Global Information Assurance Certification Paper

Intrusion Detection in Depth
GCIA Practical Assignment
Version 2.8

Author: Benjamin M.A. Robson

Assignment 1 - Network Detects

Detect Number 1

Detect of scan for 'Fore' Trojan presence on network.

Syslog:ipflog

```
Apr 13 13:11:36 02e ipmon[19498]: 13:11:35.966485 de0 @0:3 b 210.12.75.1,21 -> MY.NET.179.231,21 PR tcp len 20 40 -SF IN
Apr 13 13:11:36 02e ipmon[19498]: 13:11:35.969979 de0 @0:3 b 210.12.75.1,21 -> MY.NET.179.232,21 PR tcp len 20 40 -SF IN
Apr 13 13:11:36 02e ipmon[19498]: 13:11:36.005812 de0 @0:3 b 210.12.75.1,21 -> MY.NET.179.233,21 PR tcp len 20 40 -SF IN
Apr 13 13:11:36 02e ipmon[19498]: 13:11:36.025677 de0 @0:3 b 210.12.75.1,21 -> MY.NET.179.234,21 PR tcp len 20 40 -SF IN
Apr 13 13:11:36 02e ipmon[19498]: 13:11:36.046060 de0 @0:3 b 210.12.75.1,21 -> MY.NET.179.235,21 PR tcp len 20 40 -SF IN
Apr 13 13:11:36 02e ipmon[19498]: 13:11:36.070675 de0 @0:3 b 210.12.75.1,21 -> MY.NET.179.236,21 PR tcp len 20 40 -SF IN
Apr 13 13:11:36 02e ipmon[19498]: 13:11:36.085864 de0 @0:3 b 210.12.75.1,21 -> MY.NET.179.237,21 PR tcp len 20 40 -SF IN
Apr 13 13:11:36 02e ipmon[19498]: 13:11:36.090743 de0 @0:3 b 210.12.75.1,21 -> MY.NET.179.238,21 PR tcp len 20 40 -SF IN
Apr 13 13:11:36 02e ipmon[19498]: 13:11:36.125715 de0 @0:3 b 210.12.75.1,21 -> MY.NET.179.239,21 PR tcp len 20 40 -SF IN
```

TCPDUMP:

```
06:11:35.966317 < 210.12.75.1.ftp > MY.NET.179.231.ftp: SF 238035162:238035162(0) win 1028 [tos 0x10]
06:11:35.966549 < MY.NET.179.231.ftp > 210.12.75.1.ftp: R 0:0(0) ack 238035163 win 0 [tos 0x10]
06:11:35.969830 < 210.12.75.1.ftp > MY.NET.179.232.ftp: SF 238035162:238035162(0) win 1028 [tos 0x10]
06:11:35.970030 < MY.NET.179.232.ftp > 210.12.75.1.ftp: R 0:0(0) ack 238035163 win 0 [tos 0x10]
06:11:36.005666 < 210.12.75.1.ftp > MY.NET.179.233.ftp: SF 238035162:238035162(0) win 1028 [tos 0x10]
06:11:36.005866 < MY.NET.179.233.ftp > 210.12.75.1.ftp: R 0:0(0) ack 238035163 win 0 [tos 0x10]
06:11:36.025530 < 210.12.75.1.ftp > MY.NET.179.234.ftp: SF 238035162:238035162(0) win 1028 [tos 0x10]
06:11:36.025729 < MY.NET.179.234.ftp > 210.12.75.1.ftp: R 0:0(0) ack 238035163 win 0 [tos 0x10]
06:11:36.045911 < 210.12.75.1.ftp > MY.NET.179.235.ftp: SF 238035162:238035162(0) win 1028 [tos 0x10]
06:11:36.046112 < MY.NET.179.235.ftp > 210.12.75.1.ftp: R 0:0(0) ack 238035163 win 0 [tos 0x10]
06:11:36.070528 < 210.12.75.1.ftp > MY.NET.179.236.ftp: SF 238035162:238035162(0) win 1028 [tos 0x10]
06:11:36.070727 < MY.NET.179.236.ftp > 210.12.75.1.ftp: R 0:0(0) ack 238035163 win 0 [tos 0x10]
06:11:36.085717 < 210.12.75.1.ftp > MY.NET.179.237.ftp: SF 238035162:238035162(0) win 1028 [tos 0x10]
06:11:36.085917 < MY.NET.179.237.ftp > 210.12.75.1.ftp: R 0:0(0) ack 238035163 win 0 [tos 0x10]
06:11:36.090598 < 210.12.75.1.ftp > MY.NET.179.238.ftp: SF 238035162:238035162(0) win 1028 [tos 0x10]
06:11:36.090795 < MY.NET.179.238.ftp > 210.12.75.1.ftp: R 0:0(0) ack 238035163 win 0 [tos 0x10]
06:11:36.125565 < 210.12.75.1.ftp > MY.NET.179.239.ftp: SF 1013034023:1013034023(0) win 1028 [tos 0x10]
06:11:36.125768 < MY.NET.179.239.ftp > 210.12.75.1.ftp: R 0:0(0) ack 1013034024 win 0 [tos 0x10]
```

1. Source of Trace:
   This trace was detected on a customer firewall. The firewall is an OpenBSD based system using IPFilter.

   Source 1: IPFilter Syslog

   Source 2: TCPDUMP

2. Detect was generated by:

   - IPFilter Syslog
   Format: (Disinteresting things will be marked "<stuff>"
   <Date & Time> <Hostname> <Data source> <Unix Time> <Interface Label> <stuff> <blocked/passed/etc..> <src addr>,<src port> -> <dst addr>,<dst port> PR <protocol> <protocol specific details>

   - Tcpdump

3. Probability the source address was spoofed:
   Unlikely.
   It is unlikely that this perpetrator is spoofing their IP address. It is likely that the perpetrator is scanning for machines infected with the Fore Trojan.

4. Description of attack:
   The perpetrator of this detect is scanning for machines that have been compromised with the Fore Trojan. The Fore Trojan client utility, used by script-kiddies, scans for vulnerable machines listening on port 21 using a source port of 21.

5. Attack Mechanism:
   The attacker uses the Fore Trojan client to scan through a range of addresses to locate machines that have been infected with the Fore Trojan.

6. Correlation:
   This attack has been seen on numerous devices across disparate networks, and random time intervals. Thus firming the conclusion in the attacker using the ForeClient script-kiddie tool.

7. Evidence of active targeting:
   There is no evidence of active targeting. This appears to be a systematic scan of network spaces to determine if vulnerable hosts reside there-in.

8. Severity:
   Severity = (Criticality + Lethality) - (System Countermeasures + Network Countermeasures)
   Severity = (4 + 2) - (2 + 5) = -1
   Hosts machines are in DMZ (4); Attack is network probe (2); Systems could host trojan (2); Firewall does not allow inbound port 21 (5)

9. Defensive recommendations:

It would be recommended that updates to all systems be applied to ensure anti-virus applications are current and that other servers do not have services listening on port 21.

10. Multiple choice test question:

    If packets arrive at the firewall...

```
06:11:35.966317 < 210.12.75.1.ftp > MY.NET.179.231.ftp: SF 238035162:238035162(0) win 1028 [tos 0x10]
06:11:35.969830 < 210.12.75.1.ftp > MY.NET.179.232.ftp: SF 238035162:238035162(0) win 1028 [tos 0x10]
06:11:36.005666 < 210.12.75.1.ftp > MY.NET.179.233.ftp: SF 238035162:238035162(0) win 1028 [tos 0x10]
```

    what is the attacker likely to be probing for?

    a) FTP Warez Servers
    b) Hosts infected with a Trojan
    c) FTP Servers susceptible to Buffer Overflow
    d) A firewall allowing through Source Port 21

    b

Detect Number 2

Probe for DNS servers and inverse query.

N.B. Access to private address ranges 10.X.X.X and 172.16.X.X are via the use of inbound NAT.

Data Source: Syslog: ipflog:

```
Apr 12 07:20:04 MYHOST ipmon[4275]: 07:20:03.402051 de0 @0:173 b 63.144.78.244,1428 -> 10.1.1.18,53 PR tcp len 20 60 -S IN
Apr 12 07:20:04 MYHOST ipmon[4275]: 07:20:03.410791 de0 @0:173 b 63.144.78.244,1429 -> 10.1.1.8,53 PR tcp len 20 60 -S IN
Apr 12 07:20:04 MYHOST ipmon[4275]: 07:20:03.412327 de0 @0:173 b 63.144.78.244,1430 -> 10.1.1.2,53 PR tcp len 20 60 -S IN
Apr 12 07:20:04 MYHOST ipmon[4275]: 07:20:03.413488 de0 @0:173 b 63.144.78.244,1433 -> 10.1.1.16,53 PR tcp len 20 60 -S IN
Apr 12 07:20:04 MYHOST ipmon[4275]: 07:20:03.417833 de0 @0:173 b 63.144.78.244,1434 -> 10.1.1.20,53 PR tcp len 20 60 -S IN

.... Range Skipped for brevity ....

Apr 12 07:20:06 MYHOST ipmon[4275]: 07:20:05.209280 de0 @0:15 b 63.144.78.244,1618 -> MY.NET.56.222,53 PR tcp len 20 60 -S IN
Apr 12 07:20:06 MYHOST ipmon[4275]: 07:20:05.212050 de0 @0:15 b 63.144.78.244,1619 -> MY.NET.56.223,53 PR tcp len 20 60 -S IN

.... Note next blocked packet is UDP as a server was located on the address MY.NET.56.94 .....

Apr 12 07:20:06 MYHOST ipmon[4275]: 07:20:05.873563 de0 @0:77 b 63.144.78.244,1332 -> MY.NET.56.94,53 PR udp len 20 51 IN

.... Scan continues but is blocked by firewall ....

Data Source: TCPDUMP

17:20:03.401595 < 63.144.78.244.1428 > MY.NET.56.33.domain: S 1735927943:1735927943(0) win 32120 <mss 1460,sackOK,timestamp 13633316 0,nop,wscale 0> (DF) [tos
0x10]
17:20:03.402108 < MY.NET.56.62 > 63.144.78.244: icmp: net 10.1.1.18 unreachable [tos 0x10]
17:20:03.410371 < 63.144.78.244.1429 > MY.NET.56.34.domain: S 1724672241:1724672241(0) win 32120 <mss 1460,sackOK,timestamp 13633317 0,nop,wscale 0> (DF) [tos
0x10]
17:20:03.410841 < MY.NET.56.62 > 63.144.78.244: icmp: net 10.1.1.8 unreachable [tos 0x10]
17:20:03.411984 < 63.144.78.244.1430 > MY.NET.56.35.domain: S 1734052658:1734052658(0) win 32120 <mss 1460,sackOK,timestamp 13633317 0,nop,wscale 0> (DF) [tos
0x10]
17:20:03.412376 < MY.NET.56.62 > 63.144.78.244: icmp: net 10.1.1.2 unreachable [tos 0x10]

.... Scan Skipped for brevity ....

17:20:04.338254 < 63.144.78.244.1488 > MY.NET.56.93.domain: S 1727658165:1727658165(0) win 32120 <mss 1460,sackOK,timestamp 13633410 0,nop,wscale 0> (DF) [tos
0x10]

.... Scan finds DNS Server ....

17:20:04.342175 < 63.144.78.244.1489 > MY.NET.56.94.domain: S 1728107411:1728107411(0) win 32120 <mss 1460,sackOK,timestamp 13633410 0,nop,wscale 0> (DF) [tos
0x10]
17:20:04.342680 < MY.NET.56.94.domain > 63.144.78.244.1489: S 1389539253:1389539253(0) ack 1728107412 win 16500 <mss 500,nop,nop,sackOK,nop,wscale
0,nop,nop,timestamp 7077910 13633410>

.... Scan Continues ....

17:20:04.342843 < 63.144.78.244.1490 > MY.NET.56.95.domain: S 1722774860:1722774860(0) win 32120 <mss 1460,sackOK,timestamp 13633411 0,nop,wscale 0> (DF) [tos
0x10]
17:20:04.344757 < 63.144.78.244.1491 > MY.NET.56.96.domain: S 1731053569:1731053569(0) win 32120 <mss 1460,sackOK,timestamp 13633411 0,nop,wscale 0> (DF) [tos
0x10]
17:20:04.345142 < MY.NET.56.62 > 63.144.78.244: icmp: net MY.NET.56.96 unreachable [tos 0x10]
17:20:04.346667 < 63.144.78.244.1492 > MY.NET.56.97.domain: S 1723862840:1723862840(0) win 32120 <mss 1460,sackOK,timestamp 13633411 0,nop,wscale 0> (DF) [tos
0x10]
17:20:04.347044 < MY.NET.56.62 > 63.144.78.244: icmp: net MY.NET.56.97 unreachable [tos 0x10]

.... Scan skipped for brevity ....

17:20:04.887782 < 63.144.78.244.1601 > MY.NET.56.205.domain: S 1724531964:1724531964(0) win 32120 <mss 1460,sackOK,timestamp 13633465 0,nop,wscale 0> (DF) [tos
0x10]
17:20:04.887800 < 63.144.78.244.1602 > MY.NET.56.206.domain: S 1729142440:1729142440(0) win 32120 <mss 1460,sackOK,timestamp 13633465 0,nop,wscale 0> (DF) [tos
0x10]
17:20:04.888454 < MY.NET.56.62 > 63.144.78.244: icmp: net MY.NET.56.205 unreachable [tos 0x10]
17:20:04.888457 < MY.NET.56.62 > 63.144.78.244: icmp: net MY.NET.56.206 unreachable [tos 0x10]

.... Detect returns to 56.94 ....

17:20:04.942840 < 63.144.78.244.1489 > MY.NET.56.94.domain: F 1:1(0) ack 1 win 32696 <nop,nop,timestamp 13633470 7077910> (DF) [tos 0x10]
17:20:04.943335 < MY.NET.56.94.domain > 63.144.78.244.1489: . 1:1(0) ack 2 win 16500 <nop,nop,timestamp 7077911 13633470>
17:20:04.943642 < MY.NET.56.94.domain > 63.144.78.244.1489: F 1:1(0) ack 2 win 16500 <nop,nop,timestamp 7077911 13633470>

.... Detect returns to scan ...

17:20:05.184122 < 63.144.78.244.1603 > MY.NET.56.207.domain: S 1732638767:1732638767(0) win 32120 <mss 1460,sackOK,timestamp 13633495 0,nop,wscale 0> (DF) [tos
0x10]
```

```
17:20:05.184497 < 63.144.78.244.1604 > MY.NET.56.208.domain: S 1722169865:1722169865(0) win 32120 <mss 1460,sackOK,timestamp 13633495 0,nop,wscale 0> (DF) [tos
0x10]
17:20:05.184505 < MY.NET.56.62 > 63.144.78.244: icmp: net MY.NET.56.207 unreachable [tos 0x10]
17:20:05.184877 < MY.NET.56.62 > 63.144.78.244: icmp: net MY.NET.56.208 unreachable [tos 0x10]
17:20:05.187531 < 63.144.78.244.1605 > MY.NET.56.209.domain: S 1735525544:1735525544(0) win 32120 <mss 1460,sackOK,timestamp 13633495 0,nop,wscale 0> (DF) [tos
0x10]
17:20:05.188899 < 63.144.78.244.1606 > MY.NET.56.210.domain: S 1731234373:1731234373(0) win 32120 <mss 1460,sackOK,timestamp 13633495 0,nop,wscale 0> (DF) [tos
0x10]
17:20:05.188908 < 63.144.78.244.1607 > MY.NET.56.211.domain: S 1732061380:1732061380(0) win 32120 <mss 1460,sackOK,timestamp 13633495 0,nop,wscale 0> (DF) [tos
0x10]
17:20:05.188913 < MY.NET.56.62 > 63.144.78.244: icmp: net MY.NET.56.209 unreachable [tos 0x10]
17:20:05.189608 < MY.NET.56.62 > 63.144.78.244: icmp: net MY.NET.56.210 unreachable [tos 0x10]
17:20:05.189620 < MY.NET.56.62 > 63.144.78.244: icmp: net MY.NET.56.211 unreachable [tos 0x10]

.... Scan skipped for brevity ....


.... Detect returns to 56.94 whilst continuing scan in background (scan removed to highlight 56.94)

17:20:05.206181 < 63.144.78.244.1489 > MY.NET.56.94.domain: . 2:2(0) ack 2 win 32696 <nop,nop,timestamp 13633497 7077911> (DF) [tos 0x10]
17:20:05.608794 < 63.144.78.244.1689 > MY.NET.56.94.domain: S 1732530138:1732530138(0) win 32120 <mss 1460,sackOK,timestamp 13633537 0,nop,wscale 0> (DF) [tos
0x10]
17:20:05.609386 < MY.NET.56.94.domain > 63.144.78.244.1689: S 1389796725:1389796725(0) ack 1732530139 win 16500 <mss 500,nop,nop,sackOK,nop,wscale
0,nop,nop,timestamp 7077912 13633537>
17:20:05.872641 < 63.144.78.244.1689 > MY.NET.56.94.domain: . 1:1(0) ack 1 win 32120 <nop,nop,timestamp 13633563 7077912> (DF) [tos 0x10]

.... Detect probes UDP for inverse query on 56.96 and gets rejected by firewall ....

17:20:05.873241 < 63.144.78.244.1332 > MY.NET.56.94.domain: 43981 inv_q+ [b2&3=0x980] A? . (23) [tos 0x10] 17:20:05.873619 < MY.NET.56.62 > 63.144.78.244: icmp:
net MY.NET.56.94 unreachable [tos 0x10]

.... Completes Three-Way-Handshake and closes connection....

17:21:21.270184 < MY.NET.56.94.domain > 63.144.78.244.1689: . 0:0(0) ack 1 win 16500
17:21:21.532147 < 63.144.78.244.1689 > MY.NET.56.94.domain: . 1:1(0) ack 1 win 32696 <nop,nop,timestamp 13641130 7077912> (DF) [tos 0x10]
17:41:40.514406 < 63.144.78.244.1689 > MY.NET.56.94.domain: F 1:1(0) ack 1 win 32696 <nop,nop,timestamp 13763039 7077912> (DF) [tos 0x10]
17:41:40.514977 < MY.NET.56.94.domain > 63.144.78.244.1689: . 1:1(0) ack 2 win 16500 <nop,nop,timestamp 7080468 13763039>
17:41:40.515114 < MY.NET.56.94.domain > 63.144.78.244.1689: F 1:1(0) ack 2 win 16500 <nop,nop,timestamp 7080468 13763039>
17:41:40.777536 < 63.144.78.244.1689 > MY.NET.56.94.domain: . 2:2(0) ack 2 win 32696 <nop,nop,timestamp 13763065 7080468> (DF) [tos 0x10]
```

1. Source of Trace:
   This trace came from a customer firewall appliance.

2. Detect was generated by:
   Source 1: Syslog : ipflog

   This source was used to initially identify the detect and its significance.

   Source 2: TCPDUMP

   This source was used by correlating data from the Syslog : ipflog file against the tcpdump output to gather further information as to what communications were successful.

3. Probability the source address was spoofed:
   None.
   Until the perpetrator established a three-way-handshake the probe well may have had a spoofed source address. However as soon as they needed to establish a connection the likely hood of it being spoofed was removed.

4. Description of attack:
   It is the intention of this attack to locate DNS servers that are open to the world and then perform an inverse query against them. This can then lead to a buffer overflow attack against the victim host.
   CVE Reference: CVE-1999-0009

5. Attack mechanism:
   It is likely that this attack is being implemented through the use of a pre-packaged script-kiddie tool. Such a tool will allow the attacker to probe a network range for listening DNS servers and once found probe them for their version. Upon determining the version the attacker can acquire an appropriate tool to perform a buffer overflow attack, thus allowing the execution of arbitrary code.

6. Correlation:
   Although this is a common buffer overflow attack no correlating data has on other devices for this source has been located.

7. Evidence of active targeting:
   This appears to be a broad-ranging scan for DNS servers. The entire range of public addresses were scanned and thus presumably further networks were scanned beyond this one.

8. Severity:

   (4 + 5) - (3 + 5) = 1
   System is primary DNS server (4),
   Attack can DoS machine or worse (5),
   NAMEd on server is not too old (3),
   Firewall blocked UDP access (5)

9. Defensive recommendations:
   It would be highly recommended that the NAMEd service running on the server be brought up to the most recent version. This will ensure that all known vulnerabilities are patched against.

10. Multiple Choice test question:

    ```
    17:20:04.887782 < 63.144.78.244.1601 > MY.NET.56.205.domain: S 1724531964:1724531964(0) win 32120 <mss 1460,sackOK,timestamp 13633465 0,nop,wscale 0> (DF)
    ```

```
[tos 0x10]
17:20:04.887800 < 63.144.78.244.1602 > MY.NET.56.206.domain: S 1729142440:1729142440(0) win 32120 <mss 1460,sackOK,timestamp 13633465 0,nop,wscale 0> (DF)
[tos 0x10]
17:20:04.888454 < MY.NET.56.62 > 63.144.78.244: icmp: net MY.NET.56.205 unreachable [tos 0x10]
17:20:04.888457 < MY.NET.56.62 > 63.144.78.244: icmp: net MY.NET.56.206 unreachable [tos 0x10]
17:20:04.942840 < 63.144.78.244.1489 > MY.NET.56.94.domain: F 1:1(0) ack 1 win 32696 <nop,nop,timestamp 13633470 7077910> (DF) [tos 0x10]
17:20:04.943335 < MY.NET.56.94.domain > 63.144.78.244.1489: . 1:1(0) ack 2 win 16500 <nop,nop,timestamp 7077911 13633470>
17:20:04.943642 < MY.NET.56.94.domain > 63.144.78.244.1489: F 1:1(0) ack 2 win 16500 <nop,nop,timestamp 7077911 13633470>
17:20:05.184122 < 63.144.78.244.1603 > MY.NET.56.207.domain: S 1732638767:1732638767(0) win 32120 <mss 1460,sackOK,timestamp 13633495 0,nop,wscale 0> (DF)
[tos 0x10]
17:20:05.184497 < 63.144.78.244.1604 > MY.NET.56.208.domain: S 1722169865:1722169865(0) win 32120 <mss 1460,sackOK,timestamp 13633495 0,nop,wscale 0> (DF)
[tos 0x10]
17:20:05.184505 < MY.NET.56.62 > 63.144.78.244: icmp: net MY.NET.56.207 unreachable [tos 0x10]
```

Based on the above TCPDUMP data the attacker has:

A) Scanned the network for DNS Servers?
B) Successfully completed a TCP Three-way-handshake?
C) No attack is present, the data contains response packets.
D) A & B

D

Detect Number 3

Xmas-Tree scan.

Data Source: Snort, Portscan Log File

```
Apr 15 22:48:44 210.50.21.104:1034 -> MY.NET.18.33:80 SYN **S*****
Apr 15 22:48:44 210.50.21.104:18245 -> MY.NET.18.33:21536 XMAS 2**F*P*U RESERVEDBITS
Apr 15 22:48:44 210.50.21.104:1036 -> MY.NET.18.32:80 SYN **S*****
Apr 15 22:48:48 210.50.21.104:1038 -> MY.NET.18.25:80 SYN **S*****
Apr 15 22:49:30 210.50.21.104:1040 -> MY.NET.18.33:80 SYN **S*****
Apr 15 22:49:30 210.50.21.104:18245 -> MY.NET.18.33:21536 XMAS 2**F*P*U RESERVEDBITS
Apr 15 22:49:30 210.50.21.104:1041 -> MY.NET.18.32:80 SYN **S*****
Apr 15 22:49:30 210.50.21.104:18245 -> MY.NET.18.32:21536 NOACK 2*SFR*** RESERVEDBITS
Apr 15 22:50:40 210.50.21.104:1044 -> MY.NET.18.32:80 SYN **S*****
Apr 15 22:50:40 210.50.21.104:1045 -> MY.NET.18.33:80 SYN **S*****
Apr 15 22:50:40 210.50.21.104:18245 -> MY.NET.18.33:21536 XMAS 2**F*P*U RESERVEDBITS
Apr 15 22:50:44 210.50.21.104:1046 -> MY.NET.18.32:80 SYN **S*****
Apr 15 22:50:48 210.50.21.104:1047 -> MY.NET.18.32:80 SYN **S*****
Apr 15 22:50:48 210.50.21.104:18245 -> MY.NET.18.32:21536 XMAS 2**F*P*U RESERVEDBITS
Apr 15 22:50:52 210.50.21.104:1050 -> MY.NET.18.25:80 SYN **S*****
```

Data Source: Syslog: ipflog

```
Apr 15 22:48:45 MYHOST ipmon[4322]: 22:48:45.202907 dc0 @0:3 b 210.50.21.104,18245 -> MY.NET.18.32,21536 PR tcp len 20 274 -RSFP IN
Apr 15 22:48:49 MYHOST ipmon[4322]: 22:48:48.798246 dc0 @0:3 b 210.50.21.104,18245 -> MY.NET.18.25,21536 PR tcp len 20 277 -RSFP IN
Apr 15 22:49:30 MYHOST ipmon[4322]: 22:49:30.693549 dc0 @0:3 b 210.50.21.104,18245 -> MY.NET.18.32,21536 PR tcp len 20 328 -RSF IN
Apr 15 22:50:41 MYHOST ipmon[4322]: 22:50:40.610792 dc0 @0:3 b 210.50.21.104,18245 -> MY.NET.18.32,21536 PR tcp len 20 274 -RSFP IN
```

Data Source: TCPDUMP

```
22:48:40.333236 0:d0:ff:e6:2c:1c 0:a0:cc:50:d0:e7 0800 361: 210.50.21.104.18245 > MY.NET.18.33.21536: F [bad tcp cksum 15b6!] 790644820:790645139(319) ack
1414541105 win 3338 urg 25445 (DF) (ttl 116, id 61440)
0000: 4500 015b f000 4000 7406 49d8 d232 1568 E..[..@.t.I..2.h
0010: d208 1221 4745 5420 2f20 4854 5450 2f31 ...!GET / HTTP/1
0020: 2e31 0d0a 4163 6365 7074 3a20 6170 706c .1..Accept: appl
0030: 6963 6174 696f 6e2f 766e 642e 6d73 2d65 ication/vnd.ms-e
0040: 7863 656c 2c20 6170 706c 6963 6174 696f xcel, applicatio
0050: 6e2f 6d73 776f 7264 2c20 6170 706c 6963 n/msword, applic
0060: 6174 696f 6e2f 766e 642e 6d73 2d70 6f77 ation/vnd.ms-pow
0070: 6572 706f 696e 742c 2069 6d61 6765 2f67 erpoint, image/g
0080: 6966 2c20 696d 6167 652f 782d 7862 6974 if, image/x-xbit
0090: 6d61 702c 2069 6d61 6765 2f6a 7065 672c map, image/jpeg,
00a0: 2069 6d61 6765 2f70 6a70 6567 2c20 2a2f image/pjpeg, */
00b0: 2a0d 0a41 6363 6570 742d 4c61 6e67 7561 *..Accept-Langua
00c0: 6765 3a20 656e 2d61 750d 0a41 6363 6570 ge: en-au..Accep
00d0: 742d 456e 636f 6469 6e67 3a20 677a 6970 t-Encoding: gzip
00e0: 2c20 6465 666c 6174 650d 0a55 7365 722d , deflate..User-
00f0: 4167 656e 743a 204d 6f7a 696c 6c61 2f34 Agent: Mozilla/4
0100: 2e30 2028 636f 6d70 6174 6962 6c65 3b20 .0 (compatible;
0110: 4d53 4945 2035 2e30 313b 2057 696e 646f MSIE 5.01; Windo
<SANITISED>

22:48:44.820729 0:d0:ff:e6:2c:1c 0:a0:cc:50:d0:e7 0800 286: 210.50.21.104.18245 > MY.NET.18.33.21536: FPE [bad tcp cksum 6a2f!] 796095609:796095833(224) win
28275 urg 27749 <[bad opt]> (DF) (ttl 116, id 4609)
0000: 4500 0110 1201 4000 7406 2823 d232 1568 E.....@.t.(#.2.h
0010: d208 1221 4745 5420 2f73 7479 6c65 732f ...!GET /styles/
0020: 7769 6e73 7479 6c65 2e63 7373 2048 5454 winstyle.css HTT
0030: 502f 312e 310d 0a41 6363 6570 743a 202a P/1.1..Accept: *
0040: 2f2a 0d0a 5265 6665 7265 723a 2068 7474 /*..Referer: htt
<SANITISED>

22:48:45.202768 0:d0:ff:e6:2c:1c 0:a0:cc:50:d0:e7 0800 288: 210.50.21.104.18245 > MY.NET.18.32.21536: SFRPE [bad tcp cksum 9c49!] 794255716:794255954(238)
win 16968 (DF) (ttl 116, id 7425)
0000: 4500 0112 1d01 4000 7406 1d22 d232 1568 E.....@.t..".2.h
0010: d208 1220 4745 5420 2f57 6164 3f77 3d47 ... GET /Wad?w=G
0020: 4c4f 4248 4f4d 4550 4147 4520 4854 5450 LOBHOMEPAGE HTTP
0030: 2f31 2e31 0d0a 4163 6365 7074 3a20 2a2f /1.1..Accept: */
0040: 2a0d 0a52 6566 6572 6572 3a20 6874 7470 *..Referer: http
<SANITISED>

22:48:48.798108 0:d0:ff:e6:2c:1c 0:a0:cc:50:d0:e7 0800 291: 210.50.21.104.18245 > MY.NET.18.25.21536: SFRPE [bad tcp cksum 1570!] 794255716:794255957(241)
win 16968 (DF) (ttl 116, id 17921)
0000: 4500 0115 4601 4000 7406 f425 d232 1568 E...F.@.t..%.2.h
0010: d208 1219 4745 5420 2f57 6164 3f77 3d47 ....GET /Wad?w=G
0020: 4c4f 4248 4f4d 4550 4147 4520 4854 5450 LOBHOMEPAGE HTTP
0030: 2f31 2e31 0d0a 4163 6365 7074 3a20 2a2f /1.1..Accept: */
<SANITISED>
```

```
22:49:25.371039 0:d0:ff:e6:2c:1c 0:a0:cc:50:d0:e7 0800 439: 210.50.21.104.18245 > MY.NET.18.33.21536: FRP [bad tcp cksum 5dfb!] 5711641:5712030(389) ack
236012815 win 29557 urg 11120 (DF) (ttl 116, id 35329)
0000: 4500 01a9 8a01 4000 7406 af89 d232 1568 E.....@.t....2.h
0010: d208 1221 4745 5420 2f77 6f6d 6261 743f ...!GET /wombat?
0020: 493d 7375 6573 2b70 6f6f 6c26 5365 6172 I=sues+pool&Sear
0030: 6368 2e78 3d31 3826 5365 6172 6368 2e79 ch.x=18&Search.y
0040: 3d31 3120 4854 5450 2f31 2e31 0d0a 4163 =11 HTTP/1.1..Ac
0050: 6365 7074 3a20 6170 706c 6963 6174 696f cept: applicatio
0060: 6e2f 766e 642e 6d73 2d65 7863 656c 2c20 n/vnd.ms-excel,
0070: 6170 706c 6963 6174 696f 6e2f 6d73 776f application/mswo
0080: 7264 2c20 6170 706c 6963 6174 696f 6e2f rd, application/
0090: 766e 642e 6d73 2d70 6f77 6572 706f 696e vnd.ms-powerpoin
00a0: 742c 2069 6d61 6765 2f67 6966 2c20 696d t, image/gif, im
00b0: 6167 652f 782d 7862 6974 6d61 702c 2069 age/x-xbitmap, i
00c0: 6d61 6765 2f6a 7065 672c 2069 6d61 6765 mage/jpeg, image
00d0: 2f70 6a70 6567 2c20 2a2f 2a0d 0a52 6566 /pjpeg, */*..Ref
<SANITISED>

22:49:30.619193 0:d0:ff:e6:2c:1c 0:a0:cc:50:d0:e7 0800 416: 210.50.21.104.18245 > MY.NET.18.33.21536: FPE [bad tcp cksum b22a!] 796095609:796095963(354) win
28275 urg 27749 <[bad opt]> (DF) (ttl 116, id 40705)
0000: 4500 0192 9f01 4000 7406 9aa0 d232 1568 E.....@.t....2.h
0010: d208 1221 4745 5420 2f73 7479 6c65 732f ...!GET /styles/
0020: 7769 6e73 7479 6c65 2e63 7373 2048 5454 winstyle.css HTT
0030: 502f 312e 310d 0a41 6363 6570 743a 202a P/1.1..Accept: *
0040: 2f2a 0d0a 5265 6665 7265 723a 2068 7474 /*..Referer: htt
<SANITISED>

22:49:30.693396 0:d0:ff:e6:2c:1c 0:a0:cc:50:d0:e7 0800 342: 210.50.21.104.18245 > MY.NET.18.32.21536: SFRE [bad tcp cksum a910!] 794255716:794256004(288)
win 19539 (DF) (ttl 116, id 41473)
0000: 4500 0148 a201 4000 7406 97eb d232 1568 E..H..@.t....2.h
0010: d208 1220 4745 5420 2f57 6164 3f77 3d57 ... GET /Wad?w=W
0020: 5747 4c53 4541 5243 4826 743d 7375 6573 WGLSEARCH&t=sues
0030: 2532 3070 6f6f 6c20 4854 5450 2f31 2e31 %20pool HTTP/1.1
0040: 0d0a 4163 6365 7074 3a20 2a2f 2a0d 0a52 ..Accept: */*..R
0050: 6566 6572 6572 3a20 6874 7470 3a2f 2f77 eferer: http://w
<SANITISED>

22:50:40.610649 0:d0:ff:e6:2c:1c 0:a0:cc:50:d0:e7 0800 288: 210.50.21.104.18245 > MY.NET.18.32.21536: SFRPE [bad tcp cksum 9c49!] 794255716:794255954(238)
win 16968 (DF) (ttl 116, id 54529)
0000: 4500 0112 d501 4000 7406 6521 d232 1568 E.....@.t.e!.2.h
0010: d208 1220 4745 5420 2f57 6164 3f77 3d47 ... GET /Wad?w=G
0020: 4c4f 4248 4f4d 4550 4147 4520 4854 5450 LOBHOMEPAGE HTTP
0030: 2f31 2e31 0d0a 4163 6365 7074 3a20 2a2f /1.1..Accept: */
0040: 2a0d 0a52 6566 6572 6572 3a20 6874 7470 *..Referer: http
<SANITISED>

22:50:40.721481 0:d0:ff:e6:2c:1c 0:a0:cc:50:d0:e7 0800 282: 210.50.21.104.18245 > MY.NET.18.33.21536: FPE [bad tcp cksum 3220!] 794981740:794981964(224) win
28276 urg 26982 <[bad opt]> (DF) (ttl 116, id 55297)
0000: 4500 010c d801 4000 7406 6226 d232 1568 E.....@.t.b&.2.h
0010: d208 1221 4745 5420 2f62 756c 6c65 7470 ...!GET /bulletp
0020: 6f69 6e74 2e67 6966 2048 5454 502f 312e oint.gif HTTP/1.
0030: 310d 0a41 6363 6570 743a 202a 2f2a 0d0a 1..Accept: */*..
0040: 5265 6665 7265 723a 2068 7474 703a 2f2f Referer: http://
<SANITISED>

22:50:44.765964 0:d0:ff:e6:2c:1c 0:a0:cc:50:d0:e7 0800 400: 210.50.21.104.18245 > MY.NET.18.32.21536: F [bad tcp cksum 6e9a!] 790644820:790645178(358) ack
1414541105 win 3338 urg 25445 (DF) (ttl 116, id 58113)
0000: 4500 0182 e301 4000 7406 56b1 d232 1568 E.....@.t.V..2.h
0010: d208 1220 4745 5420 2f20 4854 5450 2f31 ... GET / HTTP/1
0020: 2e31 0d0a 4163 6365 7074 3a20 6170 706c .1..Accept: appl
0030: 6963 6174 696f 6e2f 766e 642e 6d73 2d65 ication/vnd.ms-e
0040: 7863 656c 2c20 6170 706c 6963 6174 696f xcel, applicatio
0050: 6e2f 6d73 776f 7264 2c20 6170 706c 6963 n/msword, applic
0060: 6174 696f 6e2f 766e 642e 6d73 2d70 6f77 ation/vnd.ms-pow
0070: 6572 706f 696e 742c 2069 6d61 6765 2f67 erpoint, image/g
0080: 6966 2c20 696d 6167 652f 782d 7862 6974 if, image/x-xbit
0090: 6d61 702c 2069 6d61 6765 2f6a 7065 672c map, image/jpeg,
00a0: 2069 6d61 6765 2f70 6a70 6567 2c20 2a2f image/pjpeg, */
00b0: 2a0d 0a52 6566 6572 6572 3a20 6874 7470 *..Referer: http
<SANITISED>



22:50:48.987475 0:d0:ff:e6:2c:1c 0:a0:cc:50:d0:e7 0800 292: 210.50.21.104.18245 > MY.NET.18.32.21536: FPE [bad tcp cksum fd8c!] 796095609:796095839(230) win
28275 urg 27749 <[bad opt]> (DF) (ttl 116, id 63489)
0000: 4500 0116 f801 4000 7406 421d d232 1568 E.....@.t.B..2.h
0010: d208 1220 4745 5420 2f73 7479 6c65 732f ... GET /styles/
0020: 7769 6e73 7479 6c65 2e63 7373 2048 5454 winstyle.css HTT
0030: 502f 312e 310d 0a41 6363 6570 743a 202a P/1.1..Accept: *
0040: 2f2a 0d0a 5265 6665 7265 723a 2068 7474 /*..Referer: htt
<SANITISED>
```

1. Source of Trace:
   This is a customer firewall appliance running OpenBSD 2.7, IPFilter and Snort

2. Detect was generated by:
   Three detects are present:
   - Detect 1: Snort provided initial identification of attack.
   - Detect 2: IPFilter confirmed data in Detect 1.
   - Detect 3: TCPDUMP provided all details of packets as they arrived, including full payloads.

3. Probability the source address was spoofed:
   Very Low.
   It is very unlikely that these packets were spoofed as it appears that this data is being generated by equipment on the Internet with poor stack handling capabilities. This theory is based on some discussions on SecurityFocus.com and supported by the TCP packet payloads.

4. Description of attack:
   Whilst Snort reports this as a Xmas tree scan it does not in fact appear to be an attack. It appears more likely that these packets actually represent a device on the Internet with a stack that is broken in some manner.

5. Attack Mechanism:
   As the payload of the TCP packets contains many references to WWW server content local to MY.NET it appears that this is not an attack. It is appears more likely that these packets are caused by the source address browsing content at the local MY.NET site and their device, with some poor code in its stack, issuing the mal-formed packets.
   See: http://archives.neohapsis.com/archives/incidents/2001-01/0079.html
   http://www.securityfocus.com/frames/?content=/templates/archive.pike%3Flist%3D75%26mid%3D170861

6. Correlation:
   This probe has been seen by many others and current discussions trend towards an issue with a Nortel device having issues with its IP Stack.

7. Evidence of active targeting:
   Strictly speaking this is an actively targeted probe, as the perpetrator deliberately target the machines to which these malformed packets were sent. However as this is unlikely to be an attack there was no malicious targeting involved.

8. Severity:
   (4 + 1) - (2 + 5) = -1
   - The systems probed were web servers for a major content provider (4).
   - Given that the packets were likely malformed it would be unlucky to have them cause major issues. (1)
   - The web servers themselves have little to defend them against malformed packets exploiting some vulnerability. (2)
   - The Firewall was able to intercept these packets and deny them access to the web servers (5)

9. Defensive recommendations:
   No actions are required immediately as the firewall provided adequate protection however it would be a good idea to ensure the web servers are patched to current releases limiting the number of known vulnerabilities that may be used to exploit them.

10. Multiple Choice Question:

```
22:50:40.610649 0:d0:ff:e6:2c:1c 0:a0:cc:50:d0:e7 0800 288: 210.50.21.104.18245 > MY.NET.18.32.21536: SFRPE [bad tcp cksum 9c49!] 794255716:794255954(238)
win 16968 (DF) (ttl 116, id 54529)
0000: 4500 0112 d501 4000 7406 6521 d232 1568 E.....@.t.e!.2.h
0010: d208 1220 4745 5420 2f57 6164 3f77 3d47 ... GET /Wad?w=G
0020: 4c4f 4248 4f4d 4550 4147 4520 4854 5450 LOBHOMEPAGE HTTP
0030: 2f31 2e31 0d0a 4163 6365 7074 3a20 2a2f /1.1..Accept: */
0040: 2a0d 0a52 6566 6572 6572 3a20 6874 7470 *..Referer: http
<SANITISED>

22:50:40.721481 0:d0:ff:e6:2c:1c 0:a0:cc:50:d0:e7 0800 282: 210.50.21.104.18245 > MY.NET.18.33.21536: FPE [bad tcp cksum 3220!] 794981740:794981964(224) win
28276 urg 26982 <[bad opt]> (DF) (ttl 116, id 55297)
0000: 4500 010c d801 4000 7406 6226 d232 1568 E.....@.t.b&.2.h
0010: d208 1221 4745 5420 2f62 756c 6c65 7470 ...!GET /bulletp
0020: 6f69 6e74 2e67 6966 2048 5454 502f 312e oint.gif HTTP/1.
0030: 310d 0a41 6363 6570 743a 202a 2f2a 0d0a 1..Accept: */*..
0040: 5265 6665 7265 723a 2068 7474 703a 2f2f Referer: http://
<SANITISED>

22:50:44.765964 0:d0:ff:e6:2c:1c 0:a0:cc:50:d0:e7 0800 400: 210.50.21.104.18245 > MY.NET.18.32.21536: F [bad tcp cksum 6e9a!] 790644820:790645178(358) ack
1414541105 win 3338 urg 25445 (DF) (ttl 116, id 58113)
0000: 4500 0182 e301 4000 7406 56b1 d232 1568 E.....@.t.V..2.h
0010: d208 1220 4745 5420 2f20 4854 5450 2f31 ... GET / HTTP/1
0020: 2e31 0d0a 4163 6365 7074 3a20 6170 706c .1..Accept: appl
0030: 6963 6174 696f 6e2f 766e 642e 6d73 2d65 ication/vnd.ms-e
0040: 7863 656c 2c20 6170 706c 6963 6174 696f xcel, applicatio
0050: 6e2f 6d73 776f 7264 2c20 6170 706c 6963 n/msword, applic
0060: 6174 696f 6e2f 766e 642e 6d73 2d70 6f77 ation/vnd.ms-pow
0070: 6572 706f 696e 742c 2069 6d61 6765 2f67 erpoint, image/g
0080: 6966 2c20 696d 6167 652f 782d 7862 6974 if, image/x-xbit
0090: 6d61 702c 2069 6d61 6765 2f6a 7065 672c map, image/jpeg,
00a0: 2069 6d61 6765 2f70 6a70 6567 2c20 2a2f image/pjpeg, */
```

   Based on the above data:

   A) The network is being probed for trojans.
   B) The network is being probed for vulnerable web servers.
   C) No attack is present.
   D) A Buffer-Over flow is taking place against a server listening on port 21356.

   C

Detect Number 4

Denial of Service attack against DNS server.

Data Source: Snort

Snort Alert:

```
[**] spp_portscan: PORTSCAN DETECTED from 211.10.173.241 (THRESHOLD 10 connections exceeded in 0 seconds) [**]
04/27-09:01:07.885104

[**] IDS278 - SCAN -named Version probe [**]
04/27-09:01:07.949443 0:D0:FF:E6:2C:1C -> 0:A0:CC:50:D0:E7 type:0x800 len:0x48
211.10.173.241:2212 -> MY.DMZ.NET.10:53 UDP TTL:47 TOS:0x10 ID:22325 Len: 38

[**] IDS277 - NAMED Iquery Probe [**]
04/27-09:01:08.238513 0:D0:FF:E6:2C:1C -> 0:A0:CC:50:D0:E7 type:0x800 len:0x1FB
211.10.173.241:2212 -> MY.DMZ.NET.10:53 UDP TTL:47 TOS:0x10 ID:22334 Len: 473

[**] IDS278 - SCAN -named Version probe [**]
```

```
04/27-09:01:09.389959 0:D0:FF:E6:2C:1C -> 0:A0:CC:50:D0:E7 type:0x800 len:0x48
211.10.173.241:2212 -> MY.DMZ.NET.133:53 UDP TTL:47 TOS:0x10 ID:22841 Len: 38

[**] spp_portscan: portscan status from 211.10.173.241: 169 connections across 167 hosts: TCP(167), UDP(2) [**]
04/27-09:01:11.463390

[**] IDS278 - SCAN -named Version probe [**]
04/27-09:01:12.549956 0:D0:FF:E6:2C:1C -> 0:A0:CC:50:D0:E7 type:0x800 len:0x48
211.10.173.241:2214 -> MY.DMZ.NET.132:53 UDP TTL:47 TOS:0x10 ID:24805 Len: 38

[**] spp_portscan: portscan status from 211.10.173.241: 89 connections across 88 hosts: TCP(88), UDP(1) [**]
04/27-09:01:15.277358

[**] spp_portscan: End of portscan from 211.10.173.241: TOTAL time(6s) hosts(254) TCP(255) UDP(3) [**]
```

Snort snort-0427@0000.log:

```
09:01:07.949443 211.10.173.241.2212 > MY.DMZ.NET.10.53: 25126 TXT CHAOS)? version.bind. (30) [tos 0x10]

09:01:08.238513 211.10.173.241.2212 > MY.DMZ.NET.10.53: 25126 inv_q+ [b2&3=0x980] A?
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB.CCCCCCCCCCC
!"#$%&'()*+,-
./0123456789:;<=.EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE.FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
(465) [tos 0x10]

09:01:09.389959 211.10.173.241.2212 > MY.DMZ.NET.133.53: 25128 TXT CHAOS)? version.bind. (30) [tos 0x10]

09:01:12.549956 211.10.173.241.2214 > MY.DMZ.NET.132.53: 25148 TXT CHAOS)? version.bind. (30) [tos 0x10]
```

1. Source of Trace:

   This trace was detected on a customer firewall appliance. This customer is a significant provider of Internet search engine technology and as such is an extremely attractive target for attackers.
   This customers firewall is an OpenBSD 2.7 based appliance running Snort and SnortSnarf.

2. Detect generated by:

   Trace 1: Snort Alert file (see www.snort.org)
   Trace 2: TCPDUMP (see www.tcpdump.org)

3. Likelihood of spoofed source:

   Given that this detect started as a portscan probing the customers network for DNS servers and upon finding one the perpetrator initiated a version query it is very unlikely that the source addressed was spoofed.

4. Description of the attack:

   CVE-1999-0010

   The attacker initially scans for DNS servers that he/she may attack. Upon finding a DNS server the attacker determines the version by probing the server.
   For this detect the attacker has found a copy of BIND running on the server, but the version does not respond in the normal method. This is due to the administrator modifying the BIND configuration to respond in a non-standard manner.
   The attacker then appears to telnet to a chargen server with a source address of the victim DNS server. It is likely that the chargen server used is on their localhost as the source address does not change for packets arriving at the server.
   It has been concluded that chargen was involved as the pattern of packets is the same as those received by chargen.

5. Attack mechanism:

   The perpetrator of this attack may have used entirely pre-packaged tools, or may have used some automation and some manual work based on their findings.

   It is likely that the attacker has used a port scanning tool to locate DNS servers. Upon finding a server the attacker may have executed the version query by hand (i.e. By using nslookup to query the BIND server) or used a tool.

   Once the attacker received version information that did not conform to expectations initial thoughts were that the attacker may have executed the buffer overflow by hand.

   Looking at the pattern of characters used in the buffer overflow it can be confirmed that the likely final attack method was by hand. Comparing the pattern of characters sent to the DNS server to those produced by chargen shows that they are remarkably similar. Hence it was concluded that the attacker pointed the contents of a chargen server at the victim host. Against a BIND 8 DNS server this would likely result in a Denial of Service.

6. Correlation:

   Investigation of correlation of this attack using the SANS Consensus Intrusion Database it can be seen that this attacker was very active.

   Also correlation's exist against probes from this IP address at:
   http://sailor.netlinks.co.jp/snort/211/10/173/src211.10.173.241.html

7. Evidence of Targeting:

   No evidence of active targeting exists for the initial probes. However once the attackers tools alerted the perpetrator to the presence of a likely DNS server the attacker focused purely on that server.

8. Severity:

   (5 + 5) - (5 + 4) = 1
   - The systems being probed for were DNS servers. The one located is responsible as the authoritative

server for the customers domain names. (5)
- If this attack were to be successful it could cause untold damage to the customers Internet presence. Given that this company is a true "dot-com" company it is entirely reliant on its DNS system functioning correctly. Also if the attack were to gain root privileges the attacker could use this platform as a launch pad against other devices.
- The copy of BIND on this DNS server was the latest release, with all patches applied. The customization of version information also made it much more challenging for the attacker to determine the value of the target.
- The firewall defended against all of the probes to NON external DNS servers. However as this target was an authoritative server it required that traffic be allowed through the firewall. The firewall would have assisted in the prevention of using this appliance to attack other devices if the attack were successful.

9. Defensive recommendation:

   Given the importance of this device it is advisable that all patches be maintained for the version of BIND installed on this DNS server.

   It would also be advisable to install a dedicated IDS device to better monitor attacks against such critical systems.

10. Multiple Choice Question:

```
09:01:07.949443 211.10.173.241.2212 > MY.DMZ.NET.10.53: 25126 TXT CHAOS)? version.bind. (30) [tos 0x10]

09:01:08.238513 211.10.173.241.2212 > MY.DMZ.NET.10.53: 25126 inv_q+ [b2&3=0x980] A?
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB.CCCCCCCCCCCC
!"#$%&'()*+,-
./0123456789:;<=.EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE.FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
(465) [tos 0x10]

09:01:09.389959 211.10.173.241.2212 > MY.DMZ.NET.133.53: 25128 TXT CHAOS)? version.bind. (30) [tos 0x10]

09:01:12.549956 211.10.173.241.2214 > MY.DMZ.NET.132.53: 25148 TXT CHAOS)? version.bind. (30) [tos 0x10]
```

   Based on the above Snort TcpDump records we can assume:

   A) Nothing of interest took place.
   B) An attacker was looking for vulnerable BIND servers.
   C) Snort was mis-reading packets and produced a false-positive.
   D) An attacker found a version of BIND and attempted to DoS it.

   Select the most likely answer.

   Answer: D

Detect Number 5

Probe for NetBus trojan

SnortSnarf Summary report:

```
13 such alerts.
* 1 different signatures are present for 24.201.54.163 as a source

* 13 instances of TCP **S***** scan

There are 13 distinct destination IPs in the alerts of the type on this page.

24.201.54.163 (modemcable163.54-201-24.que.mc.videotron.ca)

Jun 1 09:27:26 24.201.54.163:1960-> MY.LOCAL.NET.226:12345 SYN **S*****
Jun 1 09:27:27 24.201.54.163:1962-> MY.LOCAL.NET.228:12345 SYN **S*****
Jun 1 09:27:27 24.201.54.163:1964-> MY.LOCAL.NET.230:12345 SYN **S*****
Jun 1 09:27:27 24.201.54.163:1963-> MY.LOCAL.NET.229:12345 SYN **S*****
Jun 1 09:27:27 24.201.54.163:1965-> MY.LOCAL.NET.231:12345 SYN **S*****
Jun 1 09:27:27 24.201.54.163:1966-> MY.LOCAL.NET.232:12345 SYN **S*****
Jun 1 09:27:27 24.201.54.163:1968-> MY.LOCAL.NET.234:12345 SYN **S*****
Jun 1 09:27:27 24.201.54.163:1967-> MY.LOCAL.NET.233:12345 SYN **S*****
Jun 1 09:27:27 24.201.54.163:1969-> MY.LOCAL.NET.235:12345 SYN **S*****
Jun 1 09:27:28 24.201.54.163:1970-> MY.LOCAL.NET.236:12345 SYN **S*****
Jun 1 09:27:28 24.201.54.163:1971-> MY.LOCAL.NET.237:12345 SYN **S*****
Jun 1 09:27:29 24.201.54.163:1972-> MY.LOCAL.NET.238:12345 SYN **S*****
Jun 1 09:27:29 24.201.54.163:1973-> MY.LOCAL.NET.239:12345 SYN **S*****
```

Snort Alerts:

```
[**] spp_portscan: PORTSCAN DETECTED from 24.201.54.163 (THRESHOLD 10 connections exceeded in 2 seconds) [**]
06/01-09:27:28.144879
[**] spp_portscan: portscan status from 24.201.54.163: 13 connections across 13 hosts: TCP(13), UDP(0) [**]
06/01-09:27:42.389967
[**] spp_portscan: End of portscan from 24.201.54.163: TOTAL time(3s) hosts(13) TCP(13) UDP(0) [**]
06/01-09:27:57.740482
```

Snort Portscan Detects:

```
Jun 1 09:27:26 24.201.54.163:1960 -> MY.LOCAL.NET.226:12345 SYN **S*****
Jun 1 09:27:27 24.201.54.163:1962 -> MY.LOCAL.NET.228:12345 SYN **S*****
Jun 1 09:27:27 24.201.54.163:1964 -> MY.LOCAL.NET.230:12345 SYN **S*****
Jun 1 09:27:27 24.201.54.163:1963 -> MY.LOCAL.NET.229:12345 SYN **S*****
Jun 1 09:27:27 24.201.54.163:1965 -> MY.LOCAL.NET.231:12345 SYN **S*****
Jun 1 09:27:27 24.201.54.163:1966 -> MY.LOCAL.NET.232:12345 SYN **S*****
Jun 1 09:27:27 24.201.54.163:1968 -> MY.LOCAL.NET.234:12345 SYN **S*****
Jun 1 09:27:27 24.201.54.163:1967 -> MY.LOCAL.NET.233:12345 SYN **S*****
Jun 1 09:27:27 24.201.54.163:1969 -> MY.LOCAL.NET.235:12345 SYN **S*****
Jun 1 09:27:28 24.201.54.163:1970 -> MY.LOCAL.NET.236:12345 SYN **S*****
Jun 1 09:27:28 24.201.54.163:1971 -> MY.LOCAL.NET.237:12345 SYN **S*****
Jun 1 09:27:29 24.201.54.163:1972 -> MY.LOCAL.NET.238:12345 SYN **S*****
Jun 1 09:27:29 24.201.54.163:1973 -> MY.LOCAL.NET.239:12345 SYN **S*****
```

IPFilter/SYSLOG:

```
Jun 1 09:27:27 02e ipmon[611]: 09:27:26.959161 de0 @0:15 b 24.201.54.163,1960 -> MY.LOCAL.NET.226,12345 PR tcp len 20 48 -S IN
Jun 1 09:27:28 02e ipmon[611]: 09:27:27.953854 de0 @0:15 b 24.201.54.163,1962 -> MY.LOCAL.NET.228,12345 PR tcp len 20 48 -S IN
Jun 1 09:27:28 02e ipmon[611]: 09:27:27.969067 de0 @0:15 b 24.201.54.163,1964 -> MY.LOCAL.NET.230,12345 PR tcp len 20 48 -S IN
Jun 1 09:27:28 02e ipmon[611]: 09:27:27.969301 de0 @0:15 b 24.201.54.163,1963 -> MY.LOCAL.NET.229,12345 PR tcp len 20 48 -S IN
Jun 1 09:27:28 02e ipmon[611]: 09:27:27.973397 de0 @0:15 b 24.201.54.163,1965 -> MY.LOCAL.NET.231,12345 PR tcp len 20 48 -S IN
Jun 1 09:27:28 02e ipmon[611]: 09:27:27.985871 de0 @0:15 b 24.201.54.163,1966 -> MY.LOCAL.NET.232,12345 PR tcp len 20 48 -S IN
Jun 1 09:27:28 02e ipmon[611]: 09:27:27.990231 de0 @0:15 b 24.201.54.163,1968 -> MY.LOCAL.NET.234,12345 PR tcp len 20 48 -S IN
Jun 1 09:27:28 02e ipmon[611]: 09:27:27.990743 de0 @0:15 b 24.201.54.163,1967 -> MY.LOCAL.NET.233,12345 PR tcp len 20 48 -S IN
Jun 1 09:27:28 02e ipmon[611]: 09:27:27.999079 de0 @0:15 b 24.201.54.163,1969 -> MY.LOCAL.NET.235,12345 PR tcp len 20 48 -S IN
Jun 1 09:27:28 02e ipmon[611]: 09:27:28.003407 de0 @0:15 b 24.201.54.163,1970 -> MY.LOCAL.NET.236,12345 PR tcp len 20 48 -S IN
Jun 1 09:27:28 02e ipmon[611]: 09:27:28.016091 de0 @0:15 b 24.201.54.163,1971 -> MY.LOCAL.NET.237,12345 PR tcp len 20 48 -S IN
Jun 1 09:27:29 02e ipmon[611]: 09:27:29.005026 de0 @0:15 b 24.201.54.163,1972 -> MY.LOCAL.NET.238,12345 PR tcp len 20 48 -S IN
Jun 1 09:27:29 02e ipmon[611]: 09:27:29.011419 de0 @0:15 b 24.201.54.163,1973 -> MY.LOCAL.NET.239,12345 PR tcp len 20 48 -S IN
```

1. Source of Trace:
   This trace was detected at a customer's firewall appliance running OpenBSD and Snort. Summary reports were generated by SnortSnarf from Silicon Defense.

2. Detect was generated by:

   - IPFilter/Syslog
   Format: (uninteresting data labeled <stuff>)
   <Date & Time><Hostname><Data Source><Unix Time><Interface Label><stuff><action><src addr>,<src prt>-><dst addr>,<dst prt>PR<protocol><protocol specific details>

   - Snort

3. Probability of spoofed address:
   Unlikely.
   It is unlikely the source address was spoofed. The source address resolves to a Cable Modem user and hence the likely hood of the device being connected to a MS Windows desktop is high. In the event that the perpetrator was using Windows (up till XP) the address can not be spoofed.

4. Description of attack:

This attack is likely to be a "Script-Kiddie" using a prepackaged tool to scan for devices on the Internet that contain a resident NetBus Trojan.
The attack vector for this type of attack requires that the destined machine have the NetBus server component installed and running. To achieve this a typical method is to rename the application to something likely to be executed, such as "addvertisment.avi.exe" and email it to someone with an email that has a subject line of "Have you seen this funny ad?" This will tempt the receiver of the message to click on the attachment, thus executing and installing the Netbus server application on to the users local machine.
Upon successfully probing for such victims the attacker can then take over the victim machine.

5. Attack mechanism:
   The attacker uses the NetBus Client application to scan for machines that have had the NetBus server application executed. The attacker then connects to this machine and takes control of the machine. The attacker is then free to use this machine for their own gain.

6. Correlation:
   This attacker was also detected across two other customer firewalls. The breadth of the target addresses suggests the attacker is scanning the entire address range of a major Australian upstream ISP.

7. Evidence of active targeting:
   No active targeting is evident in this detect. The attacker is scanning an entire network for vulnerable machines. Upon detection of a victim the client software being used will stop and establish communications with the victim host.

8. Severity:
   Severity = (Criticality+Lethality) - (System Countermeasures + Network Countermeasures)
   Severity = (5+5)-(1+5) = 4
   The targeted hosts are DMZ devices for a B2B ecommerce company, if the DMZ devices are taken over then the impact on the company could be terminal, hence a criticality of 4 and a Lethality of 5. It is unknown whether the targeted devices have any local defenses against such a scan, 1, but the network has a very good firewall appliance installed that will not allow the probe through, 5.

9. Defensive recommendations:
   It is highly recommended that the local administrator of the DMZ devices puts local defenses in place on each of the DMZ devices. Installation of anti-virus tools, Host based IDS and regular system audits will go a long way to preventing such an attack ever being successful.

10. Multiple choice test question:

    The probes:

```
Jun 1 09:27:28 02e ipmon[611]: 09:27:27.953854 de0 @0:15 b 24.201.54.163,1962 -> MY.LOCAL.NET.228,12345 PR tcp len 20 48 -S IN
Jun 1 09:27:28 02e ipmon[611]: 09:27:27.969067 de0 @0:15 b 24.201.54.163,1964 -> MY.LOCAL.NET.230,12345 PR tcp len 20 48 -S IN
Jun 1 09:27:28 02e ipmon[611]: 09:27:27.969301 de0 @0:15 b 24.201.54.163,1963 -> MY.LOCAL.NET.229,12345 PR tcp len 20 48 -S IN
Jun 1 09:27:28 02e ipmon[611]: 09:27:27.973397 de0 @0:15 b 24.201.54.163,1965 -> MY.LOCAL.NET.231,12345 PR tcp len 20 48 -S IN
```

    It can be stated:

    A) The attacker is on a Modem connection and is browsing the WWW at the same time.
    B) The attacker is on a CableModem connection and is FTPing data at the same time.
    C) The attacker is on a CableModem connection and is only performing the scan.
    D) The attacker is on a Modem connection and is only performing the scan.

    Pick the most likely.

    Answer: C.

Assignment 2 - Describe the State of Intrusion Detection

The importance of network forensics.

(Halt who probes there!)

Benjamin MA Robson

April 2001

It is the intention of this paper to describe the importance of forensic data capture tools as a front-line Information Technology security defense.

The date is April 2001, companies all over the world are being attacked by people unknown. Executives are faced with the prospect of facing a board of directors curious to find out why their web site was defaced, what is being done to prevent it from happening again and what can be done to prosecute those who instigated the attack. What can the executive tell them? Can he say that he didn't think they were large enough to be of interest, that he thought that the firewall was protecting them and that its the firewall vendors fault? Can he run away and hide, putting his companies head in the sand and pray to what ever higher being exists to help them?

The dilemma faced by any executive charged with a companies IT infrastructure is how to protect the information it holds from those who may wish to cause them harm. If they go to the market they are presented with a plethora of options, including Firewall's, Routers, Virtual LANs, Virtual Private Networks, Network Address Translation, Air-Gaps, Information compartmentalization, Intrusion Detection and many others. All of these can cost any where from very little to tens of thousands of dollars.

But even if the executive implements all of these things there is a fundamental component of front-line IT defense infrastructure that is missing. That is the forensic data capture device. In the event of being attacked how does the company determine what happened, how it happened, how to prevent it happening again, and what they can do to the person who made victims of them.

As highlighted by Sharonm Gaudin in her case study "Case study of insider sabotage", it is also necessary to remember that many attacks against corporate LANs originate from inside the network. Thus if the only network defense is the company firewall and external IDS the attacker is already successful. Implementation of forensic data capture tools throughout the network will allow the detection of such internal activities and assist the company in any actions to be taken against the employee.

But what is the difference between an Intrusion Detection system and a forensic tool? An IDS tool is based on a set of signatures. A little like a wanted poster from the old days a signature based IDS passes every data stream against its 'rap' sheet and either passes it quietly or if it matches a signature sets of an alarm.

But the down side of these systems can be quickly seen when reading documents such as those found at the Computer Security Institute and their article "Five vendors answer some no-nonsense questions on IDS". The first question asks the vendors how long it is between updates to their signature sets. All, but NFR, state that there is at least a two week period between updates. With a forensic tool, even if an attack goes through undetected at the time of instigation, as soon as it is noticed all the details of the attacker have been noted.

Some of these systems also have limits on what they can detect, how quickly they can detected it and what they do with the information once they see something suspicious.

When reading documents on what to do in the event of an attack, such as Ron Gula's "How to Handle and Identify Network Probes", the key thing that is identified is the requirement for knowledge. For raw data that can accurately describe what happened.

According to the Webster dictionary the word 'forensic' means, 'relating to or dealing with the application of scientific knowledge to legal problems.' Thus a primary objective in collecting forensic data from an IT network environment is to provide information of a standard suitable for use in a court of law.

To do this we must select and implement tools that are going to be able to collect as much data as possible to a standard of use to legal proceedings.

When considering what is appropriate it is best not to attempt to re-invent the wheel by treating the issue as a new problem. Law enforcement agencies have been tackling this issue for years.

These agencies have found that one of the best tools in their arsenal when fighting crime is the surveillance camera. This is an all-seeing device that passively sits in an environment and collects very complete data of the happenings within the environment that it is watching. The forensic tools that should be placed in a networked environment should be very similar in their nature.

They should be passive in their techniques, that is they should sit quietly on a network remaining largely unnoticed by those they observe and should be difficult to attack themselves. This way when something occurs that shouldn't the data captured will be untainted and raw, and as such can be processed to produce the quality of data required for forensics.

To achieve this there are some very useful tools available. These include NetIQ's Security Manager (www.netiq.com), Niksun's NetVCR & NetDetector (www.niksun.com) and many others. These high level tools attempt to provide the user with enough information to diagnose what

happened. However the best forensic tools are those that are based on the basic principle of capturing a copy of everything that goes past it.

A very good solution, that is also very cost effective, is a UNIX based device, such as a PC running OpenBSD, and a copy of tcpdump. Such a platform is only as expensive as the hardware on which it runs, is highly secure itself and is capable of capturing copies of every packet that traverses the networks it is attached to.

Such a solution is very robust. By configuring the device to monitor several networks with network interfaces passively connected to each network and running a tcpdump on each interface recording to a file system. Due to the secure nature of the OS and the configuration of the network interfaces it poses quite a challenge for an attacker to attack the monitoring device and thus corrupt the data on it.

If the network to be monitored has numerous of these such devices scattered through it, for example one on the outside of the network and then devices through the different sub-networks within the companies environment, an excellent picture can be drawn when an incident occurs by correlating data points.

The final remaining question is what to do with all of this raw, untainted data when an incident occurs. Having deployed the monitoring devices, established a strict data capture and storage routine what do we do with the data when an incident occurs .

The key here is training, effective networking with peers within the security industry and good quality analysis tools, such as Snort (www.snort.org).

Having attended training and gained knowledge of forensic techniques the administrator can analyze the data that has been captured and determine what happened, when it happened, how it happened, what it happened to and how quickly it happened.

Because the technologies implemented to capture the data were of a robust nature and unlikely to have been corrupted it can be considered that the information that has been gathered is accurate. Then using the network of piers in the security industry it can be correlated against others captured data to see if this is a broad-ranging attack or whether the attack was a specifically targeted attack against the organization.

Having established the target it can a total picture of who, what, where, when and how can be created. From this the company can determine what needs to be done to prevent the attack from being used against them again and can make the business decision as to whether to pursue the incident further in a court of law.

Of course it must be stated that forensic tools are not only solution. But it is a very important part of a wider security strategy. Used in conjunction with perimeter security appliances providing compartmentalization of information and Intrusion Detection systems to tell the administrator when they should be looking an attacker is very unlikely to go un-noticed.

The key to effective network security is fundamentally the same as in the real, physical, world. If the perpetrator knows that it is likely they will be caught and prosecuted then it is more likely they will not proceed with the attack.

References

Five vendors answer some no-nonsense questions on IDS
Computer Security Institute.
http://www.gocsi.com/ques.htm

Case Study of Insider Sabotage
Computer Security Institute
Computer Security Journal
Sharon Gaudin, Vol XVI, Number 3, 2000
http://www.gocsi.com/insider.pdf

How to Handle and Identify Network Probes
Ron Gula, April 1999
http://www.securitywizards.com/papers/probes.html

Tough questions for IDS vendors
Computer Security Institute
http://www.gocsi.com/IDSques.htm

An Introduction To Intrusion Detection & Assessment
ICSA Labs, 4 January 2000
http://www.icsa.net/html/communities/ids/White%20paper/Intrusion1.pdf

Assignment 3 - "Analyze This" Scenario

The following report has been constructed to provide GIAC Enterprises with an informative outline to the activities seen by their network intrusion detection system between January 20th and March 12th 2001.

The report is divided into 4 areas. The first section provides a summary detects on the network.

Section 2 provides a more detailed analysis of each detect. It defines what each detect is, through the provision of a description, states any ramifications that may exist as a result of a detect, highlights any source or destinations that are of interest in relation to the test, and why they are of interest, and provides recommended actions to take, if appropriate.

The final section summarizes actions that should be taken to rectify issues that were found.

Summary Details:

Between January 20th and March 12th the following attack types were reported by Snort (www.snort.org):

Tiny Fragments - Possible Hostile Activity
TCP SMTP Source Port traffic
ICMP SRC and DST outside network
TCP SRC and DST outside network
SNMP public access
Watchlist 000220 IL-ISDNNET-990517
Queso fingerprint
Security 000516-1
Null scan!
Back Orifice
Russia Dynamo - SANS Flash 28-jul-00
SYN-FIN scan!
SMB Name Wildcard
STATDX UDP attack
WinGate 1080 Attempt
SITE EXEC - Possible wu-ftpd exploit - GIAC000623
External RPC call
Probable NMAP fingerprint attempt
SUNRPC highport access!
Possible RAMEN server activity
connect to 515 from inside
NMAP TCP ping!
Attempted Sun RPC high port access
Watchlist 000222 NET-NCFC
UDP SRC and DST outside network

Attack Details:

Tiny Fragments - Possible Hostile Activity

Description: In an attempt to bypass many perimeter security appliances attackers can fragment their data packets. This means that data can slip through firewall appliances, and by-pass IDS systems and not trigger alarms.

Ramification: If the attacker is successful in using this method of penetrating a networks perimeter then they can exploit internal vulnerabilities without the local administrators being aware of issues arising.

Interesting Sources of Attack: The following source addresses attempted to get fragmented packets through:

111.111.111.111 - Likely Spoofed address.
127.0.0.1 - Likely Spoofed address.
64.80.90.36 - CollegePark/KnightsCourt
212.89.165.5 - acn Altec Group (www.acn.gr)

Destinations of Attack: The following destinations, within the local network, were targeted:

MY.NET.228.10
MY.NET.1.10
MY.NET.1.8
MY.NET.206.254
MY.NET.160.109
MY.NET.206.58
MY.NET.98.117
MY.NET.97.231
MY.NET.223.42
MY.NET.205.242
MY.NET.98.119

Recommendations: It is recommended that security appliances being installed that are capable of handling fragmented packets, instead of ignoring them. Proxy based firewall appliances that are able to reconstruct fragments and analyze them prior to passing, or blocking, them are much better than packet filtering solutions. If a packet filtering solution is the only one available it should be considered whether or not to allow fragmented packets at all.

TCP SMTP Source Port traffic:

Description: The packets here appear to be response packets from outbound SMTP connections. The server located at 200.251.185.30 is listed as a mail server in MX records, and thus supports this theory.

Ramification: This is not an attack, as such no ramifications exist.

Interesting Sources: No interesting sources exist.

Interesting Destinations: No interesting destinations exist.

Recommendations: It would be recommended to modify this Snort detect rule to prevent such false alerts existing. It is possible for such packets to be malicious, but without a better method of capturing data, such as a forensics device, it is impossible to tell. This is due to the lack of packet content data.

ICMP SRC and DST outside network:

Description: This alert was generated as a result of a custom rule created by the administrator of the Snort sensor. Large numbers of these detects are in fact related to Multi-Cast and the bulk of the remainder are private IP addresses (RFC 1918) and as such do not route over the Internet. The final few detects that qualify for this rule appear to have occurred due to poor routing, alternate route traffic (i.e. Modems acting as back-doors) or incorrectly configured machines being placed in to the network.

Ramification: With the exception of the back-door traffic, it is unlikely that this traffic pose any risks to the local network. The apparent back-door traffic (that with public addresses for both source and destination) should be investigated, and tracked to its source. Such a security issue can allow a perpetrator to by-pass security tools.

Interesting Sources: The following source addresses were detected by the Snort sensor.

10.*.*.* - Private Address Range (RFC 1918)
140.120.*.* - Ministry of Education Comp' Cent' (Taiwan)
65.9.177.76 - cc559503-a.owml1.md.home.com
172.*.*.* - America OnLine.
128.249.*.* - Baylor College of Medicine
217.9.64.248 -Consorzio Nazionale Interuniversitario per le Telecomunicazioni

Interesting Destinations: The following destinations were detected by the Snort Sensor.

224.*.*.* - MultiCast Network
192.163.*.* - UNISYS
172.*.*.* - America Online
62.224.*.* - Deutsche Telekom AG
24.189.*.* - Cablevision Systems Corp
211.106.127.* - Korea Internet Information Service
146.145.238.* - Commercial Furniture Interiors
24.228.9.* - Cablevision Systems
61.75.17.* - KOREA TELECOM
4.*.*.* - BBN Planet
24.66.*.* - Shaw Fiberlink ltd
156.3.*.* - Los Angeles County Office of Education
193.113.0.* - British Telecommunications
208.48.50.* - Information View
209.143.81.* - Charm Net
206.242.181.* - ERS Call Center
210.149.128.* - TOKYU CABLE TELEVISION Co.,Ltd.

Recommendations: It is highly recommended that the location of the alternate routes into the network be located and removed.

TCP SRC and DST outside network

Description: These alerts are as a result of a custom rule installed in to the Snort rules file by the network intrusion detection device administrator. The detects are finding packets that appear to originate from either an alternate route in to the local network, or are a result of users on the network not following the acceptable use policy.

Ramifications: These detects to not appear to present a security issue to the local network. As such the main ramification that can be highlighted are that alternate routes can allow an attacker to by-pass all security measures in a network, and that other traffic from misbehaving network users may add to local congestion.

Interesting Sources: No sources highlight themselves as being of more interest (from a security perspective) than others. It may be worth tracking those people who are doing strange things on the network segment by MAC address.

Interesting Destinations: As per Sources.

Recommendations: The major recommendation is to track those users of the network whom are introducing an alternate route for traffic in to the network environment. A secondary suggestion would be to use the MAC address details to locate those users whom are not abiding by the organization acceptable use policy.

SNMP public access

Description: This data presents itself as having a range of options. It could be malicious in its nature, with perpetrators at Purdue University Engineering department and NASA beating on the SNMP servers, however this seems unlikely. It is more likely that these alerts are being caused by a missconfiguration that has not been a problem to date, or that remote monitoring of the SNMP servers is taking place from Purdue and NASA. The later of these is the better choice as the data has not change over an extended period of time.

Ramifications: In the likely case that remote monitoring is taking place no ramifications exist. In the event that this data is not as a result of prearranged conditions then it should be treated with extreme skepticism.

Interesting Sources: The following sources were of interest in this detect:

128.46.156.197 - Purdue University
128.183.38.30 - NASA Goddard Space Flight Center

The remaining source addresses are from the internal network and as such are unlikely be of concern.

Interesting Destinations: The destinations for these detects are likely to be network devices (such as printers) that have an SNMP agent providing status information.

Recommendations: It may be appropriate to modify this Snort rule as it is producing spurious alerts.

Watchlist 000220 IL-ISDNNET-9900517

Description: This detect is as a result of a custom rule installed by the IDS device administrator. From the data presented it appears that the rule is designed to capture all traffic originating from the 212.179.0.0/16 network.

Ramifications: This detect is to capture the fact that ISDN Net Ltd. Is not enforcing an acceptable use policy on their users and as such many probes are resulting from this network. Probes from this network should be treated with suspicion.

Interesting Sources: These detects provide alerts for traffic originating from the ISDN Net Ltd network.

Interesting Destinations: The destination ports are largely either Napster or Gnutella standard ports. As such a reasonable assumption is that the other ports are variations on the destination ports for the same services. As such it is likely that the destinations are not internal systems, and are perhaps systems being run remotely, for private use.

Recommendations: Recommendations are entirely dependent on the organizations acceptable use policy. If such file sharing applications as Gnutella and Napster are acceptable, and are not deemed to present a security risk, as per the organizations information security policy then no action is required. If they in fact violate either of these policies, action should be taken to deny services.

Queso fingerprint

Description: A Queso Fingerprint is a type of probe that uses a TCP packet with the reserved bits set and a Syn flag set. It is used to attempt to determine what operating system the target system is running. This can be a precursor to a more defined attack against a host.

Ramifications: If this fingerprinting scan against the internal network is successful the perpetrator will have specific operating system details of target hosts. Using this information the attacker could then craft a specific attack against the hosts.

Interesting Sources:

141.30.228.43, 141.30.228.222, 141.30.228.122, 141.30.228.165, 141.30.228.134 are all part of the Technische Universitaet Dresden network, and as such is likely to be student activity. The destination ports for these packets are 6346, the default Gnutella port, and are likely attempting to be disguising themselves as such.

207.96.122.8, owned by RCN Corporation, appears initially to be Active FTP data channel traffic, but it has the reserve bits set and as such are illegal.

209.85.60.183, owned by CraigK-SA, also shows up regularly as a source of illegally flagged packets. The traffic clearly shows that this is a Queso tool probing a specific host.

Interesting Destinations: Probes against the internal network appear to be diverse in nature. This would suggest that the perpetrators do not have a specific target in mind and are more focused on finding a vulnerable machine to exploit. The only exception to this is traffic originating from 209.85.60.183, which is very specifically targeting the machine at MY.NET.229.158. The probes are over an extended period of time, possibly in an attempt to not be detected.

Recommendations: There is an argument for black-listing the source addresses within the 141.30.228.0/24 network. Regular traffic is appearing with illegal flag settings. If the traffic from this network can not be justified it should be denied access to the network.

It would also be recommended to follow-up the probes from 209.85.60.183 with the IP address range owner as this traffic if highly suspicious and should be investigated more closely.

Security 000516-1

Description: The data captured for this alert is as a result of a custom rule created for Snort by the intrusion detection system administrator. Kit is unclear whether the administrator is scanning for users of Napster, or data going to or from 140.247.187.110.

Ramifications: No known ramifications exist. The result of this alert are entirely dependent on what the administrator is trying to achieve by capturing this data.

Interesting Sources:

140.247.187.110 - Harvard University.
It appears that this host is communicating with Napster server located at MY.NET.206.74

Interesting Destinations:

MY.NET.206.74 appears to be running a Napster server.

Recommendations: If the use of Napster is a violation of the acceptable use policy MY.NET.206.74 is in violation of this policy. As such the user of this host should be investigated.

Null Scan!

Description: A null scan is a TCP network scan that has no TCP flags set. The option of having no flags set is illegal for the TCP protocol and as such all traffic with this setting should be treated with suspicion. A null scan is typically used for fingerprinting network devices, by determining how a device reacts to the illegal flag settings, or it can be used to slip packets through a packet-filtering firewall that does not allow state, and is not configured correctly.

Ramifications: A Null scan is a possible precursor to a more concerted attack against specific targets. Once the perpetrator has identified a viable target they can use the information provided to select the best method of attack.

Interesting Sources: 118 different source addresses were detected performing Null scans. As such no source was more interesting than any other source.

Interesting Destinations: 90 different destinations were detected for the Null Scan. All destinations were within the local network and no host seemed greatly targeted over other hosts.

Recommendations: Deployment of perimeter security appliances capable of filtering packets with illegal flags would be highly recommended to prevent such packets getting in to the local network.

Back Orifice

Description: Back Orifice is an application designed to allow a remote user to gain control of a local MS Windows machine. The trojan server is installed on a local machine, often by tricking a local user to execute an e-mail attachment. Back Orifice client applications then search the Internet at large for devices that are listening to UDP port 31337. Back Orifice was produce by a hacker group known as the Cult of the Dead Cow.

Ramifications: The alerts to not indicate that there is any security concern here. Devices on the network were probed by 2 sources, but there is no information to suggest that a compromise took place.

Interesting Sources: Two source addresses probed the network for holders of the Back Orifice Trojan:

63.10.224.59 - UUNET Technologies, Inc.
203.170.152.87 - Unknown owner.

Interesting Destinations: No source addresses were of more interest than others.

Recommendations: To ensure defenses exist against this attack two possible strategies, or a combination of both, can be taken. The first strategy is to ensure that UDP, port 31337 is not allowed through the perimeter of the network and into the local network. The second strategy is to ensure that the Back Orifice trojan is not present on any devices within the local network.

Russia Dynamo - SANS Flash 28-jul-00

Description: No details on this can be found. The only details known of this detect is that it is likely in relation to a scam running from Russia involving payment for traffic directed to a WWW site.

Ramifications: Ramifications of this scan are unknown. It is suggested that the detect is treated as highly suspicious and the source device investigated.

Interesting Sources: The source for this detect is internal to the local network, MY.NET.203.50

Interesting Destinations: This detect indicates data transmitted to a device located in Russia, 194.87.6.79 - 79.6.87.194.dynamic.dol.ru

Recommendations: It is recommended that the host device within the local network be investigated for trojan's that have been installed and are running.

SYN-FIN scan!

Description: The SYN-FIN scan indicated here is a type of TCP scan that is sent with both the SYN and FIN flags set. This type of scan is used as a mapping tool to determine the layout of a target network.

Ramifications: This type of scan is often a precursor to an attack. It is used to determine if a target of opportunity exists. Of itself it is not of a major concern.

Interesting Sources: Some quite extensive scanning took place from only 9 sources of packets. Of particular note were:

130.234.184.112
211.248.112.67
128.61.136.233 - tann6233.mse.gatech.edu

Interesting Destinations: 10346 different destination addresses were scanned, none of these were probed more than 4 times. Most of these were uninteresting. The exception to this are the addresses:

MY.NET.219.22, MY.NET.130.81, and MY.NET.105.169. Each of these will be referenced later as they appear to have been exploited. The SYN-FIN scan could have been an initial probe.

Recommendations: It is recommended that a perimeter device capable of deny access to TCP packets with a SYN-FIN combination of flags is used.

SMB Name Wildcard

Description: This traffic looks primarily like NetBIOS names traffic. Windows uses this type of communications as part of its filesharing activities. An attacker may use this method, if it is permitted to pass perimeter security appliances, to gather machine names, domains and users logged in.

Ramifications: If this were to be permitted through in to the internal network the attacker could better understand the environment they are attacking, and as such customize the methods they use.

Interesting Sources:

165.230.77.89 - Was the most active, but only scanned 1 destination address.
141.219.84.58 - Was the next most active and scanned only 3 devices.

Interesting Destinations:

MY.NET.130.185 - was the most scanned device on the internal network. It was probed by only 1 address.

Recommendations: It is recommended that this traffic not be permitted through the perimeter security appliance unless it is absolutely required.

STATDX UDP attack

Description: This alert indicates that an attacker is attempting to exploit a vulnerability in rpc.statd services. Packets probing for this exploit are identifiable by having a data content of "/bin|c74604|/sh". This is an attempt to exploit the rpc.statd vulnerability to allow execution of code as root on the local device.

Ramifications: If this attack is successfully implemented it will allow the perpetrator to execute arbitrary code, as the root user, on the victim device.

Interesting Sources: Successful attacks are launched from:

171.65.61.201 - psych-3365-PC.Stanford.EDU
There is a very good likelihood that this device at Stanford University has been hacked. Looking at traffic originating from this IP address we see a range of STATDX attacks against the local network and probes for RPC (TCP/111).

This source appears to successfully exploit this vulnerability.

Interesting Destinations: The following devices appear to have been successfully exploited using the STATDX vulnerability:

MY.NET.105.169 -Exploited by Stanford Uni' device. Results in outbound RAMEN alerts.
MY.NET.105.91 - Exploited by Stanford Uni' device. Results in outbound RAMEN alerts.
MY.NET.181.127 - Exploited by Stanford Uni' device. Results in outbound RPC (TCP/111) SYN scan alerts.
MY.NET.130.81 - Exploited by Stanford Uni' device. Results in outbound RAMEN alert.
Several other destinations were attacked using STATDX but do not appear to have been successfully exploited.

Recommendations: It is recommended that the above destination devices be removed from the network and investigated for exploitation.

Wingate 1080 Attempt

Description: This Snort detect has been included to detect probes for WinGate servers. Probes to port 1080 are generally designed to look for SOCKS proxy servers.

Ramifications: The purpose of locating a SOCKS proxy server is to allow the attacker to relay data connections through the server. Using this method they can successfully hide themselves and transfer data.

Interesting Sources:

199.173.178.2 -proxy.monitor.twisted.ma.us.dal.net
204.117.70.5
These addresses are the major perpetrators of the probes. No indication of success exists.

Interesting Destinations:

MY.NET.98.188 and MY.NET.97.80 were the major targets. But a large range of addresses were probed.

Recommendations: No sign of an exploit exists, so no recommendations are to be made.

SITE EXEC - Possible wu-ftpd exploit - GIAC000623

Description: This alert detects attempts to exploit a very common wu-ftp daemon vulnerability. It is designed to work against servers running wu-ftpd version 2.6.0

Ramifications: This exploit can allow the attacker to gain a remote shell on the victim machine as the root user. This attack has been successful against the victim host and has allowed the RAMEN exploit to be deployed.

Interesting Sources:

128.61.136.233 - tann6233.mse.gatech.edu
This device has successfully exploited this vulnerability on a local machine. The exploit has then been used to deploy the RAMEN worm.

It is likely that this source device is also a victim of the RAMEN worm. This can be determined by the large number of TCP/21 SYN scans originating from this device against the majority of the local network.

Interesting Destinations:

MY.NET.219.22
This host has been exploited by the RAMEN worm. As a result of this we can see outbound data destined for 24.67.186.244. 24.67.186.244 is likely to be a RAMEN client that has probed the internal network for machines that have been exploited with the RAMEN worm. Investigating the communications this address has with the internal networks we find that a large number of internal devices respond to the query to the RAMEN port 27374.

Recommendations: The device ate MY.NET.219.22 should be investigated very closely for the results of the wu-ftpd attack. The possibility of root-kits and other exploits existing on this device is very high. After initial analysis this device should be reinstalled with all patches applied.

External RPC call

Description: These alerts appear to be related to the attacks launched against the local network by the RAMEN worm. They are scanning the network for rpc.statd exploits that may be used for the insertion of the RAMEN worm in to the local network.

Ramifications: If any of these probes are successful it will allow the attacker to insert the worm and proceed to exploit the local network.

Interesting Sources:

171.65.61.201 - psych-3365-PC.Stanford.EDU
As seen earlier (STATDX alerts) this device appears to have been compormised already with the RAMEN worm. This device is now scanning the local network to try and propagate the worm further.

129.105.107.190 - dhcp107190.sesp.nwu.edu
It is very likely that this device has also been exploited by the RAMEN worm. It too probes the network for RPC vulnerabilities and STATDX exploits.

Interesting Destinations:

MY.NET.105.169
This device is likely to have been successfully exploited by 171.65.61.201 using the STATDX attack.

Recommendations: It is highly recommended that the device at MY.NET 105.169 be investigated for exploitation by the RAMEN worm.

Probable NMAP fingerprint attempt

Description: NMAP is a security auditing tool that allows the user to target a network, or host, with specially designed packets that are capable of discovering the profile of the targeted machines.

Ramifications: A probe of this nature is can usually be considered a precursor to an attack. It is used to customize the attack method to suit the target.

Interesting Sources: No source addresses are of interest.

Interesting Destinations: No destination addresses are of interest.

Recommendations: No security vulnerability was identified. No recommendations.

SUNRPC highport access!

Description: Access to the SUNRPC highport is used to attempt to circumvent security measures used to prevent RPC calls to target systems.

Ramifications: successful attempts at this attack will allow the perpetrator to communicate directly with services on the target machines. In this case the target machines are not SUN devices and as such will not be vulnerable to this attack.

Interesting Sources:

24.9.158.233 - cc916074-a.catv1.md.home.com
Alerts originating from this source appear to be response packets to an SecureShell session.

Possible RAMEN server attempt

Description: The Ramen Internet worm is a worm that is targeted at RedHat 6.2 and RedHat 7.0 devices. It scans network devices for vulnerabilities, in RPC, WU-FTPD, LRPNG (against RedHat 7.0). Upon detecting a vulnerable machine it exploits the target host installing itself and continuing the propagation process. It also installs a WWW server, modifies index.html files on the target machine and fixes the vulnerability it used to exploit the machine.

Ramifications: A successful exploit of this type allows the further propagation of the worm throughout the victim network and into other networks outside of the local area.

Interesting Sources:

A very large number of internal network devices show themselves likely to have been exploited by the RAMEN worm. It is very likely that the worm has entered the network via several means and has then propagated to the large majority of RedHat devices in the network environment.

The following devices are of more interest as they are very active.

24.67.186.244 - This device appears to be probing for exploited devices on the internal network that have become victims of the Ramen worm. This can be confirmed by determining that for every inbound packet, destined for port 27374 on the local network a response packet exists coming from the local network. It could well be that this is in fact a probe for SubSeven, but considering the number of replies and the proliferation of the Ramen worm throughout the local network it is less likely.

24.48.226.183 - pa-southhills2a-695.pit.adelphia.net
This device probes a very large section of the internal network looking for devices that have been exploited by the RAMEN worm. For the vast majority of inbound stimulus packets an outbound response packet exists. This further confirms the thought that a large number of internal hosts have been victimized.

MY.NET.201.146
This device appears to be one that has clearly been exploited by the Ramen worm. It can be seen probing other devices, external to the local network, for Ramen exploits.

MY.NET.253.12
As per MY.NET.201.146

MY.NET.97.154
As per MY.NET.201.146

Interesting Destinations:

In this case interesting destinations are much the same as interesting sources. The alert detects are such that the communications is detected in both directions. Such that as MY.NET.253.12 appears as an interesting source, so too it appears as an interesting destination.

Recommendations: All internal devices that are shown to have RAMEN alerts, where the internal device is the source, should be considered exploited and investigated as such.

Connect to 515 from inside

Description: This series of alerts is showing outbound traffic, originating from the internal network, destined for external hosts on destination port 515. This is a custom rule the IDS administrator has installed.

Ramifications: The alerts highlighted by this rule shows a series of what appear to be either a bidirectional communication or crafted packets. The source address seems to be targeting a very specific destination and it's source port does not change. The time frame for the packets is also very quick. It is most likely that this is normal communications with a device, perhaps a printer, outside of the monitored network.

Interesting Sources: No sources of security interest.

Interesting Destinations: No destinations of security interest.

Recommendations: It may be worth ensuring that the traffic observed is in fact what it seems.

Other comments: There also appears to be a large number of portscan alerts showing up from sources outside the local network, probing local network devices for a service listening on 515. It may be worth monitoring this more closely to determine if this is malicious or not.

NMAP TCP ping!

Description: An NMAP TCP (Ack) ping is a method of determining whether a network device is up, or down, without using the ICMP-ECHO, ICMP-ECHOREPLY method, as some people filter ICMP traffic out.

Ramifications: This type of probe can be a precursor to an attack as it allows the perpetrator to determine machine status without using ICMP. The vast majority of alerts shown are as a response to traffic across the internal network.

Interesting Sources: No source addresses of interest are present.

Interesting Destinations: No destination addresses of interest are present.

Recommendations: It would be recommended to investigate the source of the internal NMAP TCP ping alerts and determine if this is being done in violation of the acceptable use policy.

Attempted Sun RPC high port access

Description: As per "SUNRPC highport access!" alert.

Ramifications: As per "SUNRPC highport access!" alert.

Interesting Sources:

205.188.153.97:4000
This traffic is likely to be a response packet to an ICQ session that is originating from MY.NET.221.246:32771. The source address is in fact an ICQ server.

64.244.10.40:7777
This server is likely to be a n UnReal computer game server. The close time proximity of all the alerts would also suggest as much.

Interesting Destinations: No interesting destination addresses are present.

Recommendations: Review the acceptable use policy and if the above uses are in violation follow them up with the internal users.

Watchlist 000222 NET-NCFC

Description: This Snort rule has been implemented by the administrator to watch for traffic originating from the The Computer Network Center Chinese Academy of Sciences.

Ramifications: Alerts indicating detects form this network should be treated with suspicion.

Interesting Sources:

159.226.0.0 - 159.226.255.255
All packets arriving from this network should be treated as hostile.

Interesting Destinations: No destinations of interest are present.

Recommendations: Disallow traffic arriving from the above source network to enter the internal network.

UDP SRC and DST outside network

As per "TCP SRC and DST outside network" alerts.

Conclusions:

It must be concluded from the Snort information provided that the defined network environment is in serious trouble. A very virulent worm, RAMEN, has managed to get in to the network and has infested thousands of devices, and is proceeding to infect more machines all the time.

There are also serious problems with network traffic, including alternate routes form the outside world and tiny fragments that should be locked out.

To combat the RAMEN worm the network administrators need to use the Snort alert information and investigate all internal machines that are presented as generating RAMEN traffic. All of these devices appear to have been compromised, and should either be administered or reinstalled.

All patches for devices on the internal network should be applied. This includes all Microsoft Service Packs, and all RedHat patches. It is the lack of implementation of these components that has allowed the RAMEN and WU-FTPD vulnerabilities to exist.

Unfortunately no more information, other than that presented here, can be gleaned. The inconsistency of the data provided, the issues with missing data, duplicate data, and poorly designed rules, means that inconclusive results are often the result.

It is recommended that all existing Snort rules be revisited and ensure that they're purpose is clearly understood before implementation. The result otherwise will be an inordinate number of false-positives that will make diagnosis of real issues difficult.

Tools used in analysis:

SnortSnarf (http://www.silicondefense.com)
snort_sort.pl - Andrew R. Baker

Assignment References:

Snort, The Lightweight Network Intrusion Detection System, Marty Roesch,
http://www.snort.org
Tcpdump,
https://www.tcpdump.org
Whitehats, Network Security Resources, Max Vission,
http://www.whitehats.com
NetIQ Corporation,
http://www.netiq.com
Niksun Inc,

http://www.niksun.com

Five vendors answer some no-nonsense questions on IDS
Computer Security Institute.
http://www.gocsi.com/ques.htm

Case Study of Insider Sabotage
Computer Security Institute
Computer Security Journal
Sharon Gaudin, Vol XVI, Number 3, 2000
http://www.gocsi.com/insider.pdf

How to Handle and Identify Network Probes
Ron Gula, April 1999
http://www.securitywizards.com/papers/probes.html

Tough questions for IDS vendors
Computer Security Institute
http://www.gocsi.com/IDSques.htm

An Introduction To Intrusion Detection & Assessment
ICSA Labs, 4 January 2000
http://www.icsa.net/html/communities/ids/White%20paper/Intrusion1.pdf

Last Updated: May 29th, 2017

Upcoming Training

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