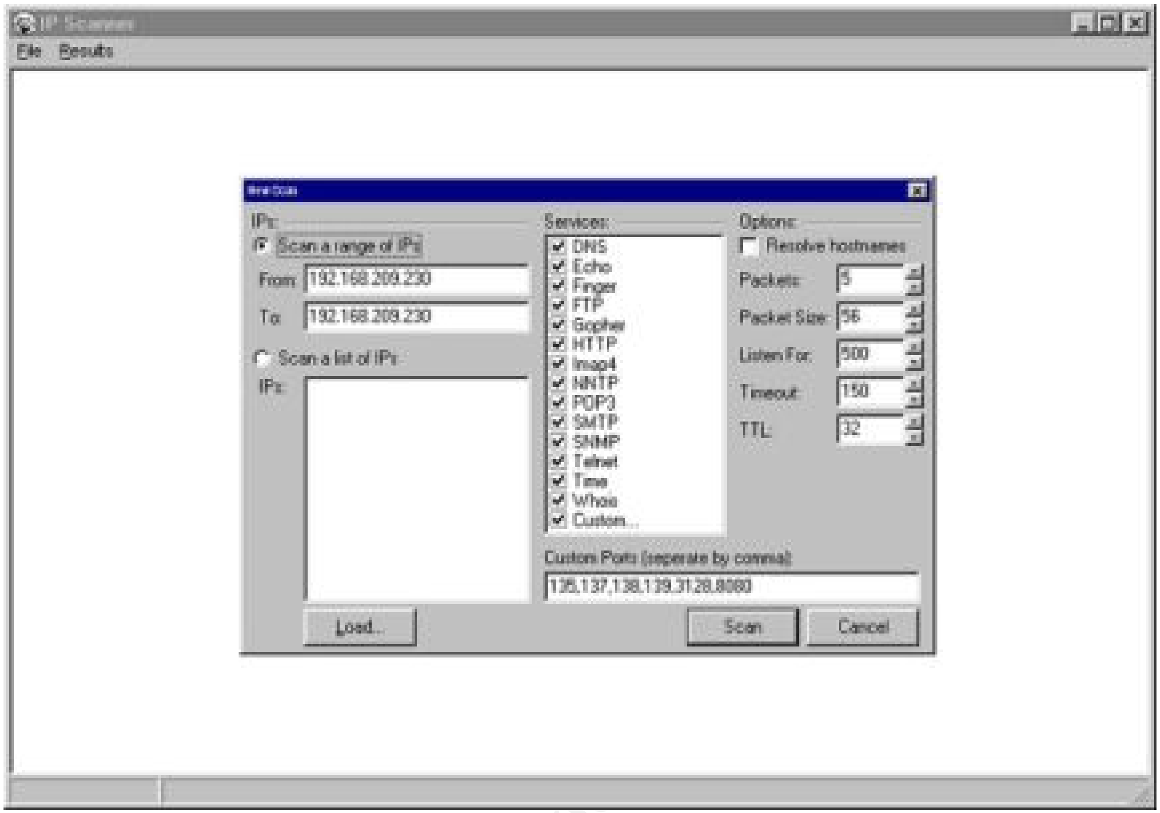
"Genius" brings together many useful online and offline utilities. Many of the online utilities are ones not normally provided in Windows 32 operating environment. Regular users of UNIX/Linux who must use a MS Windows 9x or Windows NT computer will find many of these tools indispensable.

"Genius" has the following online utilities bundled:

- Finger client
- FTP client (normal or passive mode capable)
- HTTP client (text-based)
- Ping client (ICMP)
- SMTP client
- TELNET client
- Time client
- Traceroute client
- Whois client
- Current Connections tool
- IP Scanner
- Nslookup tool
- POP3 Cleaner
- Port lookup tool
- Portscan detection routine

Most of these tools and clients are designed to aid a user in using various popular Internet protocols quickly and efficiently from one spot. The remainder are designed to help a user determine the level of activity their host has on the connected network.

The IP scanner included with "Genius" allows a user to perform a port scan that is somewhat configurable. While the speed of the scan or the protocols used is not adjustable, the target TCP ports are selectable through a simple Graphic User Interface (GUI). As Figure 1 below illustrates, only a few inputs and a few mouse clicks are required from the user to get this scanner running.



IPs:

● Scan a range of IPs

From: 192.168.209.230

To: 192.168.209.230

○ Scan a list of IPs

IPs:

Services:

- ☑ DNS
- ☑ Echo
- ☑ Finger
- ☑ FTP
- ☑ Gopher
- ☑ HTTP
- ☑ Imap4
- ☑ NNTP
- ☑ POP3
- ☑ SMTP
- ☑ SNMP
- ☑ Telnet
- ☑ Time
- ☑ Whois
- ☑ Custom...

Custom Ports (seperate by comma):
135,137,138,139,3128,8080

Options:

□ Resolve hostnames

Packets: 5

Packet Size: 56

Listen For: 500

Timeout: 150

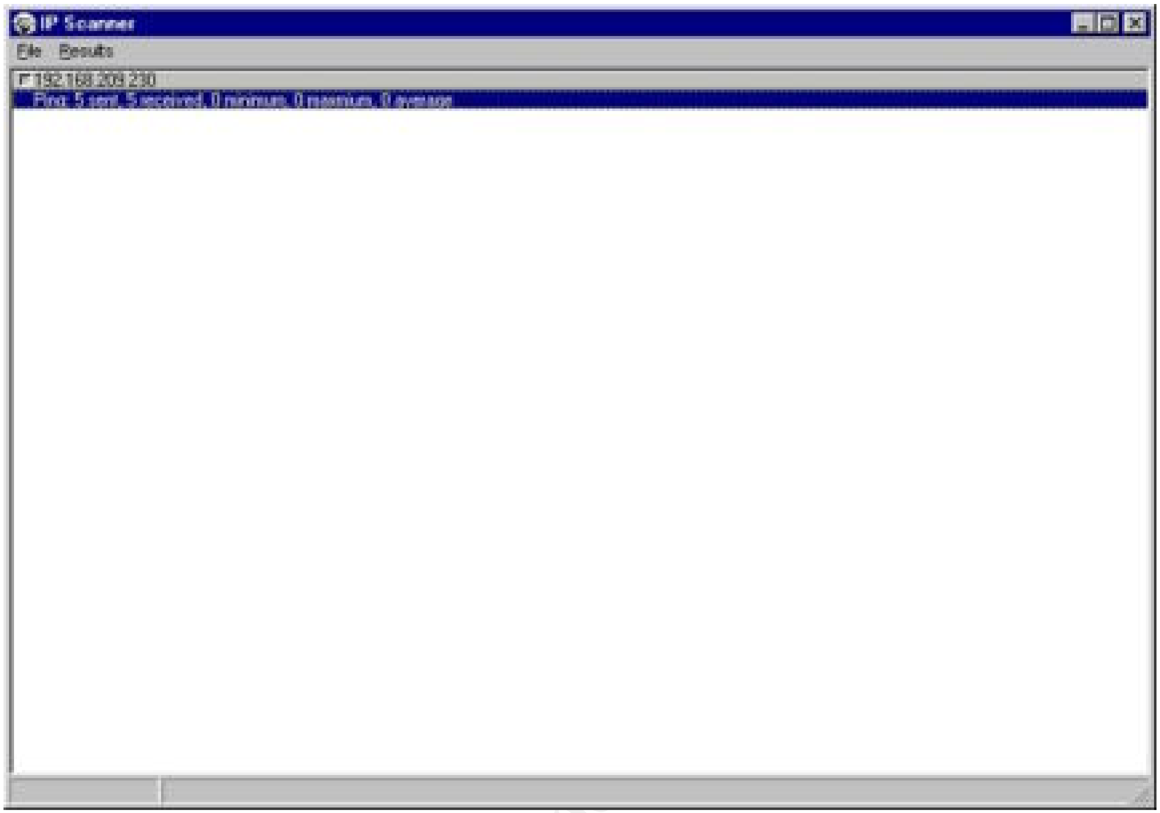
TTL: 32

Figure 1 – IP Scanner configuration GUI

The following information is provided in the Help File:

- Scan a range of IPs: If checked, IP Scanner will scan all the IPs from From IP to To IP, inclusive.
- Scan a list of IPs: If checked, IP Scanner will scan all the IPs in the IP list.
- Services: IP Scanner will see if any of the checked services are running on each IP.
- Custom Ports: If "Custom" is checked in Services, enter the ports to be checked here.
- Resolve Hostnames: If checked, IP Scanner will convert all numerical IPs (i.e., 1.2.3.4) into hostnames (i.e., www.server.com).
- Packets: The number of packets to ping each IP with.
- Packet Size: The size of the packets to ping each IP with.
- Listen For: How long to listen for a reply on each port.
- Timeout: The number of milliseconds Genius will try to connect to each port for. After that time has expired, Genius will consider that port to be closed.
- TTL: Maximum number of hops each ICMP packet can take.

Once the scan is started, the program sends the set number of ICMP echo request to determine if a host is active for each IP address in the given range. If no ICMP echo reply is received, the application ceases its efforts and moves onto the next IP. If a host replies, it is displayed in the IP Scanner window. For each active host, the statistics for the ICMP echoes is displayed, followed by information (including banners) that is made available for each TCP port that is found open. An example is shown in Figure 2 below.



```
192.168.209.230
  Ping: 5 sent, 5 received, 0 minimum, 0 maximum, 0 average
```

Figure 2 – The IP Scanner results window

As with any port scanner, the probing that results from its use can often be viewed in a negative light. This is especially true if this tool is used against Internet IP addresses that don't belong to the same network as the user's system. By using TCPdump, we can look at the scan pattern to see if it has a unique signature. In this case, the Windows NT system running the "Genius" software scans another host that is known to be running Windows NT 4.0 as well.

```
<Default Scan begins...>
[ICMP echoes begin...]
07:52:06.896731 192.168.209.243 > 192.168.209.245: icmp: echo request
07:52:06.896731 192.168.209.245 > 192.168.209.243: icmp: echo reply
07:52:06.896731 192.168.209.243 > 192.168.209.245: icmp: echo request
07:52:06.896731 192.168.209.245 > 192.168.209.243: icmp: echo reply
07:52:06.896731 192.168.209.243 > 192.168.209.245: icmp: echo request
07:52:06.896731 192.168.209.245 > 192.168.209.243: icmp: echo reply
07:52:06.896731 192.168.209.243 > 192.168.209.245: icmp: echo request
07:52:06.906745 192.168.209.245 > 192.168.209.243: icmp: echo reply
07:52:06.906745 192.168.209.243 > 192.168.209.245: icmp: echo request
07:52:06.906745 192.168.209.245 > 192.168.209.243: icmp: echo reply
[ICMP echoes end, TCP SYN scan begins...]
[TCP 53 DNS scans (x4)]
07:52:06.916760 192.168.209.243.1362 > 192.168.209.245.53: S 327140:327140(0) win 8192
<mss 1460> (DF)
07:52:06.916760 192.168.209.245.53 > 192.168.209.243.1362: R 0:0(0) ack 327141 win 0
07:52:07.367408 192.168.209.243.1362 > 192.168.209.245.53: S 327140:327140(0) win 8192
<mss 1460> (DF)
07:52:07.367408 192.168.209.245.53 > 192.168.209.243.1362: R 0:0(0) ack 1 win 0
07:52:07.868128 192.168.209.243.1362 > 192.168.209.245.53: S 327140:327140(0) win 8192
<mss 1460> (DF)
07:52:07.868128 192.168.209.245.53 > 192.168.209.243.1362: R 0:0(0) ack 1 win 0
07:52:08.368848 192.168.209.243.1362 > 192.168.209.245.53: S 327140:327140(0) win 8192
<mss 1460> (DF)
07:52:08.368848 192.168.209.245.53 > 192.168.209.243.1362: R 0:0(0) ack 1 win 0
[TCP 7 Echo scans (x4)]
```

```
07:52:08.378862 192.168.209.243.1363 > 192.168.209.245.7: S 327147:327147(0) win 8192
<mss 1460> (DF)
07:52:08.378862 192.168.209.245.7 > 192.168.209.243.1363: R 0:0(0) ack 327148 win 0
07:52:08.869568 192.168.209.243.1363 > 192.168.209.245.7: S 327147:327147(0) win 8192
<mss 1460> (DF)
07:52:08.869568 192.168.209.245.7 > 192.168.209.243.1363: R 0:0(0) ack 1 win 0
07:52:09.370288 192.168.209.243.1363 > 192.168.209.245.7: S 327147:327147(0) win 8192
<mss 1460> (DF)
07:52:09.370288 192.168.209.245.7 > 192.168.209.243.1363: R 0:0(0) ack 1 win 0
07:52:09.871008 192.168.209.243.1363 > 192.168.209.245.7: S 327147:327147(0) win 8192
<mss 1460> (DF)
07:52:09.871008 192.168.209.245.7 > 192.168.209.243.1363: R 0:0(0) ack 1 win 0
[TCP 79 Finger scans (x4)]
07:52:09.881022 192.168.209.243.1364 > 192.168.209.245.79: S 327156:327156(0) win 8192
<mss 1460> (DF)
07:52:09.881022 192.168.209.245.79 > 192.168.209.243.1364: R 0:0(0) ack 327157 win 0
07:52:10.371728 192.168.209.243.1364 > 192.168.209.245.79: S 327156:327156(0) win 8192
<mss 1460> (DF)
07:52:10.371728 192.168.209.245.79 > 192.168.209.243.1364: R 0:0(0) ack 1 win 0
07:52:10.872448 192.168.209.243.1364 > 192.168.209.245.79: S 327156:327156(0) win 8192
<mss 1460> (DF)
07:52:10.872448 192.168.209.245.79 > 192.168.209.243.1364: R 0:0(0) ack 1 win 0
07:52:11.373168 192.168.209.243.1364 > 192.168.209.245.79: S 327156:327156(0) win 8192
<mss 1460> (DF)
07:52:11.373168 192.168.209.245.79 > 192.168.209.243.1364: R 0:0(0) ack 1 win 0
[TCP 21 FTP scans (x4)]
07:52:11.383182 192.168.209.243.1365 > 192.168.209.245.21: S 327160:327160(0) win 8192
<mss 1460> (DF)
07:52:11.383182 192.168.209.245.21 > 192.168.209.243.1365: R 0:0(0) ack 327161 win 0
07:52:11.873888 192.168.209.243.1365 > 192.168.209.245.21: S 327160:327160(0) win 8192
<mss 1460> (DF)
07:52:11.873888 192.168.209.245.21 > 192.168.209.243.1365: R 0:0(0) ack 1 win 0
07:52:12.374608 192.168.209.243.1365 > 192.168.209.245.21: S 327160:327160(0) win 8192
<mss 1460> (DF)
07:52:12.374608 192.168.209.245.21 > 192.168.209.243.1365: R 0:0(0) ack 1 win 0
07:52:12.875328 192.168.209.243.1365 > 192.168.209.245.21: S 327160:327160(0) win 8192
<mss 1460> (DF)
07:52:12.875328 192.168.209.245.21 > 192.168.209.243.1365: R 0:0(0) ack 1 win 0
[TCP 70 Gopher scans (x4)]
07:52:12.885342 192.168.209.243.1366 > 192.168.209.245.70: S 327166:327166(0) win 8192
<mss 1460> (DF)
07:52:12.885342 192.168.209.245.70 > 192.168.209.243.1366: R 0:0(0) ack 327167 win 0
07:52:13.376048 192.168.209.243.1366 > 192.168.209.245.70: S 327166:327166(0) win 8192
<mss 1460> (DF)
07:52:13.376048 192.168.209.245.70 > 192.168.209.243.1366: R 0:0(0) ack 1 win 0
07:52:13.876768 192.168.209.243.1366 > 192.168.209.245.70: S 327166:327166(0) win 8192
<mss 1460> (DF)
07:52:13.876768 192.168.209.245.70 > 192.168.209.243.1366: R 0:0(0) ack 1 win 0
07:52:14.377488 192.168.209.243.1366 > 192.168.209.245.70: S 327166:327166(0) win 8192
<mss 1460> (DF)
07:52:14.377488 192.168.209.245.70 > 192.168.209.243.1366: R 0:0(0) ack 1 win 0
[TCP 80 HTTP scans (x4)]
07:52:14.387502 192.168.209.243.1367 > 192.168.209.245.80: S 327182:327182(0) win 8192
<mss 1460> (DF)
07:52:14.387502 192.168.209.245.80 > 192.168.209.243.1367: R 0:0(0) ack 327183 win 0
07:52:14.878208 192.168.209.243.1367 > 192.168.209.245.80: S 327182:327182(0) win 8192
<mss 1460> (DF)
07:52:14.878208 192.168.209.245.80 > 192.168.209.243.1367: R 0:0(0) ack 1 win 0
07:52:15.378928 192.168.209.243.1367 > 192.168.209.245.80: S 327182:327182(0) win 8192
<mss 1460> (DF)
07:52:15.378928 192.168.209.245.80 > 192.168.209.243.1367: R 0:0(0) ack 1 win 0
07:52:15.879648 192.168.209.243.1367 > 192.168.209.245.80: S 327182:327182(0) win 8192
<mss 1460> (DF)
07:52:15.879648 192.168.209.245.80 > 192.168.209.243.1367: R 0:0(0) ack 1 win 0
[TCP 143 IMAP scans (x4)]
07:52:15.889662 192.168.209.243.1368 > 192.168.209.245.143: S 327200:327200(0) win 8192
<mss 1460> (DF)
07:52:15.889662 192.168.209.245.143 > 192.168.209.243.1368: R 0:0(0) ack 327201 win 0
07:52:16.380368 192.168.209.243.1368 > 192.168.209.245.143: S 327200:327200(0) win 8192
<mss 1460> (DF)
07:52:16.380368 192.168.209.245.143 > 192.168.209.243.1368: R 0:0(0) ack 1 win 0
```

```
07:52:16.881088 192.168.209.243.1368 > 192.168.209.245.143: S 327200:327200(0) win 8192
<mss 1460> (DF)
07:52:16.881088 192.168.209.245.143 > 192.168.209.243.1368: R 0:0(0) ack 1 win 0
07:52:17.381808 192.168.209.243.1368 > 192.168.209.245.143: S 327200:327200(0) win 8192
<mss 1460> (DF)
07:52:17.381808 192.168.209.245.143 > 192.168.209.243.1368: R 0:0(0) ack 1 win 0
[TCP 119 NNTP scans (x4)]
07:52:17.391822 192.168.209.243.1369 > 192.168.209.245.119: S 327212:327212(0) win 8192
<mss 1460> (DF)
07:52:17.391822 192.168.209.245.119 > 192.168.209.243.1369: R 0:0(0) ack 327213 win 0
07:52:17.882528 192.168.209.243.1369 > 192.168.209.245.119: S 327212:327212(0) win 8192
<mss 1460> (DF)
07:52:17.882528 192.168.209.245.119 > 192.168.209.243.1369: R 0:0(0) ack 1 win 0
07:52:18.383248 192.168.209.243.1369 > 192.168.209.245.119: S 327212:327212(0) win 8192
<mss 1460> (DF)
07:52:18.383248 192.168.209.245.119 > 192.168.209.243.1369: R 0:0(0) ack 1 win 0
07:52:18.883968 192.168.209.243.1369 > 192.168.209.245.119: S 327212:327212(0) win 8192
<mss 1460> (DF)
07:52:18.883968 192.168.209.245.119 > 192.168.209.243.1369: R 0:0(0) ack 1 win 0
[TCP 110 POP3 scans (x4)]
07:52:18.893982 192.168.209.243.1370 > 192.168.209.245.110: S 327226:327226(0) win 8192
<mss 1460> (DF)
07:52:18.893982 192.168.209.245.110 > 192.168.209.243.1370: R 0:0(0) ack 327227 win 0
07:52:19.384688 192.168.209.243.1370 > 192.168.209.245.110: S 327226:327226(0) win 8192
<mss 1460> (DF)
07:52:19.384688 192.168.209.245.110 > 192.168.209.243.1370: R 0:0(0) ack 1 win 0
07:52:19.885408 192.168.209.243.1370 > 192.168.209.245.110: S 327226:327226(0) win 8192
<mss 1460> (DF)
07:52:19.885408 192.168.209.245.110 > 192.168.209.243.1370: R 0:0(0) ack 1 win 0
07:52:20.386128 192.168.209.243.1370 > 192.168.209.245.110: S 327226:327226(0) win 8192
<mss 1460> (DF)
07:52:20.386128 192.168.209.245.110 > 192.168.209.243.1370: R 0:0(0) ack 1 win 0
[TCP 25 SMTP scans (x4)]
07:52:20.396142 192.168.209.243.1371 > 192.168.209.245.25: S 327235:327235(0) win 8192
<mss 1460> (DF)
07:52:20.396142 192.168.209.245.25 > 192.168.209.243.1371: R 0:0(0) ack 327236 win 0
07:52:20.886848 192.168.209.243.1371 > 192.168.209.245.25: S 327235:327235(0) win 8192
<mss 1460> (DF)
07:52:20.886848 192.168.209.245.25 > 192.168.209.243.1371: R 0:0(0) ack 1 win 0
07:52:21.387568 192.168.209.243.1371 > 192.168.209.245.25: S 327235:327235(0) win 8192
<mss 1460> (DF)
07:52:21.387568 192.168.209.245.25 > 192.168.209.243.1371: R 0:0(0) ack 1 win 0
07:52:21.888288 192.168.209.243.1371 > 192.168.209.245.25: S 327235:327235(0) win 8192
<mss 1460> (DF)
07:52:21.888288 192.168.209.245.25 > 192.168.209.243.1371: R 0:0(0) ack 1 win 0
[TCP 161 SNMP scans (x4)]
07:52:21.898302 192.168.209.243.1372 > 192.168.209.245.161: S 327246:327246(0) win 8192
<mss 1460> (DF)
07:52:21.898302 192.168.209.245.161 > 192.168.209.243.1372: R 0:0(0) ack 327247 win 0
07:52:22.389008 192.168.209.243.1372 > 192.168.209.245.161: S 327246:327246(0) win 8192
<mss 1460> (DF)
07:52:22.389008 192.168.209.245.161 > 192.168.209.243.1372: R 0:0(0) ack 1 win 0
07:52:22.889728 192.168.209.243.1372 > 192.168.209.245.161: S 327246:327246(0) win 8192
<mss 1460> (DF)
07:52:22.889728 192.168.209.245.161 > 192.168.209.243.1372: R 0:0(0) ack 1 win 0
07:52:23.390448 192.168.209.243.1372 > 192.168.209.245.161: S 327246:327246(0) win 8192
<mss 1460> (DF)
07:52:23.390448 192.168.209.245.161 > 192.168.209.243.1372: R 0:0(0) ack 1 win 0
[TCP 23 TELNET scans (x4)]
07:52:23.400462 192.168.209.243.1373 > 192.168.209.245.23: S 327251:327251(0) win 8192
<mss 1460> (DF)
07:52:23.400462 192.168.209.245.23 > 192.168.209.243.1373: R 0:0(0) ack 327252 win 0
07:52:23.891168 192.168.209.243.1373 > 192.168.209.245.23: S 327251:327251(0) win 8192
<mss 1460> (DF)
07:52:23.891168 192.168.209.245.23 > 192.168.209.243.1373: R 0:0(0) ack 1 win 0
07:52:24.391888 192.168.209.243.1373 > 192.168.209.245.23: S 327251:327251(0) win 8192
<mss 1460> (DF)
07:52:24.391888 192.168.209.245.23 > 192.168.209.243.1373: R 0:0(0) ack 1 win 0
07:52:24.892608 192.168.209.243.1373 > 192.168.209.245.23: S 327251:327251(0) win 8192
<mss 1460> (DF)
07:52:24.892608 192.168.209.245.23 > 192.168.209.243.1373: R 0:0(0) ack 1 win 0
```

```
[TCP 37 TIME scans (x4)]
07:52:24.902622 192.168.209.243.1374 > 192.168.209.245.37: S 327258:327258(0) win 8192
<mss 1460> (DF)
07:52:24.902622 192.168.209.245.37 > 192.168.209.243.1374: R 0:0(0) ack 327259 win 0
07:52:25.393328 192.168.209.243.1374 > 192.168.209.245.37: S 327258:327258(0) win 8192
<mss 1460> (DF)
07:52:25.393328 192.168.209.245.37 > 192.168.209.243.1374: R 0:0(0) ack 1 win 0
07:52:25.894048 192.168.209.243.1374 > 192.168.209.245.37: S 327258:327258(0) win 8192
<mss 1460> (DF)
07:52:25.894048 192.168.209.245.37 > 192.168.209.243.1374: R 0:0(0) ack 1 win 0
07:52:26.394768 192.168.209.243.1374 > 192.168.209.245.37: S 327258:327258(0) win 8192
<mss 1460> (DF)
07:52:26.394768 192.168.209.245.37 > 192.168.209.243.1374: R 0:0(0) ack 1 win 0
[TCP 63 WHOIS scans (x4)]
07:52:26.404782 192.168.209.243.1375 > 192.168.209.245.63: S 327275:327275(0) win 8192
<mss 1460> (DF)
07:52:26.404782 192.168.209.245.63 > 192.168.209.243.1375: R 0:0(0) ack 327276 win 0
07:52:26.895488 192.168.209.243.1375 > 192.168.209.245.63: S 327275:327275(0) win 8192
<mss 1460> (DF)
07:52:26.895488 192.168.209.245.63 > 192.168.209.243.1375: R 0:0(0) ack 1 win 0
07:52:27.396208 192.168.209.243.1375 > 192.168.209.245.63: S 327275:327275(0) win 8192
<mss 1460> (DF)
07:52:27.396208 192.168.209.245.63 > 192.168.209.243.1375: R 0:0(0) ack 1 win 0
07:52:27.896928 192.168.209.243.1375 > 192.168.209.245.63: S 327275:327275(0) win 8192
<mss 1460> (DF)
07:52:27.896928 192.168.209.245.63 > 192.168.209.243.1375: R 0:0(0) ack 1 win 0
<...Default scan ends, Custom scan begins...>
[TCP 135 DCE endpoint scan (NOTE: connection successful!)]
07:52:27.906942 192.168.209.243.1376 > 192.168.209.245.135: S 327294:327294(0) win 8192
<mss 1460> (DF)
07:52:27.906942 192.168.209.245.135 > 192.168.209.243.1376: S 65985:65985(0) ack 327295
win 8760 <mss 1460> (DF)
07:52:27.906942 192.168.209.243.1376 > 192.168.209.245.135: . ack 1 win 8760 (DF)
07:52:28.417676 192.168.209.243.1376 > 192.168.209.245.135: R 327295:327295(0) win 0 (DF)
[TCP 137 NetBIOS name svc scans (x4)]
07:52:28.417676 192.168.209.243.1377 > 192.168.209.245.137: S 327313:327313(0) win 8192
<mss 1460> (DF)
07:52:28.417676 192.168.209.245.137 > 192.168.209.243.1377: R 0:0(0) ack 327314 win 0
07:52:28.898368 192.168.209.243.1377 > 192.168.209.245.137: S 327313:327313(0) win 8192
<mss 1460> (DF)
07:52:28.898368 192.168.209.245.137 > 192.168.209.243.1377: R 0:0(0) ack 1 win 0
07:52:29.399088 192.168.209.243.1377 > 192.168.209.245.137: S 327313:327313(0) win 8192
<mss 1460> (DF)
07:52:29.399088 192.168.209.245.137 > 192.168.209.243.1377: R 0:0(0) ack 1 win 0
07:52:29.899808 192.168.209.243.1377 > 192.168.209.245.137: S 327313:327313(0) win 8192
<mss 1460> (DF)
07:52:29.899808 192.168.209.245.137 > 192.168.209.243.1377: R 0:0(0) ack 1 win 0
[TCP 138 NetBIOS datagram svc scans (x4)]
07:52:29.909822 192.168.209.243.1378 > 192.168.209.245.138: S 327316:327316(0) win 8192
<mss 1460> (DF)
07:52:29.909822 192.168.209.245.138 > 192.168.209.243.1378: R 0:0(0) ack 327317 win 0
07:52:30.400528 192.168.209.243.1378 > 192.168.209.245.138: S 327316:327316(0) win 8192
<mss 1460> (DF)
07:52:30.400528 192.168.209.245.138 > 192.168.209.243.1378: R 0:0(0) ack 1 win 0
07:52:30.901248 192.168.209.243.1378 > 192.168.209.245.138: S 327316:327316(0) win 8192
<mss 1460> (DF)
07:52:30.901248 192.168.209.245.138 > 192.168.209.243.1378: R 0:0(0) ack 1 win 0
07:52:31.401968 192.168.209.243.1378 > 192.168.209.245.138: S 327316:327316(0) win 8192
<mss 1460> (DF)
07:52:31.401968 192.168.209.245.138 > 192.168.209.243.1378: R 0:0(0) ack 1 win 0
[TCP 139 NetBIOS session svc scans (NOTE: successful connection!)]
07:52:31.411982 192.168.209.243.1379 > 192.168.209.245.139: S 327337:327337(0) win 8192
<mss 1460> (DF)
07:52:31.411982 192.168.209.245.139 > 192.168.209.243.1379: S 65993:65993(0) ack 327338
win 8760 <mss 1460> (DF)
07:52:31.411982 192.168.209.243.1379 > 192.168.209.245.139: . ack 1 win 8760 (DF)
07:52:31.922716 192.168.209.243.1379 > 192.168.209.245.139: R 327338:327338(0) win 0 (DF)
[TCP 3128 Squid Proxy scans (x4)]
07:52:31.922716 192.168.209.243.1380 > 192.168.209.245.3128: S 327350:327350(0) win 8192
<mss 1460> (DF)
07:52:31.922716 192.168.209.245.3128 > 192.168.209.243.1380: R 0:0(0) ack 327351 win 0
```

```
07:52:32.403408 192.168.209.243.1380 > 192.168.209.245.3128: S 327350:327350(0) win 8192
<mss 1460> (DF)
07:52:32.403408 192.168.209.245.3128 > 192.168.209.243.1380: R 0:0(0) ack 1 win 0
07:52:32.904128 192.168.209.243.1380 > 192.168.209.245.3128: S 327350:327350(0) win 8192
<mss 1460> (DF)
07:52:32.904128 192.168.209.245.3128 > 192.168.209.243.1380: R 0:0(0) ack 1 win 0
07:52:33.404848 192.168.209.243.1380 > 192.168.209.245.3128: S 327350:327350(0) win 8192
<mss 1460> (DF)
07:52:33.404848 192.168.209.245.3128 > 192.168.209.243.1380: R 0:0(0) ack 1 win 0
07:52:33.414862 192.168.209.243.1381 > 192.168.209.245.8080: S 327355:327355(0) win 8192
<mss 1460> (DF)
[TCP 8080 WinGate Proxy scans (x4)]
07:52:33.414862 192.168.209.245.8080 > 192.168.209.243.1381: R 0:0(0) ack 327356 win 0
07:52:33.905568 192.168.209.243.1381 > 192.168.209.245.8080: S 327355:327355(0) win 8192
<mss 1460> (DF)
07:52:33.905568 192.168.209.245.8080 > 192.168.209.243.1381: R 0:0(0) ack 1 win 0
07:52:34.406288 192.168.209.243.1381 > 192.168.209.245.8080: S 327355:327355(0) win 8192
<mss 1460> (DF)
07:52:34.406288 192.168.209.245.8080 > 192.168.209.243.1381: R 0:0(0) ack 1 win 0
07:52:34.907008 192.168.209.243.1381 > 192.168.209.245.8080: S 327355:327355(0) win 8192
<mss 1460> (DF)
07:52:34.907008 192.168.209.245.8080 > 192.168.209.243.1381: R 0:0(0) ack 1 win 0
<...Custom scan ends>
```

As you can see, the scan probes the ports in the order they appear in the configuration GUI shown in Figure 1. No anomalous TCP flags are set, and there is no unusual actions performed beyond completing the connection if a port is found open. For all intents and purposes, this is a vanilla TCP SYN port scan.

The only identifying features of this scan are as follows:

- Begins with ICMP echo requests; (If ICMP echo replies are received…)
- Scans the following TCP ports in this order – 53, 7, 79, 21, 70, 80, 143, 119, 110, 25, 161, 23, 37 and 63
- If and 'Custom Ports' have been identified, these will follow.

The defenses required to protect against this particular scanner are trivial. This scanner assumes there is no host present at a specific IP address if it doesn't receive a reply to its ICMP echo requests. As a result, an ACL on a router that denies all ICMP will cause this scanner to move on without scanning any TCP ports. If ICMP echoes need to be allowed into the perimeter, a firewall can be used to protect a segment from these scans. Unfortunately, this scanner will locate any ports that remain open.

Assignment 3 – "Analyze This"

Summary for the "MY.NET" network – 22 Sep 00

Based on the data provided, it was immediately apparent that there was a large volume of anomalous traffic picked up by the SNORT IDS during the period of 27 June to 10 Aug 2000. Close to 28 Mb of data were logged, however it should be noted that this does not represent the full potential of possible activity that may have occurred during the timeframe that was monitored.

Serious gaps in monitoring were noted. During the 44 calendar days that monitoring occurred, only 23 days of traffic were actually recorded by the SNORT IDS. On two separate days, (July 24th and August 3rd), more than 12 hours of traffic is missing. Whole days worth of log files are missing – these include: July 1st through the 7th, July 9th, 13th, 15th, 16th, July 18th through the 23rd, July 25th and 31st. As well, August 6th, 7th and 9th have no logs.

During the monitoring period, based on the number of port scans and other activity noted by the SNORT IDS, this site was a subject of great interest to several parties. Because of the limitations imposed by SNORT IDS (i.e. – it only logs source half of completed connections) and due to the lack of additional correlating data (i.e. – SHADOW logs, TCPdump logs, etc.), the following data is believed to be correct. This correctness must be taken within the context of the source data – it is presented wherever possible as accurate and definitive. When the lack of data makes it impossible to draw and exact conclusion, theories and estimates are presented.

The traffic has been separated into four categories: Port Scans, Compromised Hosts, Unsafe Traffic and False Positives. These distinctions have been made in an attempt to present the data in a most meaningful manner. After these categories are presented, a summary with recommendations will be made to wrap up this report.

Port Scans

This section will present fifteen examples of some of the most active sources of port scan activity. The data is presented in chronological order.

Date	Time	Signature	Source IP	Src Pt	Dest IP	Dest Pt
06/27/2000	03:34:23	spp_portscan PORTSCAN DETECTED from	193.251.35.190			
06/28/2000	06:52:48	SYN-FIN scan!	202.0.178.98	53	MY.NET.1.25	53
[Scan of entire Class-B occurs]						
06/28/2000	07:14:23	SYN-FIN scan!	202.0.178.98	53	MY.NET.254.249	53
06/28/2000	07:16:33	spp_portscan PORTSCAN DETECTED from	202.0.178.98			
07/11/2000	17:46:59	spp_portscan PORTSCAN DETECTED from	4.54.218.59			
07/11/2000	21:54:37	spp_portscan PORTSCAN DETECTED from	4.54.38.36			
07/12/2000	02:20:07	spp_portscan PORTSCAN DETECTED from	211.36.253.174			
07/12/2000	03:50:48	SUNRPC highport access!	204.137.237.8	3097	MY.NET.97.112	32771
07/12/2000	03:51:21	WinGate 1080 Attempt	204.137.237.8	1803	MY.NET.97.112	1080
07/12/2000	03:56:30	GIAC 218 VACIRT port 34555	204.137.237.8	3875	MY.NET.97.112	34555
07/12/2000	04:02:26	spp_portscan PORTSCAN DETECTED from	204.137.237.8			
07/12/2000	13:24:04	spp_portscan PORTSCAN DETECTED from	141.44.164.142			
07/12/2000	17:31:40	spp_portscan PORTSCAN DETECTED from	165.138.228.4			
07/17/2000	01:18:59	spp_portscan PORTSCAN DETECTED from	24.2.123.9			
07/19/2000	17:18:38	spp_portscan PORTSCAN DETECTED from	130.149.41.70			

Table 1 – Port Scan Activity

Compromised Hosts

Based on the traffic patterns noted, there are some hosts on MY.NET that are very likely compromised. Each host that is possibly compromised will be discussed separately below:

MY.NET.99.51 was noted in use as a WinGate Proxy for many IP addresses from the Internet. This host was also involved in some scanning activity against other MY.NET hosts.

Date	Time	Signature	Source IP	Src Pt	Dest IP	Dest Pt
06/27/2000	06:35:20	WinGate 1080 Attempt	24.93.191.134	1814	MY.NET.99.51	1080
06/27/2000	06:35:21	WinGate 1080 Attempt	24.93.191.134	1814	MY.NET.99.51	1080
06/29/2000	04:40:46	WinGate 1080 Attempt	207.114.4.46	3816	MY.NET.99.51	1080
06/30/2000	05:54:34	WinGate 1080 Attempt	207.114.4.46	4360	MY.NET.99.51	1080
06/30/2000	05:54:34	WinGate 1080 Attempt	207.114.4.46	4360	MY.NET.99.51	1080
06/30/2000	11:17:36	WinGate 1080 Attempt	24.93.191.134	2848	MY.NET.99.51	1080
06/30/2000	11:17:36	WinGate 1080 Attempt	24.93.191.134	2848	MY.NET.99.51	1080
06/30/2000	11:17:36	WinGate 1080 Attempt	24.93.191.134	2848	MY.NET.99.51	1080

06/30/2000	11:17:36	WinGate 1080 Attempt	24.93.191.134	2848	MY.NET.99.51	1080
06/30/2000	19:54:21	WinGate 1080 Attempt	24.93.191.134	4022	MY.NET.99.51	1080
06/30/2000	19:54:21	WinGate 1080 Attempt	24.93.191.134	4022	MY.NET.99.51	1080
06/30/2000	19:54:21	WinGate 1080 Attempt	24.93.191.134	4022	MY.NET.99.51	1080
06/30/2000	19:54:21	WinGate 1080 Attempt	24.93.191.134	4022	MY.NET.99.51	1080
06/30/2000	19:54:21	WinGate 1080 Attempt	24.93.191.134	4022	MY.NET.99.51	1080
06/30/2000	19:54:21	WinGate 1080 Attempt	24.93.191.134	4022	MY.NET.99.51	1080
07/10/2000	04:51:53	WinGate 1080 Attempt	204.210.198.252	4742	MY.NET.99.51	1080
07/10/2000	04:51:53	WinGate 1080 Attempt	204.210.198.252	4742	MY.NET.99.51	1080
07/10/2000	04:51:53	WinGate 1080 Attempt	204.210.198.252	4742	MY.NET.99.51	1080
07/10/2000	19:43:24	WinGate 1080 Attempt	24.0.167.156	1520	MY.NET.99.51	1080
07/10/2000	19:43:25	WinGate 1080 Attempt	24.0.167.156	1520	MY.NET.99.51	1080
07/10/2000	19:43:26	WinGate 1080 Attempt	24.0.167.156	1520	MY.NET.99.51	1080
07/10/2000	19:43:26	WinGate 1080 Attempt	24.0.167.156	1520	MY.NET.99.51	1080
07/10/2000	19:43:27	WinGate 1080 Attempt	24.0.167.156	1520	MY.NET.99.51	1080
07/10/2000	19:43:28	WinGate 1080 Attempt	216.249.0.51	3078	MY.NET.99.51	1080
07/10/2000	19:43:28	WinGate 1080 Attempt	24.0.167.156	1520	MY.NET.99.51	1080
07/10/2000	19:43:28	WinGate 1080 Attempt	24.0.167.156	1520	MY.NET.99.51	1080
07/10/2000	19:43:29	WinGate 1080 Attempt	24.0.167.156	1520	MY.NET.99.51	1080
07/10/2000	19:43:29	WinGate 1080 Attempt	24.0.167.156	1520	MY.NET.99.51	1080
07/10/2000	19:43:30	WinGate 1080 Attempt	216.249.0.51	3078	MY.NET.99.51	1080
07/11/2000	17:24:14	WinGate 1080 Attempt	24.23.132.16	2344	MY.NET.99.51	1080
07/11/2000	17:24:17	WinGate 1080 Attempt	24.23.132.16	2344	MY.NET.99.51	1080
07/11/2000	17:38:36	WinGate 1080 Attempt	24.23.132.16	3049	MY.NET.99.51	1080
07/12/2000	11:33:01	WinGate 1080 Attempt	212.158.123.66	4911	MY.NET.99.51	1080
07/17/2000	00:58:49	WinGate 1080 Attempt	24.189.238.21	1431	MY.NET.99.51	1080
07/17/2000	00:58:50	WinGate 1080 Attempt	24.189.238.21	1431	MY.NET.99.51	1080
07/17/2000	00:58:50	WinGate 1080 Attempt	24.189.238.21	1431	MY.NET.99.51	1080
07/17/2000	00:58:51	WinGate 1080 Attempt	24.189.238.21	1431	MY.NET.99.51	1080
07/17/2000	01:15:22	WinGate 1080 Attempt	24.189.238.21	1616	MY.NET.99.51	1080
07/17/2000	01:15:22	WinGate 1080 Attempt	24.189.238.21	1616	MY.NET.99.51	1080
07/17/2000	01:15:23	WinGate 1080 Attempt	24.189.238.21	1616	MY.NET.99.51	1080
07/17/2000	01:15:23	WinGate 1080 Attempt	24.189.238.21	1616	MY.NET.99.51	1080
07/17/2000	01:15:31	WinGate 1080 Attempt	195.75.32.144	1312	MY.NET.99.51	1080
07/17/2000	01:15:33	WinGate 1080 Attempt	195.75.32.144	1312	MY.NET.99.51	1080
07/17/2000	01:15:39	WinGate 1080 Attempt	195.75.32.144	1312	MY.NET.99.51	1080
07/17/2000	01:18:03	WinGate 1080 Attempt	24.189.238.21	1655	MY.NET.99.51	1080
07/17/2000	01:18:04	WinGate 1080 Attempt	24.189.238.21	1655	MY.NET.99.51	1080
07/17/2000	01:18:04	WinGate 1080 Attempt	24.189.238.21	1655	MY.NET.99.51	1080
07/17/2000	01:18:05	WinGate 1080 Attempt	195.75.32.144	1341	MY.NET.99.51	1080
07/17/2000	01:18:05	WinGate 1080 Attempt	24.189.238.21	1655	MY.NET.99.51	1080
07/17/2000	01:18:06	WinGate 1080 Attempt	195.75.32.144	1341	MY.NET.99.51	1080
07/17/2000	01:18:08	WinGate 1080 Attempt	195.75.32.144	1341	MY.NET.99.51	1080
07/17/2000	01:22:31	WinGate 1080 Attempt	24.189.238.21	1706	MY.NET.99.51	1080
07/17/2000	01:22:32	WinGate 1080 Attempt	24.189.238.21	1706	MY.NET.99.51	1080
07/17/2000	01:22:33	WinGate 1080 Attempt	24.189.238.21	1706	MY.NET.99.51	1080

07/17/2000	01:22:34	WinGate 1080 Attempt	195.75.32.144	1362	MY.NET.99.51	1080
07/17/2000	01:32:24	WinGate 1080 Attempt	195.75.32.144	1411	MY.NET.99.51	1080
07/17/2000	01:32:27	WinGate 1080 Attempt	195.75.32.144	1411	MY.NET.99.51	1080
07/17/2000	01:32:29	WinGate 1080 Attempt	195.75.32.144	1411	MY.NET.99.51	1080
07/17/2000	09:40:10	WinGate 1080 Attempt	202.138.47.233	1340	MY.NET.99.51	1080
07/17/2000	09:40:12	WinGate 1080 Attempt	202.138.47.233	1340	MY.NET.99.51	1080
07/26/2000	02:46:25	WinGate 1080 Attempt	207.114.4.46	3875	MY.NET.99.51	1080
07/28/2000	05:44:51	WinGate 1080 Attempt	207.114.4.46	1272	MY.NET.99.51	1080

Table 2 – Compromised hosts being used as WinGate proxy

Of particular concern was an indication from the SNORT IDS that the TELNET daemon on MY.NET.99.51 was activated on one occasion, which would suggest that the attacker had decided to open this service to improve connectivity to the compromised host.

Date	Time	Signature	Source IP	Src Pt	Dest IP	Dest Pt
08/05/2000	19:03:45	IDS08 - TELNET - daemon-active	MY.NET.99.51	23	24.25.111.117	1029

Table 3 – Compromised hosts running TELNET daemon

Another possible compromised host is MY.NET.253.41. This host was noted acting in the capacity of mail server for addresses not belonging to MY.NET.

Date & Hour	Mins	Secs	Signature	Source IP	Src Pt	Dest IP	Dest Pt
06/28/2000 10	47	54	Watchlist 222 NET-NCFC	159.226.115.1	33364	MY.NET.253.41	25
06/28/2000 11	50	33	Watchlist 222 NET-NCFC	159.226.45.3	113	MY.NET.253.41	35075

Table 4 – Compromised host acting as mail server

There is no other mail-related traffic noted for MY.NET.253.41, but this may be due to the SNORT rule used. This host should be checked to ensure it is serving in its intended role and is not being used inappropriately. If MY.NET.253.41 is a mail server, it might be prudent to verify that it is not being used in an undesirable manner by IP addresses belonging to Watchlist 222.

There was also a fair volume of traffic attributed to IP addresses associated with Watchlist 220. While most of this traffic appeared to be related to the use of Napster (see the “Unsafe Traffic” section below), this particular pattern could not be identified.

Date & Hour	Mins	Secs	Signature	Source IP	Src Pt	Dest IP	Dest Pt
07/17/2000 11	50	58	Watchlist 220 ILISDNNET990517	212.179.4.238	1072	MY.NET.53.28	4110

Table 4 – Traffic involving an unknown service

This host needs to be inspected to see if what software or application might be using TCP port 4110. This port, while not readily identified as being associated with one, may represent a backdoor.

Unsafe Traffic

The use of several applications that, depending on the security and “acceptable use” policies governing this network, might be considered undesirable were noted on numerous occasions throughout the SNORT logs. Applications such as Napster, ICQ, and Real Player have been picked up in use – these detects have actually occurred on the most part because of SNORT filter rules that are actually intended to detect other activity.

ICQ – this application allows a user to chat, share files and communicate with other users around the world via the Internet. On first glance, hosts like MY.NET.217.126 appear to be compromised by a user operating from IP 205.188.153.11 because of the volume of traffic that originated from outside of the network. On closer inspection it became apparent that, despite the alarms involving SNORT signature “Attempted Sun RPC high port access,” this traffic was not malicious in nature.

Date	Time	Signature	Source IP	Src Pt	Dest IP	Dest Pt
08/03/2000	11:15:26	Attempted Sun RPC high port access	205.188.153.111	4000	MY.NET.217.126	32771
08/03/2000	11:16:25	Attempted Sun RPC high port access	205.188.153.111	4000	MY.NET.217.126	32771
08/03/2000	11:17:25	Attempted Sun RPC high port access	205.188.153.111	4000	MY.NET.217.126	32771
08/03/2000	11:19:25	Attempted Sun RPC high port access	205.188.153.111	4000	MY.NET.217.126	32771
08/03/2000	11:20:25	Attempted Sun RPC high port access	205.188.153.111	4000	MY.NET.217.126	32771

Table 5 – ICQ-related traffic

Because this traffic always has a source port of 4000, a hostname resolution attempt was made using Whois. Fortunately, IP 205.188.153.111 resolves to fes-d015.icq.aol.com – this is a well-known ICQ server, and it probably serves as an alternate to the main ICQ server residing at IP 205.188.153.112 (which resolves to icq.mirabilis.com). Notice to the interval that these logs are occurring. ICQ has a keep-alive feature that helps the user’s client keep in sync with the server. This is useful for helping the user track the other ICQ users that they have added to their “contact list,” since its function is to notify the user when these people come online with ICQ. Also, ICQ can be configured to run at system startup – this accounts for some of the logs showing MY.NET.217.126 receiving these apparent ICQ keep-alive packets at wee hours of the morning.

Napster – this application was designed primarily with the purpose of sharing compressed music files (MP3s) in mind. It is a very robust application when it comes to connectivity. It is known to use any method possible to connect to the Napster server and other users of the program. It can even be set to attempt using well-known service ports to pass its traffic in an attempt to circumvent firewalls. There are other such applications, such as Gnutella, but with the popularity of the Napster utility amongst its user community, this seems the likely choice.

Date & Hour	Mins	Secs	Signature	Source IP	Src Pt	Dest IP	Dest Pt
06/28/2000 02	43	30	Watchlist 220 ILISDNNET990517	212.179.23.4	6699	MY.NET.179.51	1088
07/26/2000 04	50	11	Watchlist 220 ILISDNNET990517	212.179.54.69	6699	MY.NET.182.94	3661
08/01/2000 01	21	6	Watchlist 220 ILISDNNET990517	212.179.38.141	2792	MY.NET.217.38	6699

Table 6 – Napster-related traffic

The SNORT logs are literally jammed with Napster file transfers from IP addresses on Watchlist 220 to addresses within MY.NET. In one instance, an address picked up by the same SNORT rule noted a file being downloaded from MY.NET.217.38. The only reason this activity appears to be getting logged is due to the watchlist.

Finally, MY.NET.101.192 has shown up through the detection of SNMP-related traffic by the SNORT IDS. Normally, SNMP is associated with network administration and infrastructure performance monitoring/tuning. This address most likely appeared in the SNORT IDS logs due to the used of the SNMP “Public” access group by MY.NET.101.192 to query hosts like MY.NET.97.87, MY.NET.101.160 and MY.NET.97.237.

Date & Hour	Mins	Secs	Signature	Source IP	Src Pt	Dest IP	Dest Pt
07/11/2000 15	55	47	SNMP public	MY.NET.97.122	1059	MY.NET.101.192	161
07/14/2000 08	13	15	SNMP public	MY.NET.97.237	1041	MY.NET.101.192	161
07/14/2000 11	28	7	SNMP public	MY.NET.97.80	1623	MY.NET.101.192	161

Table 7 – MY.NET.101.192 using SNMP to manage hosts/devices

This traffic is probably normal for this network, but should be reconsidered due to the vulnerabilities associated with the use of this common SNMP group name.

False Positives

Several of the detects noted by the SNORT IDS were the result of a rule that might have worked better if they were fine-tuned. In some cases, the detects were the result of legitimate activity triggering a rule that is still worth maintaining.

Date & Hour	Mins	Secs	Signature	Source IP	Src Pt	Dest IP	Dest Pt
06/27/2000 01	58	2	GIAC 218 VACIRT port 35555	209.132.14.35	25	MY.NET.6.34	35555

Table 8 – Example of false Trinoo traffic detection

Date & Hour	Mins	Secs	Signature	Source IP	Src Pt	Dest IP	Dest Pt
06/28/2000 07	35	7	GIAC 218 VACIRT port 34555	165.251.8.33	25	MY.NET.253.24	34555
06/28/2000 07	35	7	GIAC 218 VACIRT port 34555	165.251.8.33	25	MY.NET.253.24	34555
06/28/2000 07	35	7	GIAC 218 VACIRT port 34555	165.251.8.33	25	MY.NET.253.24	34555
06/28/2000 07	35	7	GIAC 218 VACIRT port 34555	165.251.8.33	25	MY.NET.253.24	34555
06/28/2000 07	35	8	GIAC 218 VACIRT port 34555	165.251.8.33	25	MY.NET.253.24	34555
06/28/2000 07	35	9	GIAC 218 VACIRT port 34555	165.251.8.33	25	MY.NET.253.24	34555

Table 9 – Another example of false Trinoo traffic detection

Date & Hour	Mins	Secs	Signature	Source IP	Src Pt	Dest IP	Dest Pt
07/30/2000 22	49	42	Watchlist 222 NET-NCFC	159.226.5.152	719	MY.NET.100.165	80

Table 10 – A Watchlist 222 involving suspected harmless web surfing

Assuming that the IP addresses involved in these detects are functioning within the parameters of the assigned uses (i.e. – mail server, web server), there is no reason for concern. While it might be good to track if a person from China (Watchlist 222) is surfing your website, it is no real cause for alarm.

Recommendations

- Fine-tune some of the SNORT rules (i.e. – some of the watchlists) to log more carefully and reduce the likelihood of false alarms. While false positives are considered unavoidable when using Network IDS technologies, such tuning can help reduce the overall number of these false alarms significantly;
- Review the security policy of this network and confirm the acceptability of certain software noted in the “Unsafe Traffic” section of this report. Many of these applications are potentially unsafe because of the unintended access they provide to the host on which the software is in use. By preventing the use of such software, the likelihood of being victimized through a vulnerability associated with these programs is eliminated and the general security posture of the network is greatly improved;
- Continue to monitor the hosts noted in the “Compromised Hosts” section of this report. While no confirmation has been made as to whether or not these hosts are indeed under the control of an external attacker, it is prudent that these IP addresses be observed for further activity that is normally attributed to unauthorized access. A detailed vulnerability analysis scan should be run on these hosts ASAP; and,
- Consider using another IDS to complement SNORT. While SNORT is good at detecting the general level of anomalous activity on a given network, it is not useful for analysis on its own. The use of another tool such as SHADOW is encouraged, largely in part because of the bigger picture that it can provide through its TCPdump filters. The use of these two IDS applications together can provide a more meaningful picture of the activity (both good and bad) on the network.

Conclusion

This network is very busy, with many different types of hosts active on it. The user community that utilizes this network may need to be reminded of the need to be ever aware of the threats to the security of their systems and the network. To compliment the recommendations made above, a general security awareness program should be implemented, with an emphasis placed on computer security. If this program already exists and is ongoing, it should be reviewed and adjusted to take this matter into consideration.

To ensure the safety of this network, a fairly intensive effort will likely be required in the short-term; both in regards on network monitoring and vulnerability analysis scans against the systems in use. This may require the use of several teams rotating through shifts covering these activities 24/7 for about a week. The costs of this effort will depend largely on the number of personnel assigned to the task and the whether or not the installation of SHADOW is pursued.

A detailed costing can be provided once the scope of the improvements on the security monitoring procedures desired is determined.

Assignment 4 – Analysis Process

Based on the assignment parameters, the sheer bulk of the data that needed to be analyzed (28 megs!!!), the relatively short timeframe allowed to conduct the analysis on this data, it became clear there was a lot to consider before writing the required report. It became quickly apparent that priorities had to be set and a plan devised in order to succeed with this assignment.

Based on these considerations, I decided to take a "high-level summary" approach to generate a report suitable for presenting to an organization's Executive. There is enough detail to convince the technical staff of this organization the necessity for adopting the recommends I will make, and it will also convince the Executive (especially the CEO and/or CFO) that the financial resources required to improve the security posture is well worth the expense.

The traffic was compiled into a database. This was accomplished by one of my co-workers (Jamie French) through the use of MS Access 97. He downloaded the log files and imported them into a custom database. After some formatting and after removing some spurious data, he had a very usable information store that made it easy to sort the logs by meaningful criteria. Sorts were done on Date/Time, Source IP, Destination IP and Destination Port to get an impression of what attacks had been made on the network and to determine the probable sources of these attacks. Jamie's database allowed me to get some very good snapshots of what going on without having to spend literally days trying to piece through the SNORT logs.

I still felt there was too much data to allow me to make an organized and useful report. While preparing for my CISSP examination, if have found it has been stated many times that you have to get the attention and support of your organization's Executive if you want to be taken seriously when it comes to addressing security concerns and getting support for a complete security program. I've also heard it said that you have a very limited attention span to deal with when it comes to presenting your finding to these executives. Quite frankly, they usually have better things to do then listen to someone drone on incessantly about security (at least they do as far as they are concerned!)

I looked at the assignment and read again "your organization has been asked to provide a bid to provide security services" and thought about the level involved in making decisions when it comes to this type of thing… I decided it might be a good idea to focus some and make sure this was something a CEO or CFO would want to pay attention to. I separated the traffic I had in the database into four (4) fairly logical categories: Port Scans, Compromised Hosts, Unsafe Traffic and False Positives.

At this point, I was ready to write my “presentation” and decided there were several great examples I could use as my guide. The analysis presented by Joe Church, Al Evans and Andy Siske that was based on “Laurie’s network” detects that are posted on the GIAC site proved to be one I chose as my best guide to formatting my presentation. This analysis can be found at the following URL: http://www.sans.org/y2k/092000.htm

Furthermore, I found that Lenny Zeltser’s submission for the Intrusion Detection Curriculum at SANS Security DC 2000 contained an excellent example of how best to present specific technical details from the SNORT logs in a clear and concise manner. This analysis can be found at the following URL: http://www.sans.org/y2k/Lenny Zeltser.htm

With these examples, some collaboration with my co-workers with regards to which hosts might be compromised etc., I set out to write my presentation – hopefully these methods have helped me succeed… ☺