Global Information Assurance Certification Paper

Intrusion Detection in Depth
SANS GIAC Practical Assignment

SANS Darling Harbour 2001
Version 2.8b

David Sarmanian

Contents

Assignment 1

My Network - Attacked!

Attack 1. IIS Unicode Attack

```
May 13 12:48:27 210.33.68.1:63021 -> x.x.42.17:80 SYN ******S*
May 13 12:48:27 210.33.68.1:63039 -> x.x.42.35:80 SYN ******S*
May 13 12:48:27 210.33.68.1:63006 -> x.x.42.2:80 SYN ******S*
May 13 12:48:27 210.33.68.1:63014 -> x.x.42.10:80 SYN ******S*
May 13 12:48:27 210.33.68.1:63033 -> x.x.42.29:80 SYN ******S*
May 13 12:48:27 210.33.68.1:63035 -> x.x.42.31:80 SYN ******S*
May 13 12:48:27 210.33.68.1:63008 -> x.x.42.4:80 SYN ******S*
May 13 12:48:27 210.33.68.1:63009 -> x.x.42.5:80 SYN ******S*
May 13 12:48:27 210.33.68.1:63013 -> x.x.42.9:80 SYN ******S*
May 13 12:48:27 210.33.68.1:63015 -> x.x.42.11:80 SYN ******S*
May 13 12:48:31 210.33.68.1:63022 -> x.x.42.18:80 SYN ******S*
May 13 12:48:32 210.33.68.1:65280 -> x.x.42.76:80 SYN ******S*
May 13 12:48:32 210.33.68.1:65281 -> x.x.42.77:80 SYN ******S*
May 13 12:48:32 210.33.68.1:65282 -> x.x.42.78:80 SYN ******S*
May 13 12:48:32 210.33.68.1:65272 -> x.x.42.68:80 SYN ******S*
May 13 12:48:32 210.33.68.1:65291 -> x.x.42.87:80 SYN ******S*
May 13 12:48:32 210.33.68.1:65274 -> x.x.42.70:80 SYN ******S*
May 13 12:48:32 210.33.68.1:65294 -> x.x.42.90:80 SYN ******S*
May 13 12:48:32 210.33.68.1:65276 -> x.x.42.72:80 SYN ******S*
May 13 12:48:32 210.33.68.1:65277 -> x.x.42.73:80 SYN ******S*
May 13 12:48:32 210.33.68.1:65278 -> x.x.42.74:80 SYN ******S*
May 13 12:48:32 210.33.68.1:65279 -> x.x.42.75:80 SYN ******S*
May 13 12:48:36 210.33.68.1:33993 -> x.x.42.126:80 SYN ******S*
[**] spp http decode: IIS Unicode attack detected [**]
05/13-15:13:58.929261 210.33.68.1:54703 -> x.x.42.4:80
TCP TTL:229 TOS:0x0 ID:12853 IpLen:20 DgmLen:106 DF
***AP*** Seq: 0x9915DB72 Ack: 0xED0F1F0D Win: 0x5B4 TcpLen: 20 [Snort log]
```

```
Snort Application Layer Dump:
[**] spp_http_decode: IIS Unicode attack detected [**]
05/13-15:13:58.929261 210.33.68.1:54703 -> x.x.42.4:80
TCP TTL:229 TOS:0x0 ID:12853 IpLen:20 DgmLen:106 DF
***AP*** Seq: 0x9915DB72  Ack: 0xED0F1F0D  Win: 0x5B4  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 25  GET /scripts/..%
63 30 25 61 66 2E 2E 2F 77 69 6E 6E 74 2F 73 79  c0%af../winnt/sy
73 74 65 6D 33 32 2F 63 6D 64 2E 65 78 65 3F 2F  stem32/cmd.exe?/
63 2B 64 69 72 20 48 54 54 50 2F 31 2E 30 0D 0A  c+dir HTTP/1.0..
0D 0A                                            ..

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

Detect 1

Source of the Trace:
This trace comes from a sensor placed on my network between our border router and our Internet firewall.

Detect was Generated By:
This detect was generated by a Redhat Linux system running Snort 1.7 using the standard rule set from the Snort homepage. The specific rule, which captured the traffic, was:
web-misc.rules:alert tcp $HTTP_SERVERS 80 -> $EXTERNAL_NET any (msg:"WEB-MISC 403 Forbidden";flags: A+; content:"HTTP/1.1 403";)
This rule was set to flag abnormal Unicode to or through the firewall on port 80.

Here is the output from ARIN. It is evident the IP address space belongs to:

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

   Netname: APNIC-CIDR-BLK2
   Netblock: 210.0.0.0 - 211.255.255.255

   Coordinator:
      Administrator, System  (SA90-ARIN)  sysadm@APNIC.NET
      +61-7-3367-0490

   Domain System inverse mapping provided by:

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

```
Regional Internet Registry for the Asia-Pacific Region.
```

Digging a little deeper I was able to track down exactly who owns this address

http://www.apnic.net/apnic-bin/whois.pl?search=210.33.68.1

```
Regional Internet Registry for the Asia-Pacific Region.
      inetnum:     210.33.68.0 - 210.33.71.255
      netname:     WZTC-CN
      descr:       ~{NBV]J&76Q'T:~}
      descr:       Wenzhou Teachers College
      descr:       Wenzhou, Zhejiang 325003, China
      country:     CN
      admin-c:     GW6-CN
      tech-c:      GX5-CN
      notify:      address-allocation-staff@net.edu.cn
      changed:     szhu@net.edu.cn 970114
      source:      APNIC
```

Probability the Source Address was Spoofed:
The probability of this attack coming from a spoofed address is not likely. For this attack to be effective the attacker must see the response from our web server. The packet that caused this attack is usually part of an already established TCP session. Therefore the attacking IP address is valid. It is evident, the attacker first scanned the subnet specifically targeting systems with port 80 open and then attacked. Doing so gave his intentions away.

Description of Attack:
This type of attack is generally designed to exploit known weakness in Microsoft IIS web servers when trying to parse Unicode requests. The attacker sends GET requests that contain commands to run a program instead of normal GET requests to access data from the web pages.

Attack Mechanism:
This type of attack is again almost always directed at Microsoft IIS 4 & 5 web servers. The goal of the attack is to execute arbitrary commands on the webserver and gain access to the root file system. Numerous variations of this attack have been identified at whitehats, CVE and bugtrack. The systems, which are affected, are usually MS IIS 4.0 on NT 4.0, which were configured with the default settings.

Correlations:
According to Insidents.org the targeting of port 80 is still a very popular target port today. The CVE, Bugtrack and ADVice numbers are as follows for this exploit.
CVE: CAN-2000-0884
Bugtrack: http://www.securityfocus.com/bid/1806
Advice: http://advice.networkice.com/Advice/Intrusions/2000639/default.htm

Evidence of Active Targeting:
It is evident from the trace above, the attacker is actively targeting any system running any www services. First the attacker does a syn scans. Once the attacker finds a system servicing port 80 requests, they run their kiddy script against the system. The good news is, the script fails.

Severity:
Target Criticality =2
This is our primary firewall that allows all users to access the Internet during off work hours.

Attack Lethality = 2
This attack was directed at a Microsoft IIS server. Our firewall is not Microsoft-based.

System Countermeasures = 2
According to the manufacturer, the firewall does not have any known vulnerabilities to IIS Unicode exploits.

Network Countermeasures = 3
Continue to monitor all traffic directed to our firewall on port 80. Continually update all snort rules to detect this type of activity.

(Criticality + Lethality) – (System Countermeasures + Network Countermeasures) = Severity
(2+2) - (2+3) = -1

Defensive Recommendation:
Patch your systems or run Apache or any other non-Microsoft web server. The patch was released with the advisory MS00-057 from Microsoft. This patch eliminates the vulnerability.

Multiple-Choice Test Question:

```
05/13-15:13:58.929261 210.33.68.1:54703 -> x.x.42.4:80
TCP TTL:229 TOS:0x0 ID:12853 IpLen:20 DgmLen:106 DF
***AP*** Seq: 0x9915DB72  Ack: 0xED0F1F0D  Win: 0x5B4  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 25  GET /scripts/..%
63 30 25 61 66 2E 2E 2F 77 69 6E 6E 74 2F 73 79  c0%af../winnt/sy
73 74 65 6D 33 32 2F 63 6D 64 2E 65 78 65 3F 2F  stem32/cmd.exe?/
63 2B 64 69 72 20 48 54 54 50 2F 31 2E 30 0D 0A  c+dir HTTP/1.0..
0D 0A
                                                 . .
```

What is the purpose of this scan?

A. Access a web page on a remote server
B. Push data to a web page on a remote server
C. Check to see if this server is really an IIS server
D. Read documents outside of the web root, and possibly execute arbitrary commands

Answer: D

Detect 2

EXPLOIT x86 stealth noop

Snort Application Layer Dump:

```
[**] IDS181/shellcode-x86-nops [**]
05/16-11:43:26.221476 212.208.244.69:80 -> x.x.42.2:38127
TCP TTL:45 TOS:0x0 ID:16209 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0xAB1CCC6  Ack: 0x2201049D  Win: 0xB68  TcpLen: 20
33 00 00 00 33 00 33 00 33 00 66 00 33 00 99 00  3...3.3.3.f.3...
33 00 CC 00 33 00 FF 00 33 33 00 00 33 33 33 00  3...3...33..333.
33 33 66 00 33 33 99 00 33 33 CC 00 33 33 FF 00  33f.33..33..33..
33 66 00 00 33 66 33 00 33 66 66 00 33 66 99 00  3f..3f3.3ff.3f..
33 66 CC 00 33 66 FF 00 33 99 00 00 33 99 33 00  3f..3f..3...3.3.
33 99 66 00 33 99 99 00 33 99 CC 00 33 99 FF 00  3.f.3...3...3...
33 CC 00 00 33 CC 33 00 33 CC 66 00 33 CC 99 00  3...3.3.3.f.3...
33 CC CC 00 33 CC FF 00 33 FF 33 00 33 FF 66 00  3...3...3.3.3.f.
33 FF 99 00 33 FF CC 00 33 FF FF 00 66 00 00 00  3...3...3...f...
66 00 33 00 66 00 66 00 66 00 99 00 66 00 CC 00  f.3.f.f.f...f...
66 00 FF 00 66 33 00 00 66 33 33 00 66 33 66 00  f...f3..f33.f3f.
66 33 99 00 66 33 CC 00 66 33 FF 00 66 66 00 00  f3..f3..f3..ff..
66 66 33 00 66 66 66 00 66 66 99 00 66 66 CC 00  ff3.fff.ff..ff..
66 99 00 00 66 99 33 00 66 99 66 00 66 99 99 00  f...f.3.f.f.f...
66 99 CC 00 66 99 FF 00 66 CC 00 00 66 CC 33 00  f...f...f...f.3.
66 CC 99 00 66 CC CC 00 66 CC FF 00 66 FF 00 00  f...f...f...f...
66 FF 33 00 66 FF 99 00 66 FF CC 00 CC 00 FF 00  f.3.f...f.......
FF 00 CC 00 99 99 00 00 99 33 99 00 99 00 99 00  .........3......
99 00 CC 00 99 00 00 00 99 33 33 00 99 00 66 00  .........33...f.
99 33 CC 00 99 00 FF 00 99 66 00 00 99 66 33 00  .3.......f...f3.
99 33 66 00 99 66 99 00 99 66 CC 00 99 33 FF 00  .3f..f...f...3..
99 99 33 00 99 99 66 00 99 99 99 00 99 99 CC 00  ..3...f.........
99 99 FF 00 99 CC 00 00 99 CC 33 00 66 CC 66 00  ..........3.f.f.
99 CC 99 00 99 CC CC 00 99 CC FF 00 99 FF 00 00  ................
99 FF 33 00 99 CC 66 00 99 FF 99 00 99 FF CC 00  ..3...f.........
99 FF FF 00 CC 00 00 00 99 00 33 00 CC 00 66 00  ..........3...f.
CC 00 99 00 CC 00 CC 00 99 33 00 00 CC 33 33 00  .........3...33.
CC 33 66 00 CC 33 99 00 CC 33 CC 00 CC 33 FF 00  .3f..3...3...3..
CC 66 00 00 CC 66 33 00 99 66 66 00 CC 66 99 00  .f...f3..ff..f..
CC 66 CC 00 99 66 FF 00 CC 99 00 00 CC 99 33 00  .f...f........3.
CC 99 66 00 CC 99 99 00 CC 99 CC 00 CC 99 FF 00  ..f.............
CC CC 00 00 CC CC 33 00 CC CC 66 00 CC CC 99 00  ......3...f.....
CC CC CC 00 CC CC FF 00 CC FF 00 00 CC FF 33 00  ..............3.
99 FF 66 00 CC FF 99 00 CC FF CC 00 CC FF FF 00  ..f.............
CC 00 33 00 FF 00 66 00 FF 00 99 00 CC 33 00 00  ..3...f......3..
FF 33 33 00 FF 33 66 00 FF 33 99 00 FF 33 CC 00  .33..3f..3...3..
FF 33 FF 00 FF 66 00 00 FF 66 33 00 CC 66 66 00  .3...f...f3..ff.
```

```
FF 66 99 00 FF 66 CC 00 CC 66 FF 00 FF 99 00 00  .f...f...f......
FF 99 33 00 FF 99 66 00 FF 99 99 00 FF 99 CC 00  ..3...f.........
FF 99 FF 00 FF CC 00 00 FF CC 33 00 FF CC 66 00  ..........3...f.
FF CC 99 00 FF CC CC 00 FF CC FF 00 FF FF 33 00  ..............3.
CC FF 66 00 FF FF 99 00 FF FF CC 00 66 66 FF 00  ..f.........ff..
66 FF 66 00 66 FF FF 00 FF 66 66 00 FF 66 FF 00  f.f.f....ff..f..
FF FF 66 00 21 00 A5 00 5F 5F 5F 00 77 77 77 00  ..f.!...___.www.
86 86 86 00 96 96 96 00 CB CB CB 00 B2 B2 B2 00  ................
D7 D7 D7 00 DD DD DD 00 E3 E3 E3 00 EA EA EA 00  ................
F1 F1 F1 00 F8 F8 F8 00 F0 FB FF 00 A4 A0 A0 00  ................
80 80 80 00 00 00 FF 00 00 FF 00 00 00 FF FF 00  ................
FF 00 00 00 FF 00 FF 00 FF FF 00 00 FF FF FF 00  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 8B 66 0A 0A 0A 0A 66 90 90  ........f....f..
90 90 8B 04 04 04 04 04 8B 90 90 90 90 90 90 90  ................
DB DB DB DB DB DB 66 0A 23 32 32 32 32 0A 66 90  ......f.#2222.f.
DB 8B 66 38 38 38 38 38 12 8B DB DB DB DB DB DB  ..f88888........
90 90 90 90 90 8B 0A 32 32 32 32 32 32 32 32 0A  .......2222222.f
8B EB 38 38 38 38 38 38 38 EB 8B 90 90 90 90 90  ..8888888.......
90 90 90 90 8B 66 22 32 32 32 32 32 32 32 32 0A  .....f"22222222.
04 38 38 38 38 38 38 38 38 38 66 8B 90 90 90 90  .888888888f.....
DB DB DB DB 86 23 2A 32 32 32 32 32 32 32 32 0A  .....#*22222222.
EA 38 38 38 38 38 38 38 38 38 30 86 DB DB DB DB  .8888888880.....
90 90 90 90 04 23 2A 32 32 32 32 32 32 32 32 0A  .....#*22222222.
37 38 38 38 38 38 38 38 38 38 38 04 90 90 90 90  78888888888.....
90 90 90 90 04 30 22 2A 22 32 32 32 32 32 32 0A  .....0"*"222222.
29 0A 0A 0A 0A 0A 29 37 38 38 38 04 90 90 90 90  ).....)7888.....
DB DB DB DB 04 38 30 23 29 23 32 32 32 32 32 0A  .....80#)#22222.
0A 32 32 32 32 32 2A 29 38 38 38 04 DB DB DB DB  .22222*)888.....
90 90 90 90 86 73 38 38 03 23 32 32 32 32 32 0A  .....s88.#22222.
0A 32 32 32 32 32 29 29 38 38 EB 8B 90 90 90 90  .22222))88......
90 90 90 90 8B 86 38 38 03 23 32 32 32 32 32 0A  ......88.#22222.
0A 32 32 32 32 32 23 03 38 37 86 90 90 90 90 90  .22222#.87......
DB DB DB DB DB 8B 12 38 03 23 32 32 32 32 32 0A  .......8.#22222.
0A 32 32 32 32 32 23 03 38 12 8B DB DB DB DB DB  .22222#.8.......
90 90 90 8B 86 66 86 36 03 23 32 32 32 32 32 0A  .....f.6.#22222.
0A 32 32 32 32 32 23 03 73 EB 12 04 8B 90 90 90  .22222#.s.......
90 90 8B 04 37 38 38 38 03 23 32 32 32 32 32 0A  ....7888.#22222.
0A 32 32 32 32 32 23 03 38 38 38 38 86 8B 90 90  .22222#.8888....
DB 8B 04 38 38 38 38 38 03 23 32 32 32 32 32 0A  ...88888.#22222.
0A 32 32 32 32 32 23 03 38 38 38 38 38 04 8B DB  .22222#.88888...
90 86 36 38 38 38 38 38 03 23 32 32 32 32 32 0A  ..688888.#22222.
0A 32 32 32 32 32 23 03 38 38 38 38 38 38 04 90  .22222#.888888..
90 04 38 38 38 38 38 38 03 23 32 32 32 32 32 0A  ..888888.#22222.
0A 32 32 32 32 32 23 03 38 38 38 38 38 38 04 90  .22222#.888888..
90 66 38 38 38 38 38 38 03 23 32 32 32 32 32 0A  .f888888.#22222.
0A 32 32 32 32 32 29 29 38 38 38 38 38 38 66 90  .22222))888888f.
8B 12 38 38 38 38 38 38 03 23 32 32 32 32 32 0A  ..888888.#22222.
0A 32 32 32 32 32 22 0A 29 38 38 38 38 38 12 8B  .22222".)88888..
90 04 38 38 38 38 38 38 03 23 32 32 32 32 32 0A  ..888888.#22222.
0A 32 32 32 32 32 32 32 1D 37 38 38 38 38 04 90  .2222222.78888..
DB 86 37 38 38 38 38 38 03 23 32 32 32 32 32 0A  ..788888.#22222.
0A 32 32 32 32 32 32 32 0A 38 38 38 38 37 86 DB  .2222222.88887..
90 8B 04 38                                      ...8
```

Source of the Trace:
Snort 1.7 captured the following trace. The sensor was located between a border router and a main firewall.

Detect was Generated By:
This detect was generated by the following whitehats.com rule set.

alert TCP $EXTERNAL any -> $INTERNAL any (msg: "IDS181/shellcode-x86-nops"; flags: A+; content: "|90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90|";) (from *vision.rules*)

Here is the Output from ARIN. It is evident the IP address space belongs to:

```
European Regional Internet Registry/RIPE NCC (NET-RIPE-NCC-)
   These addresses have been further assigned to European users.
   Contact info can be found in the RIPE database, via the
   WHOIS and TELNET servers at whois.ripe.net, and at
   http://www.ripe.net/db/whois.html
   NL

   Netname: RIPE-NCC-212
   Netblock: 212.0.0.0 - 212.255.255.255
   Maintainer: RIPE

   Coordinator:
      Reseaux IP European Network Co-ordination Centre Singel 258
(RIPE-NCC-ARIN)  nicdb@RIPE.NET
      +31 20 535 4444

   Domain System inverse mapping provided by:

   NS.RIPE.NET                 193.0.0.193
   NS.EU.NET                   192.16.202.11
   AUTH03.NS.UU.NET            198.6.1.83
   NS2.NIC.FR                  192.93.0.4
   SUNIC.SUNET.SE              192.36.125.2
   MUNNARI.OZ.AU               128.250.1.21
   NS.APNIC.NET                203.37.255.97
```

Probability the Source Address was Spoofed:
Both the source and destination addresses are almost certainly real.

Description of Attack:
This type of attack is directed at all x86 processor systems. Adding 90 90 90 90 90 in the string is a technique in executing a buffer overflow against the system. What is abnormal about this attack is the use of 888888.#22222 within the packet.

Attack Mechanism:

In first analyzing this attack I had to ask the four basic questions any analyst asks when trying to determine the validity of an attack. "Is this a stimulus or response? What service is being targeted? Does the service have known vulnerabilities or exposures? Is this benign, an exploit, denial of service, or reconnaissance?" (Northcut, Cooper, Fearnon, Frederick, 2001) My answers are as follows: This is a response because the source port of the attacking host is 80 (web server) and the destination port (38127) is an ephemeral port. Just to make sure I was not missing anything, I ran a check against port 38127 at Snort.org Port Search Database and it returned that "no record was found". Also, with the ACK (A) flag being set, this almost assures that the host "being attacked" requested the packet.

I then asked the person who owned the system in question if they had ever visited the www site in question. They indicated yes! After further review, the signature in question actually matches a JPG that is located on the www site in question. This confirms my theory that the 888888.#22222 is just a picture and not a new exploit being tested.

Correlations:
After many searches on google.com, sans.org, whitehats.com, and bugtrack, I was unable to match the pattern that set off the sensor. This pattern is just a false positive generated by the vision.rules from whitehats.

Evidence of Active Targeting:
Because this is a false positive, there was no active targeting or reconnaissance involved.

Severity:
Target Criticality =1
This system belongs to a user who dials in from home to access remote web pages on the Internet.

Attack Lethality = 0
This was a false positive and not an attack.

System Countermeasures = 4
The system in question was a windows 2000 system, which was not vulnerable to this type of exploit. I did recommend a personal firewall the next time they dial into the network to access the Internet.

Network Countermeasures = 5
Continue to monitor all traffic directed to our firewall on port 80. Continually update all snort rules to detect this type of activity.

(Criticality + Lethality) – (System Countermeasures + Network Countermeasures) = Severity
(1+0) - (4+5) = -8

Defensive recommendation:

Add a personal firewall to the system in question so they are not prone to attacks from the Internet. Make sure all Snort rules are kept up-to-date.

Multiple-Choice Test Question:
What type of detect is this?

```
05/13-15:13:58.929261 210.33.68.1:54703 -> x.x.42.4:80
TCP TTL:45 TOS:0x0 ID:16209 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0xAB1CCC6  Ack: 0x2201049D  Win: 0xB68  TcpLen: 20
47 45 54 20 2F 90 90 90 90 90 90 90 90 90 90 90  GET /.....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
```

A. Trojan Horse exploit.
B. Buffer Overflow.
C. Normal HTTP traffic.
D. A good one.

Answer: B

Detect 3

Named-probe-iquery

```
May 25 11:42:36 202.130.248.188:2200 -> x.x.42.10:53 SYN ******S*
May 25 11:42:39 202.130.248.188:3101 -> x.x.42.10:53 UDP
May 25 11:42:36 202.130.248.188:2192 -> x.x.42.2:53 SYN ******S*
May 25 11:42:38 202.130.248.188:3098 -> x.x.42.2:53 UDP
May 25 11:42:36 202.130.248.188:2199 -> x.x.42.9:53 SYN ******S*
May 25 11:42:38 202.130.248.188:3100 -> x.x.42.9:53 UDP
May 25 11:42:39 202.130.248.188:2217 -> x.x.42.27:53 SYN ******S*
May 25 11:42:36 202.130.248.188:2215 -> x.x.42.25:53 SYN ******S*
May 25 11:42:36 202.130.248.188:2256 -> x.x.42.66:53 SYN ******S*
May 25 11:42:39 202.130.248.188:2261 -> x.x.42.71:53 SYN ******S*
May 25 11:42:36 202.130.248.188:2259 -> x.x.42.69:53 SYN ******S*
May 25 11:42:45 202.130.248.188:2195 -> x.x.42.5:53 SYN ******S*
May 25 11:42:36 202.130.248.188:2280 -> x.x.42.90:53 SYN ******S*
May 25 11:42:36 202.130.248.188:2283 -> x.x.42.93:53 SYN ******S*
May 25 11:42:36 202.130.248.188:2196 -> x.x.42.6:53 SYN ******S*
May 25 11:42:36 202.130.248.188:2201 -> x.x.42.11:53 SYN ******S*
May 25 11:42:39 202.130.248.188:3103 -> x.x.42.11:53 UDP
May 25 11:42:36 202.130.248.188:2194 -> x.x.42.4:53 SYN ******S*
May 25 11:42:38 202.130.248.188:3099 -> x.x.42.4:53 UDP
May 25 11:42:36 202.130.248.188:2197 -> x.x.42.7:53 SYN ******S*
May 25 11:42:39 202.130.248.188:2208 -> x.x.42.18:53 SYN ******S*
May 25 11:42:36 202.130.248.188:2207 -> x.x.42.17:53 SYN ******S*
May 25 11:42:36 202.130.248.188:2212 -> x.x.42.22:53 SYN ******S*
May 25 11:42:36 202.130.248.188:2268 -> x.x.42.78:53 SYN ******S*
```

```
May 25 11:42:36 202.130.248.188:2260 -> x.x.42.70:53 SYN ******S*
May 25 11:42:39 202.130.248.188:2282 -> x.x.42.92:53 SYN ******S*
May 25 11:42:39 202.130.248.188:2281 -> x.x.42.91:53 SYN ******S*
May 25 11:42:39 202.130.248.188:2284 -> x.x.42.94:53 SYN ******S*
May 25 11:42:39 202.130.248.188:2316 -> x.x.42.126:53 SYN ******S*
May 25 11:42:45 202.130.248.188:2208 -> x.x.42.18:53 SYN ******S*
May 25 11:42:45 202.130.248.188:2217 -> x.x.42.27:53 SYN ******S*
May 25 11:42:45 202.130.248.188:2261 -> x.x.42.71:53 SYN ******S*
May 25 11:42:45 202.130.248.188:2284 -> x.x.42.94:53 SYN ******S*
May 25 11:42:45 202.130.248.188:2281 -> x.x.42.91:53 SYN ******S*
```

Snort Application Layer Dump:

```
[**] IDS277/named-probe-iquery [**]
05/25-11:42:38.146232 202.130.248.188:3099 -> x.x.42.4:53
UDP TTL:52 TOS:0x0 ID:10868 IpLen:20 DgmLen:493
Len: 473
6F 62 09 80 00 00 00 01 00 00 00 00 3E 41 41 41  ob..........>AAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 3E 42 42 42 42  AAAAAAAAAAA>BBBB
42 42 42 42 42 42 42 42 42 42 42 42 42 42 42 42  BBBBBBBBBBBBBBBB
42 42 42 42 42 42 42 42 42 42 42 42 42 42 42 42  BBBBBBBBBBBBBBBB
42 42 42 42 42 42 42 42 42 42 42 42 42 42 42 42  BBBBBBBBBBBBBBBB
42 42 42 42 42 42 42 42 42 42 3E 43 43 43 43 43  BBBBBBBBBB>CCCCC
43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43  CCCCCCCCCCCCCCCC
43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43  CCCCCCCCCCCCCCCC
43 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43  CCCCCCCCCCCCCCCC
43 43 43 43 43 43 43 43 43 3E 00 01 02 03 04 05  CCCCCCCCC>......
06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15  ................
16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25  .......... !"#$%
26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35  &'()*+,-./012345
36 37 38 39 3A 3B 3C 3D 3E 45 45 45 45 45 45 45  6789:;<=>EEEEEEE
45 45 45 45 45 45 45 45 45 45 45 45 45 45 45 45  EEEEEEEEEEEEEEEE
45 45 45 45 45 45 45 45 45 45 45 45 45 45 45 45  EEEEEEEEEEEEEEEE
45 45 45 45 45 45 45 45 45 45 45 45 45 45 45 45  EEEEEEEEEEEEEEEE
45 45 45 45 45 45 45 3E 46 46 46 46 46 46 46 46  EEEEEEE>FFFFFFFF
46 46 46 46 46 46 46 46 46 46 46 46 46 46 46 46  FFFFFFFFFFFFFFFF
46 46 46 46 46 46 46 46 46 46 46 46 46 46 46 46  FFFFFFFFFFFFFFFF
46 46 46 46 46 46 46 46 46 46 46 46 46 46 46 46  FFFFFFFFFFFFFFFF
46 46 46 46 46 46 3D 47 47 47 47 47 47 47 47 47  FFFFFF=GGGGGGGGG
47 47 47 47 47 47 47 47 47 47 47 47 47 47 47 47  GGGGGGGGGGGGGGGG
47 47 47 47 47 47 47 47 47 47 47 47 47 47 47 47  GGGGGGGGGGGGGGGG
47 47 47 47 47 47 47 47 47 47 47 47 47 47 47 47  GGGGGGGGGGGGGGGG
47 47 47 47 00 00 01 00 01 00 00 00 01 00 FF 00  GGGG............
44                                               D

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

Source of trace:

The sensor was located between a border router and the main firewall

Detect was Generated By:

This detect was generated by the following whitehats.com rule set.

alert UDP $EXTERNAL any -> $INTERNAL 53 (msg: "IDS277/named-probe-iquery"; content: "|0980 0000 0001 0000 0000|"; depth: 16; offset: 2;) (from *vision.rules*)

Probability the Source Address was Spoofed:
This attack shows how the attacker attempted to determine if the name server on a DNS server supports IQUERY. Since a UDP packet generated the request by the attacker, the source IP address could be easily forged. However, due to the SYN scan before the actual iquery took place, the SYN scan confirms the validity of the source IP address of the attacking host.

The attacking host was from the following address space:

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

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

   Coordinator:
      Administrator, System  (SA90-ARIN)  sysadm@APNIC.NET
      +61-7-3367-0490

   Domain System inverse mapping provided by:

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

After further review the following address comes up again!

http://www.apnic.net/apnic-bin/whois.pl?search=202.130.248.188

```
inetnum:    202.130.224.0 - 202.130.255.255
netname:     EASTTELECOM
descr:      EAST TELECOMMUNICATION CO. LTD.
descr:      National ISP in PRC
descr:      Beijing, P.R.China
country:     CN
admin-c:     GH5-AP
tech-c:     HG2-AP
rev-srv:     ns.east.cn.net
rev-srv:     info.orinet.co.cn
remarks:     service provider
changed:     haixiang@public.east.cn.net 970610
source:     APNIC
```

Description of Attack:
This type of attack is directed at systems running pre versions of Bind 4.9.8 and 8.1.2. There are numerous buffer overflow attacks, which cause the nameserver daemon to fail, and root access to be granted when sent certain types of queries fail to properly process an inverse query. The exploit can cause memory to not copy portions of the request thus allowing portions of the program to be overwritten, then arbitrary commands can be run on the exploited host by anyone. Since the target host computer was another burb on our firewall, and the OS was hardened with the latest patches, this attack did not have any effect on our network DNS server.

Attack Mechanism:
This is definitely a stimulus from the attacker. Using many ephemeral ports to pre SYN scan our subnet, the attacker tries to flush out all systems with port 53 open. After the reconnaissance is done, the attacker systematically tries to exploit those systems they think are running bind. The attacker does do a version test before they run the named probe iquery. The actual named probe is not shown here. The Duck principle applies here. According to the Insidents.org web page, during the time of this attack, the DNS vulnerability was one of the top exploits.

Correlations:
This is a very common attack out on the Internet. According to incidents.org, scanning for port 53 is the second most common attack out on the Internet next to port 111.
http://www.incidents.org/cid/query/top_10port_7.php
This DNS attack has CVE, Bugtracq and ADVice numbers.
CVE CVE-1999-009
Bugtracq 134
ADVICE 2000409

Evidence of Active Targeting:
Pre SYN scans by the attacking hosts against this hosts is a strong indicator that the IP address is valid and this attack was deliberate.
If this was just a simple udp scan targeting port 53 with no pre SYN attack, then maybe one would consider this just a misguided packet looking for another DNS server. This was not the case.

Severity:
Target Criticality =4
This system is the primary DNS server for all of the Internet access on that network. This system is also the primary firewall, which connects that network to the public Internet.

Attack Lethality = 4
This attack was an exploit, although the exploit failed to work due to the hardened OS and the patch levels were up-to-date.

System Countermeasures = 4
This system is a firewall, which is hardened and monitored. If it were another system which was not hardened then there would have been a different outcome.

Network Countermeasures = 5
The firewall stopped this attack.

(Criticality + Lethality) – (System Countermeasures + Network Countermeasures) = Severity
(4+4) - (4+5) = -1

Defensive Recommendation:
Continue patching all DNS systems on the network with the latest version of Bind. Make sure that all IDS systems are also up-to-date with new rules files.

Multiple-Choice Test Question:
What version of Bind is suitable to the Named-probe-iquery?

A. Bind 9.1
B. Bind 4.9.8
C. Bind 8.2.3
D. Bind 4.0

Answer is D

Detect 4

Linuxconf Buffer Overflow

```
portscan.log:May 28 01:33:24 200.204.151.236 :1891 -> x.x.42.5:98 SYN ******S*
portscan.log:May 28 01:33:21 200.204.151.236 :1893 -> x.x.42.7:98 SYN ******S*
portscan.log:May 28 01:33:24 200.204.151.236 :1895 -> x.x.42.9:98 SYN ******S*
portscan.log:May 28 01:33:21 200.204.151.236 :1903 -> x.x.42.17:98 SYN ******S*
portscan.log:May 28 01:33:24 200.204.151.236 :1896 -> x.x.42.10:98 SYN ******S*
portscan.log:May 28 01:33:21 200.204.151.236 :1908 -> x.x.42.22:98 SYN ******S*
portscan.log:May 28 01:33:21 200.204.151.236 :1952 -> x.x.42.66:98 SYN ******S*
portscan.log:May 28 01:33:21 200.204.151.236 :1955 -> x.x.42.69:98 SYN ******S*
portscan.log:May 28 01:33:21 200.204.151.236 :1963 -> x.x.42.77:98 SYN ******S*
portscan.log:May 28 01:33:21 200.204.151.236 :1957 -> x.x.42.71:98 SYN ******S*
portscan.log:May 28 01:33:24 200.204.151.236 :1958 -> x.x.42.72:98 SYN ******S*
portscan.log:May 28 01:33:21 200.204.151.236 :1961 -> x.x.42.75:98 SYN ******S*
portscan.log:May 28 01:33:21 200.204.151.236 :1956 -> x.x.42.70:98 SYN ******S*
portscan.log:May 28 01:33:24 200.204.151.236 :1978 -> x.x.42.92:98 SYN ******S*
portscan.log:May 28 01:33:24 200.204.151.236 :1980 -> x.x.42.94:98 SYN ******S*
portscan.log:May 28 01:33:24 200.204.151.236 :1888 -> x.x.42.2:98 SYN ******S*
portscan.log:May 28 01:33:24 200.204.151.236 :1911 -> x.x.42.25:98 SYN ******S*
portscan.log:May 28 01:33:24 200.204.151.236 :1890 -> x.x.42.4:98 SYN ******S*
portscan.log:May 28 01:33:24 200.204.151.236 :1979 -> x.x.42.93:98 SYN ******S*
portscan.log:May 28 01:33:24 200.204.151.236 :1892 -> x.x.42.6:98 SYN ******S*
portscan.log:May 28 01:33:24 200.204.151.236 :1913 -> x.x.42.27:98 SYN ******S*
portscan.log:May 28 01:33:24 200.204.151.236 :1897 -> x.x.42.11:98 SYN ******S*
portscan.log:May 28 01:33:24 200.204.151.236 :1904 -> x.x.42.18:98 SYN ******S*
portscan.log:May 28 01:33:24 200.204.151.236 :1962 -> x.x.42.76:98 SYN ******S*
portscan.log:May 28 01:33:24 200.204.151.236 :1972 -> x.x.42.86:98 SYN ******S*
portscan.log:May 28 01:33:24 200.204.151.236 :1975 -> x.x.42.89:98 SYN ******S*
portscan.log:May 28 01:33:24 200.204.151.236 :1976 -> x.x.42.90:98 SYN ******S*
```

Source of the Trace:

This trace comes from a sensor placed on my network between our border router and our Internet firewall.

Detect was Generated By:

This detect was generated by a Redhat Linux System running Snort 1.7 using the standard rule set from the Snort homepage. The specific rule, which captured the traffic, was:

alert TCP $EXTERNAL any -> $INTERNAL 98 (msg: "Local Rules";) (Local Rules)

Here is the Output from ARIN.

```
RNP (Brazilian Research Network) (NETBLK-BRAZIL-BLK2)
   These addresses have been further assigned to Brazilian users.
   Contact information can be found at the WHOIS server located
   at whois.registro.br and at http://whois.nic.br
   BR
```

```
   Netname: BRAZIL-BLK2
   Netblock: 200.128.0.0 - 200.255.255.255
   Maintainer: RNP

   Coordinator:
      Gomide, Alberto Courrege  (ACG8-ARIN)  gomide@nic.br
      +55 19 9119-0304 (FAX) +55 19 9119-0304

   Domain System inverse mapping provided by:

   NS.DNS.BR                     143.108.23.2
   NS1.DNS.BR                    200.255.253.234
   NS2.DNS.BR                    200.19.119.99

CIDR:          200.204/16
ASN:           AS10429
ID abusos:     LUA72
entidade:      TELECOMUNICACOES DE SAO PAULO S/A - TELESP
documento:     002.558.157/0001-62
responsável:   Milton Kendi Ue
endereço:      Av. Brigadeiro Faria Lima, 1188, 5 andar
endereço:      01451-051 - Sao Paulo - SP
telefone:      (011) 3038-7253 []
```

Probability the Source Address was Spoofed:
There is a very strong possibility that the source address of this scan was spoofed. The attacker only initiated one part of the 3-way handshake. IP spoofing is strongly involved. The IP space belongs to known hackers so either the attacking system is real and has compromised a system on that network or the attacker is using a spoofed IP to carry out their scans.

Description of Attack:
In the past there was thought to be a Linuxconf buffer overflow vulnerability shipping with some RedHat 6.0 versions of Linux. Linuxconf was designed to help with remote administration of Linux systems. The vulnerability appeared to be in the way HTTP headers were handled by the program when remote administration was executed. In the case of the exploit, when "An attacker supplying excess data to the USER_AGENT field in vulnerable versions of Linuxconf. This data can overflow the relevant buffer, creating a stack overflow and, properly exploited, allowing remote execution of arbitrary code as root. Linuxconf 1.1.6r10 February 12, 2001" (http://www.securityfocus.com/bid/2352)

According to securityfous.com's website, the initial testing of the exploit did not reveal that the exploit code actually did not work. If this scan continues, this one will be one to watch for in the future.

Attack Mechanism:
This is definitely a stimulus from the attacker. The IP address may be spoofed or the attack came from a compromised host. Scanning for an unproven exploit is very

interesting. This scan could have been a good way to remotely scan this subnet for any Linux boxes. This could be a precurser for an even bigger attack. Knowing what is out there I am surprised that the attacker did not try to exploit x.x.42.66. This is a known Linux box with many common Linux services running.

Correlations:
This exploit is known but it is also known not to work. According to the following links the code does not do what is promised so why scan for it?

http://www.networkice.com/advice/exploits/ports/98/default.htm
http://oliver.efri.hr/~crv/security/bugs/Linux/lconf4.html
http://www.securityfocus.com/bid/2352

Evidence of Active Targeting:
This attacker is actively targeting any system running any Linuxconf services on that subnet. The final outcome of this scan is still to be determined. After checking other logs, I was unable to find similar patterns of scanning. This is one to watch.

Severity:
Target Criticality =1
Because no system was directly targeted by this sweep the criticality is low.

Attack Lethality = 1
This was reconnaissance and not active targeting so the threat is also low.

System Countermeasures = 2
Make sure the if there are any Linux systems out in front of the firewall they have Linuxconf services turned off for remote administration.

Network Countermeasures = 2
Continue to monitor all traffic directed to this subnet on port 98. Continually update all snort rules to detect this type of activity.

(Criticality + Lethality) – (System Countermeasures + Network Countermeasures) = Severity
(1+1) - (2+2) = 2

Defensive Recommendation:
Make sure that the Snort rules are current and look for any updates on bugtrack.

Multiple-Choice Test Question:
portscan.log:May 28 01:33:24 200.204.151.236 :1976 -> x.x.42.90:98 SYN ******S*

What service runs on port 98 and what exploit is it known for?

A. Finger, Firehotcker
B. XNS Mail, DMSetup

C. Linuxconf, Remote Administration
D. Metagram Relay, Hidden Port

Answer: C

Detect 5

ICMP Mobile Host Redirect

```
 [**] ICMP Mobile Host Redirect (Undefined Code!) [**]
05/07-18:47:22.705576 x.x.42.82 -> 144.42.0.80
ICMP TTL:127 TOS:0x0 ID:208 IpLen:20 DgmLen:669
Type:32  Code:54  UNKNOWN
20 64 39 38 39 32 39 65 31 32 30 63 38 33 39 64   d98929e120c839d
61 0A 36 36 20 36 31 20 33 34 20 36 36 20 36 36  a.66 61 34 66 66
20 33 37 20 33 34 20 33 32 20 36 31 20 36 31 20   37 34 32 61 61
36 36 20 36 33 20 33 42 20 32 30 20 34 45 20 35  66 63 3B 20 4E 5
33 20 20 66 61 34 66 66 37 34 32 61 61 66 63 3B  3  fa4ff742aafc;
20 4E 53 0A 34 33 20 35 30 20 35 46 20 35 35 20   NS.43 50 5F 55
35 33 20 34 35 20 35 32 20 35 46 20 34 43 20 34  53 45 52 5F 4C 4
46 20 34 37 20 34 39 20 34 45 20 33 31 20 35 46  F 47 49 4E 31 5F
20 34 45 20 20 43 50 5F 55 53 45 52 5F 4C 4F 47   4E  CP_USER_LOG
49 4E 31 5F 4E 0A 34 35 20 35 37 20 33 44 20 35  IN1_N.45 57 3D 5
33 20 34 38 20 34 31 20 33 31 20 33 44 20 46 31  3 48 41 31 3D F1
20 30 33 20 30 34 20 38 37 20 45 43 20 32 39 20   03 04 87 EC 29
41 30 20 30 38 20 20 45 57 3D 53 48 41 31 3D 2E  A0 08  EW=SHA1=.
2E 2E 2E 2E 29 2E 2E 0A 44 36 20 44 45 20 30 34  ....)...D6 DE 04
20 39 44 20 30 46 20 45 44 20 34 42 20 45 38 20   9D 0F ED 4B E8
32 46 20 45 42 20 34 31 20 44 46 20 35 42 20 32  2F EB 41 DF 5B 2
44 20 35 44 20 35 35 20 2E 2E 2E 2E 2E 2E 4B  D 5D 55  ......K
2E 2F 2E 41 2E 5B 2D 5D 55 0A 35 32 20 33 32 20  ./.A.[-]U.52 32
35 46 20 35 35 20 35 33 20 34 35 20 35 32 20 35  5F 55 53 45 52 5
46 20 34 39 20 34 34 20 33 44 20 35 33 20 37 35  F 49 44 3D 53 75
20 37 33 20 36 31 20 36 45 20 20 52 32 5F 55 53   73 61 6E  R2_US
45 52 5F 49 44 3D 53 75 73 61 6E 0A 34 43 20 36  ER_ID=Susan.4C 6
31 20 36 45 20 36 34 20 36 37 20 37 32 20 36 31  1 6E 64 67 72 61
20 36 36 20 36 36 20 35 42 20 32 44 20 35 44 20   66 66 5B 2D 5D
35 35 20 35 32 20 33 32 20 35 46 20 20 4C 61 6E  55 52 32 5F  Lan
64 67 72 61 66 66 5B 2D 5D 55 52 32 5F 0A 34 46  dgraff[-]UR2_.4F
20 34 43 20 34 34 20 35 46 20 35 35 20 35 33 20   4C 44 5F 55 53
34 35 20 35 32 20 35 46 20 34 39 20 34 34 20 33  45 52 5F 49 44 3
44 20 35 33 20 37 35 20 37 33 20 36 31 20 20 4F  D 53 75 73 61  O
4C 44 5F 55 53 45 52 5F 49 44 3D 53 75 73 61 0A  LD_USER_ID=Susa.
36 45 20 34 43 20 36 31 20 36 45 20 36 34 20 36  6E 4C 61 6E 64 6
37 20 37 32 75 73 26 2E 62 79 70 61 73 73 3D 26  7 72us&.bypass=&
2E 70 61 72 74 6E 65 72 3D 26 2E 75 3D 30 67 73  .partner=&.u=0gs
39 72 62 38 74 66 63 67 6C 33 26 2E 76 3D 30 26  9rb8tfcgl3&.v=0&
68 61 73 4D 75 73 26 2E 62 79 70 61 73 73 3D 26  hasMus&.bypass=&
2E 70 61 72 74 6E 65 72 3D 26 2E 75 3D 30 67 73  .partner=&.u=0gs
39 72 62 38 74 66 63 67 6C 33 26 2E 76 3D 30 26  9rb8tfcgl3&.v=0&
68 61 73 4D 73 67 72 3D 30 26 2E 63 68 6B 50 3D  hasMsgr=0&.chkP=
```

```
59 26 2E 64 6F 6E 65 3D 26 6C 6F 67 69 6E 3D 66  Y&.done=&login=k
75 6C 6C 65 72 6A 6D 26 70 61 73 73 77 64 3D 61  billsjm&passwd=a
6C 69 63 69 61                                   licia
```

Source of the Trace:
This trace comes from a sensor placed on my network between our border router and our Internet firewall.

Detect was Generated By:
This detect was generated by a Redhat Linux system running Snort 1.7 using the standard rule set from the Snort homepage. The specific rule, which captured the traffic, was:

```
alert icmp any any -> any any (msg:"ICMP Mobile Host Redirect"; itype:
32; icode: 0;)
```

Here is the Output from ARIN.

```
Independence Blue Cross (NET-IBC-NET)
   1901 Market Street, 6th Floor
   Philadelphia, PA 19103
   US

   Netname: IBC-NET
   Netblock: 144.42.0.0 - 144.42.255.255

   Coordinator:
      Eshbach, William  (WE49-ARIN)  william.eshbach@ibx.com
      (215) 241 - 4228 (FAX) (215) 241 - 4272

   Domain System inverse mapping provided by:

   NETSRV01.IBX.COM            144.42.100.2
   NETSRV02.IBX.COM            144.42.100.7

   Record last updated on 13-Mar-2001.
   Database last updated on 26-May-2001 22:57:19 EDT.
```

Probability the Source Address was Spoofed:
The IP address in question appears to be valid. This packet caught my eye due to the nature of the alert. Mobile ICMP, this one is new to me. After further review, the source of this stimulus appears to be from an already established connection from within my network.

Description of Attack:
I thought this was very interesting due to the nature of the Internet and mobile computing. At first I thought this was some kind of new attack circulating around the Internet. After careful review I now understand that this packet is nothing more than a mobile computer who has lost their way and the once established connection to a host on my network is looking for a route back to its host.

"In mobile environments, as computers move to unknown networks, they need to discover new service providers, applications, and other network resources. Since the

performance characteristics of such environments are often poor (due mainly to wireless communications and the restricted power of machines), mobile hosts require access to the nearest equivalent of some resource. On the other hand, services and applications located on the fixed part of the network may need to be aware of mobile host locations, in order to redirect messages, replies, references, files, displays, and so on." (Baggio, Piumarta, 1996) http://www-sor.inria.fr/publi/MHTRD_sigops96.html

Attack Mechanism:
This is definitely a response from an already established connection.

Correlations:
Oct 1996, RFC2002, http://www.faqs.org/rfcs/rfc2002.html, IP Mobility Support
Oct 1996, RFC 2005, http://www.faqs.org/rfcs/rfc2005.html, Applicability Statement for IP Mobility Support
Oct 1996, RFC2006, http://www.faqs.org/rfcs/rfc2006.html, The Definitions of Managed Objects for IP Mobility Supportusing SMIv2
http://www.lk.cs.ucla.edu/JWTSENG/ZIP/
http://www-sor.inria.fr/publi/MHTRD_sigops96.html
http://www.computer.org/internet/v2n1/perkins.htm

Evidence of Active Targeting:
There was no evidence of active targeting. If Port 434 (Mobile IP) was actively being targeted then there would be a case for suspicion.

Severity:
Target Criticality =1
This was a false positive. The system in question had already established the connection to a friendly system. This is not a critical system on the network

Attack Lethality = 1
No exploit was involved here.

System Countermeasures = 2
Continue to monitor similar traffic like this just in case mobile exploits become popular.

Network Countermeasures = 2
Continue utilizing IDS and checking all log activity for similar situations.

(Criticality + Lethality) – (System Countermeasures + Network Countermeasures) = Severity
(1+1) - (2+2) = 2

Defensive Recommendation:
Continue to monitor all sensors, logs, and make sure Snort logs are up-to-date.

Multiple-Choice Test Question:

Mobile devices communicate via IP on what port?

A. 434
B. 443
C. 89
D. 125

Answer: A

Assignment 2

The State of Intrusion Detection

Passive network mapping:

There are two methods of mapping networks available to hackers today - active and passive mapping. Both methods have their advantages and their disadvantages. Active mapping involves generating a predetermined order of IP packets to a host and analyzing its response. It is a stimulus-based mechanism from the attacker. Nmap (Stimulus-based) uses this method to "finger print" remote operating systems (OS). Active mapping also tends to be very fast and can be considered "noisy" at times. Stimulus-based mapping is very well known and with today's technology can be quickly identified with the use of any IDS system.

"Passive scanning is a response based technique, where one listens to a choke point in a network and uses the data they gather to map the entire domain for their own benefit" (Giovanni, 200). This type of technique requires more time for data gathering and analysis. The use of passive mapping appears to have come about due to the increase in domains putting up firewalls and installing Intrusion Detection Systems (IDS) to protect their assets from being attacked or compromised. This paper will discuss how the idea of passive mapping came about, the way it works, and the benefits of utilizing such a technique on a network to increase the overall security of the network.

How passive mapping came to be:

In an article by Coretez Giovanni, (2000), he writes about how "Intrusion Detection systems (IDS) are used to help defend domains by sitting on network choke points and recording all inbound and outbound packet traffic. These well-positioned tools have always been thought of as a defense weapon against cyber crime. But a tool has no say in

how it is used." His statement makes a strong case that there may be a good possibility that the next phase of hacking could be targeted at IDS systems. This argument would indeed make sense since most domains utilize IDS and firewalls as their first line of defense. Stimulus-based attack tools like Nmap are very well known and every IDS system can detect them. Today's hackers need better tools and a different approach to finding out information about their targeted domains. This article appears to have inspired a number of authors to write about passive mapping, and others to create tools as a proof of concept about this new technique. The Siphon project by Bind and Aempirel is one such tool that appears to have come out of Giovanni's article. The Siphon project web site has been removed from the Internet and trying to find a copy of Siphon to test was difficult. This does not mean the project has been abandoned; it may have moved underground temporarily.

How passive mapping works:

The key to understanding how passive mapping works, is being able to understand the difference between a traditional stimulus-based mapping and a response-based mapping. In stimulus-based mapping, the attacker knows nothing about the network they are attacking or what systems are located on that domain. The goal of the attacker is to find out as much information as they can about a particular domain through the use of stimulus tools like Nmap, Nessus and various other scanning tools that generate IP packets. These types of tools send pre-defined IP packets to the targeted domain. How the hosts and network systems respond to the active probe tells the attacker a lot of information about that network. The problem with IP based tools is again, many domains now have firewalls and IDS systems, which are set up to alert security analysts when such intrusive traffic hits their address space.

Passive network mapping takes a different approach over mapping by stimulus. Networks are mapped with the networks own inbound and outbound traffic patterns. Locations in and out of the domain are recorded along with source and destination ports. Recording the source and destination ports can give a very good description as to the types of services that the domain is offering and the frequency that the users are visiting sites on the Internet. Additional information like routing and spanning tree also help to determine exactly how large the network spans and over time, help to map the entire architecture of the domain (Nazario, 2000). Additionally, passive mapping is a very useful tool in finding and analyzing systems that might only be on for a matter of seconds while they transmit and receive data.

How it may be used in the future:

Due to the current state of intrusion detection and the growing popularity with domains putting up firewalls and IDS systems, hackers are going to have to take a more covert approach to data reconnaissance and exploitations to continue breaking into domains successfully. Utilizing such a technique as passive mapping could be the next generation

in hacking. Although unproven, there are a number of new programs out there, which now passively detect operating systems based on TCP/IP flag settings, sackOk options, nop options, and window scaling options. "Passive OS fingerprinting can be done on huge portions of input data - eg. information gathered on firewall, proxy, routing device or Internet server, without causing any network activity. You can launch passive OS detection software on such machine and leave it for days, weeks or months, collecting really interesting statistical and - *erm* - just interesting information (Zalewski, 2001). An example of the output of his code is listed below. It is evident, some OS types are still not recognizable. There is still much more work to be done in this area of research.
wwww:ttt:mmm:D:W:S:N:OS Description
wwww- window size, ttt – time to live ,mmm- maximum segment size,
D – don't fragment flag
W- window scaling
S- sackOK flag
N- nop flag
172.22.42.3: UNKNOWN [32768:64:36865:1:0:1:1].
172.22.42.2 [1 hops]: Digital UNIX V4.0E
172.22.42.5: UNKNOWN [32768:64:36865:1:0:1:1].
172.22.42.2 [4 hops]: Digital UNIX
172.22.42.27 [15 hops]: Windows NT 4.0 *
172.22.42.30 [15 hops]: Windows NT 4.0 *
172.22.42.73: UNKNOWN [44032:127:1360:1:-1:1:1].
172.22.42.28 [1 hops]: Windows NT 4.0 *
172.22.10.197 [2 hops]: Linux 2.2.14 or Cobalt Linux 2.2.12C3
172.22.10.5: UNKNOWN [32850:63:1460:1:1:1:1].- actually a Solaris 8 server

In Conclusion:

The use of passive mapping may be the new tool of choice by experienced hackers. It affords data collection with maximum stealth capabilities. Just by listening to the domain traffic, the hacker can gain both network- and user-based knowledge without triggering any IDS alarms. In the future, passive mapping will be highly automated and replace current active mapping tools preferred by today's hackers.

Additionally, passive mapping will perform "OS detection, but it has a different stimulus and therefore will be better at some mapping concepts then others. When defending a network, passive techniques are not just used for intrusion detection, but also in discovering unreported services and new systems to validate against the security posture" (Giovanni, 2000). As an offensive tool hackers will use passive mapping as a way to determine hidden vulnerabilities within a selected domain. "Passive mapping is a great way to determine the target network's defensive systems by detecting noisy security tools. For example, the determination of port 2998 indicates to an attacker that this is an IIS Real Secure system without needing to scan processes and ports. Also, passive mapping can be used to profile a network to determine acceptable use of protocols that

will allow exploit communications to mimic the technique and avoid threshold alarming (Giovanni, 2000).

References:

Giovanni, Coretez. “Passive Mapping: The Importance of Stimuli”2000 Available http://www.eurocompton.net/stick/papers/PassiveMappingviaStimulus.pdf

Giovanni, Coretez. “Passive Mapping: An Offensive Use of IDS” 2000 Available http://www.eurocompton.net/stick/papers/OffensiveUseofIDS.pdf

Nazario, Jose. Passive System fingerprinting using Network Client Applications, Nov 2000. Available http://groups.google.com/groups?ic=1&selm=bugtraq/Pine.BSO.4.21.0101171613030.9156-100000@spam.thegeekempire.net

Fyodor, The Art of Port scanning Sep 6 1997 Available http://www.insecure.org/nmap/nmap_doc.html

Fyodor, Fyodor,Max Vision,Marty Roesch,Edward Skoudis,Dragos Ruiu,Craig Smith Peter Grundl . Know Your Enemy: Passive Fingerprinting*IDing remote hosts, without them knowing* May 2000 Available http://project.honeynet.org/papers/finger/

Zalewski Michal, passive OS fingerprinting tool version 1.7 Readme, http://lcamtuf.hack.pl/p0f.tgz Available

Assignment 3

Analyse This

SnortA*.txt (6 files)
Snort Fast Alert file. Each alert provides a timestamp, alert message, source and destination IP addresses with ports details.

SnortS*.txt (9 files)
Snort Scan preprocessor files used to detect netork scans. Each line contained a timestamp, source and destination IP addresses with port details and details of the protocol and flags.

UMBCN*.txt (30 files)

Snort Scan detection preprocessor output files used to detect netork scans. Each line contained a timestamp, source and destination IP addresses with port details and details of the protocol and flags.

OOSche*.txt (12 files)
Snort Logs showing details of the protocol headers and payload of packets.
As advised, the data files are not complete due to power failures or lack of disk space.

Overall the network is considered very unsecure. My recommendation is to install a firewall and IDS sensors throughout the network to help protect assets from would-be hackers while providing more visibility into the network. Filtering known hacker ports at the router lever is also advisable. There are many cases of hosts on the network that either have been compromised or are in the process of being targeted for an attack. There are numerous hosts running gaming servers, music servers and Email relaying. In analyzing the supplied data, I was unable to come up with the rule set in which data was captured. The analyst process will be described at the end of this summary.

Table One

Earliest alert at **00:00:07**.303804 *on 01/30/2001*
Latest alert at **23:52:55**.217654 *on 02/11/2001*

Signature (click for sig info)	# Alerts	# Sources	# Destinations
Russia Dynamo - SANS Flash 28-jul-00	1	1	1
TCP SMTP Source Port traffic	4	4	3
SUNRPC highport access!	4	3	3
SNMP public access	5	2	1
NMAP TCP ping!	12	6	4
ICMP THE SOURCE and THE DESTINATION outside network	19	14	12
TCP THE SOURCE and THE DESTINATION outside network	60	17	31
Null scan!	72	68	47
Tiny Fragments - Possible Hostile Activity	111	17	9
WinGate 1080 Attempt	191	56	82
Queso fingerprint	210	30	52

Attempted Sun RPC high port access	507	4	4
connect to 515 from inside	590	5	4
SYN-FIN scan!	1112	5	1111
Watchlist 000220 IL-ISDNNET-990517	3702	12	17
Possible RAMEN server activity	3779	1140	2479
Watchlist 000222 NET-NCFC	5388	17	10
UDP THE SOURCE and THE DESTINATION outside network	148246	267	984

Table II – There were over 233,421 alerts during this time period of
Earliest alert at **00:00:04**.226757 *on 02/20/2001* Latest alert at **23:51:59**.244669 *on 03/10/2001*

Signature (click for sig info)	# Alerts	# Sources	# Destinations
connect to 515 from inside	1	1	1
Probable NMAP fingerprint attempt	1	1	1
Tiny Fragments - Possible Hostile Activity	2	2	2
Back Orifice	9	1	1
STATDX UDP attack	16	2	8
Attempted Sun RPC high port access	23	2	2
ICMP THE SOURCE and THE DESTINATION outside network	62	7	6
Null scan!	63	36	32
Queso fingerprint	203	31	49
SUNRPC highport access!	204	3	3
WinGate 1080 Attempt	274	40	62
Possible RAMEN server activity	539	139	288
Watchlist 000222 NET-NCFC	608	5	5
SMB Name Wildcard	689	229	310
SNMP public access	889	2	5

TCP THE SOURCE and THE DESTINATION outside network	2347	33	54
External RPC call	3024	2	1461
NMAP TCP ping!	5550	7	2488
SYN-FIN scan!	9338	3	8682
Watchlist 000220 IL-ISDNNET-990517	10549	30	43
UDP THE SOURCE and THE DESTINATION outside network	199030	356	911

The descriptions below are correlated with the two charts above. Both charts were generated using SnortSnarf from Silicon Defense.

Below are descriptions of the above attacks.

Russia Dynamo - SANS Flash 28-jul-00-
172.22.203.50:6346-> 194.87.6.79:1791
This alert was caused from traffic being generated from a node on the network to a computer located in Russia. Not having the orginal 'rule' makes it hard to examine this incident further. The source port of the system in question is using a port commonly known for Gnutella. Gnutella is a fully distributed information-sharing technology used to distribute software anonymously.
For more information see
http://www.gnutellanews.com/information/what_is_gnutella.shtml

TCP SMTP Source Port traffic
IP addresses with SMTP Source Ports
200.251.185.3- RNP (Brazilian Research Network)
17.135.218.56- Apple Computer, Inc
11.125.218.156- DoD Intel Information Systems
195.211.49.18- JIPPII-LAXIN-DE Laxin.de ShellServices DE
Destination of SMTP traffic on the network.
172.22.60.17
172.22.158.238
172.22.139.54

SUNRPC highport access!
The following systems on the network have SUNRPC Highport access activity on port 32771 and should be examined.

SUNRPC highport access! [**] 205.188.5.157:5190 (America Online, Inc) -> 172.22.98.227:32771

SUNRPC highport access! [**] 200.233.81.13:13765 RNP (Brazilian Research Network)-> 172.22.60.17:32771
SUNRPC highport access! [**] 24.9.203.188:61207(@Home Network)-> 172.22.165.129:32771

Note: The @ Home attacker also had the following attacks against the following systems
WinGate 1080 Attempt [**] 24.9.203.188:64450-> 172.22.165.129:1080
Null scan! [**] 24.9.203.188:63602-> 172.22.165.129:427
Port 427 – The attacker was checking to see if this system is a SCO Unix server. There is a known vulnerability under SCO Openserver, which allows any file, which is group writeable by the 'auth' group to become writable to the world. Both the /etc/passwd and /etc/shadow files fall into this category. If this system is not SCO based then the system is not vulnerable. For further reference: http://www.securityfocus.com/bid/701.html

SNMP public access
One internal system appears to be generating SNMP traffic destined for two other systems within the network. This may be a false positive or a misconfigured SNMP trap. The system that should have its SNMP configuation checked should be the following destinations.
SNMP public access [**] 172.22.70.42:2155-> 172.22.50.154:161
SNMP public access [**] 172.22.111.156:1737-> 172.22.50.154:161
SNMP public access [**] 128.46.156.197:1191-> 172.22.100.99:161
SNMP public access [**] 128.46.156.197:1200-> 172.22.100.206:161
SNMP public access [**] 128.46.156.197:1251-> 172.22.100.143:161
SNMP public access [**] 128.183.38.30:1030-> 172.22.154.26:161
SNMP public access [**] 128.46.156.197:1160-> 172.22.100.45:161

NMAP TCP ping!
Nmap TCP Ping is a way of checking to see if a host is available. Typically Nmap is used by hackers to map networks. The following IP addresses were probed using Nmap.
NMAP TCP ping! [**] 63.119.91.2:80 (UUNET Technologies, Inc) -> 172.22.1.3:53
NMAP TCP ping! [**] 63.119.91.2:80(UUNET Technologies, Inc -> 172.22.110.39:25
NMAP TCP ping! [**] 192.102.197.234:53(Intel Corporation) -> 172.22.1.8:53
NMAP TCP ping! [**] 2.2.2.2:80(Information Sciences Institute University of Southern California)-> 172.22.1.5:53
NMAP TCP ping! [**] 12.40.36.194:80(AT&T ITS)-> 172.22.1.5:53
NMAP TCP ping! [**] 194.133.58.129:80(European Regional Internet Registry/RIPE NCC)-> 172.22.1.5:53
NMAP TCP ping! [**] 208.5.219.131:53(Sprint)-> 172.22.1.8:53
You can also find more information at:
For more information go to http://www.nmap.org
CVE: CAN-1999-0523
http://www.whitehats.com/info/IDS28
http://advice.networkice.com/Advice/Intrusions/2000310/default.htm

Null scan! – 68 sources, 47 destinations
Packets with no flags set. This is a scan where the attacker is trying to avoid detection while looking for open ports on remote systems.
Top 4 sources of this attack signature and the ports that have been targeted
Null scan! [**] 24.180.66.185:1121 (@Home Network)-> 172.22.201.234:900 OMG Initial Refs (TCP/UDP)
Null scan! [**] 128.40.224.18:4141(University College London)-> 172.22.211.74:6346 **GNUtella scan**
Null scan! [**] 24.17.73.154:1592(@Home Network)-> 172.22.211.74:6346 **GNUtella scan!**
Null scan! [**] 24.9.203.188:63602(@Home Network)-> 172.22.165.129:427 SCO OpenServer 5.0.5 'userOsa' symlink Vulnerability

Tiny Fragments - Possible Hostile Activity- This probe is a port scan trying not to be detected by IDS or other network monitoring techniques. Here, it appears the attacker is mapping the network.
Tiny Fragments - Possible Hostile Activity 17 sources 9 destinations
Top Source addresses:
Tiny Fragments - Possible Hostile Activity [**] 64.80.90.36(PaeTec Communications, Inc.) -> 172.22.98.117 (53 occurances)
Tiny Fragments - Possible Hostile Activity [**] 64.80.90.36(PaeTec Communications, Inc.)-> 172.22.97.231 (20 occurances)
Tiny Fragments - Possible Hostile Activity [**] 202.205.5.10(Asia Pacific Network Information Center) -> 172.22.1.8
Tiny Fragments - Possible Hostile Activity [**] 202.96.96.3(CHINANET Zhejiang province network) -> 172.22.1.10
Tiny Fragments - Possible Hostile Activity [**] 202.96.96.3 (CHINANET Zhejiang province network) -> 172.22.1.8

WinGate 1080 Attempt 56 sources, 82 destinations These systems should be checked to see if they have been compromised.
SOCKS (Port 1080) is a firewall tunneling service. By design, it allows many machines behind a firewall to access the Internet without actually being on the Internet. In theory, SOCKS should only be visible from the internal side of the server, and not from the Internet. Hackers will frequently probe to see if SOCKS is visible from the other side. A common attack technique is to install "telnet redirectors" on a system they have compromised. This allows them to telnet to the redirector and then telnet out from there anonymously, masking their true point of origin.
WinGate's Winsock redirector service is susceptible to a buffer overflow vulnerability that will crash all WinGate services. May 2001, http://www.securityfocus.com/bid/509.html,
Top 5 Source IP Addresses triggering this event
WinGate 1080 Attempt [**] 24.1.201.200:1606 (@Home Network)-> 172.22.221.30:1080 29 instances of this attack from this IP address!

WinGate 1080 Attempt [**] 128.121.244.217:1632 (Verio, Inc.)-> 172.22.15.178:1080
29 instances of this attack from this IP address!
WinGate 1080 Attempt [**] 199.173.178.2:2892 (UUNET Technologies, Inc)->
172.22.209.234:1080 18 instances of this attack from this IP address!
WinGate 1080 Attempt [**] 216.179.0.32:2020 (BestWeb Corporation)->
172.22.222.178:1080 15 instances of this attack from this IP address!
WinGate 1080 Attempt [**] 63.151.165.130:4473 (Creative Internet Techniques)->
172.22.98.118:1080

Queso fingerprint- 30 sources, 52 destinations
The attacker is using a tool called Queso to determine the OS of the target systems. There are 16 different source IP addresses from the same source address space (TU Dresden Universitaetsrechenzentrum) associated with this scan. The rest of the source addresses vary. The source addresses appear to be looking for destination ports associated with the Commonly used Gnutella port
Queso fingerprint [**] 141.30.228.134:3625(TU Dresden Universitaetsrechenzentrum)->
172.22.224.242:6355 29 instances
Queso fingerprint [**] 141.30.228.43:2266(TU Dresden Universitaetsrechenzentrum)->
172.22.229.22:6346 23 instances

TOP Destinations receiving this attack signature
Queso fingerprint [**] 141.30.228.199:3435-> 172.22.203.50:6346
Queso fingerprint [**] 141.30.228.134:2287-> 172.22.206.30:6346
Queso fingerprint [**] 141.30.228.222:2614-> 172.22.211.74:6346
Note: This destination address 172.22.211.74 also has, and should be checked out for possible compromises
1 instance of SYN-FIN scan!
9 instances of Null scan!
19 instances of Queso fingerprint
133 instances of Watchlist 000220 IL-ISDNNET-990517
Attempted Sun RPC high port access 4 sources 4 destinations

Attempted Sun RPC high port access [**] 64.244.10.40:7777 (Business Internet, Inc)-> 172.22.223.254:32771
Although this alert appears to be an attempted SUN RPC high port access, the alert really is a response from a napster server allowing a client on the network to listen to music on their system

Attempted Sun RPC high port access [**] 205.188.153.97:4000 (America Online, Inc)**-> 172.22.221.246:32771**
This trace could actually be a number of things. According to http://www.networkice.com/Advice/Exploits/Ports/32771/default.htm port 4000 is known for Ghost Portmapper. Some SunOS machines listen at this port for portmapper. Since firewalls frequently don't filter at high ports, it can allow the attacker access to portmapper even when port 111 is blocked.

Due to the nature of the connection (AOL) this trace could actually be an ICQ session taking place or VDOPhone, Intel Internet phone connection.
However according to bugtrack,
http://advice.networkice.com/Advice/Exploits/Ports/groups/NAT/default.htm this port is listed as a commonly used port for Network Address Translation gateways. Also listed as a UDP port used for the Command and Conquer multiplayer game
May 2001, http://advice.networkice.com/advice/exploits/ports/4000/default.htm, (UDP)
The game "Command and Conquer" by Westwood Studios uses this UDP port.

Connect to 515 from inside 5 sources 4 destinations
All of these systems need to be checked to see if they have the Lion worm or Ramen toolkit installed on their systems.
Port 515 is typically a BSD LPD or print spooler; however there are a number of exploits associated with this port.
Aug 1990, RFC1179, http://www.faqs.org/rfcs/rfc1179.html, Line Printer Daemon Protocol
Jan 2001, CERT/CC, http://www.cert.org/incident_notes/IN-2001-01.html, Widespread compromises via "ramen" toolkit. (TCP)
According to Incidents, Apr 2001: "...and 515/tcp (RedHat 7.0 lpd exploit). Ramen and Lion also use these ports to spread their exploits around. There is a very strong possibility that if this system is a Linux box, it has been infected with the Ramen or Lion worm and must be further examined.

514 instances of this connection to 515 from inside the network
connect to 515 from inside [**] 172.22.98.190:1025-> 216.181.129.185:515
connect to 515 from inside [**] 172.22.97.88:1025-> 216.181.129.185:515
connect to 515 from inside [**] 172.22.7.20:22-> 216.88.97.58:515
connect to 515 from inside [**] 172.22.201.170:2697-> 209.50.66.2:515
connect to 515 from inside [**] 172.22.162.71:2878-> 209.249.182.79:515

SYN-FIN scan! (1112 alerts) 5 sources 1111 destinations

Korea Network Information Center KR Scanned the entire Class B addresses looking for any DNS server.
Feb 2001 (Nov 2000), CERT/CC, http://www.cert.org/advisories/CA-2000-20.html
Multiple Denial-of-Service Problems in Internet Software Consortium (ISC) BIND
The first vulnerability is referred to by the ISC as the "zxfr bug" and affects ISC BIND version 8.2.2, patch levels 1 through 6. The second vulnerability, the "srv bug", affects ISC BIND versions 8.2 through 8.2.2-P6. Derivatives of the above code sets should also be presumed vulnerable unless proven otherwise.
SYN-FIN scan! [**] 211.248.112.67:53-> 172.22.1.130:53
SYN-FIN scan! [**] 211.248.112.67:53-> 172.22.254.215:53

Adelphia Cable Communications is looking for a DBStar service
[**] SYN-FIN scan! [**] 24.50.25.5:6699-> 172.22.211.122:1415

Splitrock Services, Inc Scan for http protocol over TLS/SSL
The known vulnerabilities: Certain versions of Network Associates Inc.'s Net Tools PKI (Public Key Infrastructure) server ship with a vulnerability which allows remote attackers to read any file in the system which the PKI server resides
http://www.securityfocus.com/bid/1537.html,
[**] SYN-FIN scan! [**] 63.252.15.242:2754-> 172.22.5.29:443

BBN Planet
This scan is looking for gnutella-server, Information can be found at
http://gnutella.wego.com
[**]SYN-FIN scan! [**] 4.35.4.244:1837-> 172.22.211.74:6346

Splitrock Services, Inc could be looking for a FW-1 firewall to see if they can exploit a known port, which by default is open.
[**] SYN-FIN scan! [**] 209.255.180.130:32808-> 172.22.5.29:259

Watchlist 000220 IL-ISDNNET-990517 12 sources 17 destinations
All 12 sources systems are from Iserial and appear to be sharing music via Napster related servers. Below are the three biggest systems that appear to be serving music for other people's enjoyment.

[**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.21.179:1172-> 172.22.207.226:6699
May 2001, http://advice.networkice.com/advice/exploits/ports/6699/default.htm, A program called "napster" for exchanging MP3 files defaults to this port.
2,186 instances of this alert. There is a very strong suspicion that this system is being used as a Napster server.

[**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.42.21:6699 (European Regional Internet Registry/RIPE NCC)-> 172.22.222.94:2609
321 instances of this alert from the source site to a system on the network. This alert also relates to Napster. This alert is actually a response from a Napster server to a host inside the network.

[**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.79.2:29459-> 172.22.97.30:4116
There are over 277 instances of *Watchlist 000220 IL-ISDNNET-990517* that are yet to be identified. The source address comes from Iserial. The port range from the source to its destination indicates multiple User Systems running VRML protocol. This traffic needs further analysis before any definite answer can be determined whether this is friendly or not.
212.179.79.0 - 212.179.79.63
netname: CREOSCITEX

descr: CREOSCITEX-SIFRA
country: IL

Destination supplying Napster music services and/or listening to Napster music.

172.22.207.226	172.22.204.78
172.22.222.94	172.22.97.30
172.22.204.22	172.22.206.94
172.22.224.34	172.22.97.62
172.22.217.98	172.22.221.114
172.22.225.186	172.22.201.242
172.22.211.74	172.22.221.162
172.22.204.78	172.22.60.17
172.22.97.30	172.22.224.126
172.22.206.94	172.22.98.185
172.22.97.62	

Possible RAMEN server activity-
This alert was the highest during the periods of **00:23:15**.036525 *on 01/30/2001 and 02/11/200*
There were over 3,779 alerts with this signature, 1,140 source addresses and 2,479 destinations that need further review.
What is in question is the type of traffic which is triggering all of the alerts. There are two types, "Quake-based games (e.g. Half-Life, Quakeworld, QuakeIII, etc.) that use numerous ports in the 26000-28000 range."http://advice.networkice.com/Advice/Exploits/Ports/26000/default.htm and Trojans SubSeven v2.1, Source: http://www.sans.org/y2k/subseven.htm (TCP)
This is just one of the alerts that attracted my attention.

Possible RAMEN server activity [**] 24.48.226.183:1580 (Adelphia Cable Communications)-> 172.22.1.37:27374
The source address of this computer scanned the entire class B address of the network looking for either the Quake-game or SubSeven 2.1 trojan.

Watchlist 000222 NET-NCFC
During this month of monitoring there were 17 sources and 10 destination hosts that triggered this alert.

[**] Watchlist 000222 NET-NCFC [**] 159.226.81.1:3762-> 172.22.6.47:25
The largest to stand out with over 5,362 instances of *Watchlist 000222 NET-NCFC* . The source address of this attack is registered to the following location.
The Computer Network Center Chinese Academy of Sciences (NET-NCFC)
P.O. Box 2704-10,
Institute of Computing Technology Chinese Academy of Sciences
Beijing 100080, China CN
It appears that the Chinese are using the Email server to relay all of their Email to various sites throughout the world. Typically, allowing public relaying is not good practice. Spammers and various other methods of Email fraud are performed this way. This needs to be corrected as soon as possible. The other 16 remote hosts are also using the Email server to relay their Spam mail.
The following IP address on the network allow SNMP relaying from anywhere in the world.

172.22.6.47	172.22.6.34
172.22.253.43	172.22.6.7
172.22.60.17	172.22.145.9
172.22.6.35	172.22.253.42
172.22.100.230	172.22.253.51

STATDX UDP attack-
According to Whitehats.com, this attack is targeted against Red Hat Linux 6.0 systems. It is evident from the signature below, the source address was doing rcp calls to over 1,230 distinct destination IPs on the network. Not allowing RCP calls from the Internet is highly recommended to prevent further instances of this type of remote mapping of the network.

Reference:
CVE CVE-2000-0666
Bugtrack 1480
Advice 2001702

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

```
Netname: NETBLK-SUNET
Netblock: 171.64.0.0 - 171.67.255.255
```

```
02/20-19:41:07.758966 [**] External RPC call [**] 171.65.61.201:2214->
```

172.22.4.0:111
02/20-19:41:07.758966 [**] External RPC call [**] 171.65.61.201:2214-> 172.22.4.0:111
02/20-19:41:07.759014 [**] External RPC call [**] 171.65.61.201:2215-> 172.22.4.1:111
02/20-19:41:07.759014 [**] External RPC call [**] 171.65.61.201:2215-> 172.22.4.1:111
02/20-19:41:07.760502 [**] External RPC call [**] 171.65.61.201:2226-> 172.22.4.12:111
02/20-19:41:07.760502 [**] External RPC call [**] 171.65.61.201:2226-> 172.22.4.12:111
02/20-19:41:07.760553 [**] External RPC call [**] 171.65.61.201:2227-> 172.22.4.13:111
02/20-19:41:07.760553 [**] External RPC call [**] 171.65.61.201:2227-> 172.22.4.13:111
02/20-19:41:07.761569 [**] External RPC call [**] 171.65.61.201:2241-> 172.22.4.27:111
02/20-19:41:07.761569 [**] External RPC call [**] 171.65.61.201:2241-> 172.22.4.27:111
02/20-19:41:07.761622 [**] External RPC call [**] 171.65.61.201:2242-> 172.22.4.28:111
02/20-19:41:07.761622 [**] External RPC call [**] 171.65.61.201:2242-> 172.22.4.28:111
02/20-19:41:07.763084 [**] External RPC call [**] 171.65.61.201:2256-> 172.22.4.42:111

SMB Name Wildcard (229 sources and 310 destinations)

SMB Name Wildcards are used by Microsoft systems to request remote machine netbios names. Typically computer names and MS client information is gathered by hackers or this can be nothing more than normal Microsoft traffic. If the network has Microsoft NT/98.95 systems this is a typical traffic pattern. After careful analysis of the traffic patterns, it is strongly recommended that some type of configuration management be put into place to make sure that all of the MS systems are up-to-date with the latest security patches and hot fixes and that all unnecessary services be turned off.

Back Orifice- Basically, Back Orifice works as a client-server program, with the intruder controlling the client. Once the Trojan horse is on the user's system, the client (which may be running anywhere on the Internet) can access the affected system with the

privileges of the user who inadvertently installed it. http://www.cert.org/vul_notes/VN-98.07.backorifice.html
The following hosts appear to be infected with this remote control Trojan and should be examined immediately.

The source of this attack comes from :
UUNET Technologies, Inc. (NETBLK-NETBLK-UUNET97DU)
3060 Williams Drive, Suite 601
Fairfax, VA 22031

02/24-17:04:09.754841 [**] Back Orifice [**] 63.10.224.59:2382-> 172.22.97.3:31337
02/24-17:04:16.714295 [**] Back Orifice [**] 63.10.224.59:2382-> 172.22.97.119:31337
02/24-17:04:19.102521 [**] Back Orifice [**] 63.10.224.59:2382-> 172.22.97.162:31337
02/24-17:04:22.457194 [**] Back Orifice [**] 63.10.224.59:2382-> 172.22.97.225:31337
02/24-17:04:24.335687 [**] Back Orifice [**] 63.10.224.59:2382-> 172.22.98.3:31337
02/24-17:04:25.359418 [**] Back Orifice [**] 63.10.224.59:2382-> 172.22.98.28:31337
02/24-17:04:27.815284 [**] Back Orifice [**] 63.10.224.59:2382-> 172.22.98.75:31337
02/24-17:04:30.711389 [**] Back Orifice [**] 63.10.224.59:2382-> 172.22.98.123:31337
02/24-17:04:36.800828 [**] Back Orifice [**] 63.10.224.59:2382-> 172.22.98.238:31337

The Top 20 source and destination addresses for the time period of:

Earliest alert at **00:00:07**.303804 *on 01/30/2001*
Latest alert at **23:52:55**.217654 *on 02/11/2001*

The top 20 Source IP addresses

1. The Source IP 172.22.218.90 appears 34496 times
2. The Source IP 172.22.150.220 appears 17804 times
3. The Source IP 172.22.204.66 appears 14252 times
4. The Source IP 172.22.202.50 appears 14003 times
5. The Source IP 172.22.150.133 appears 10408 times
6. The Source IP 172.22.228.54 appears 10098 times
7. The Source IP 206.112.192.106 appears 9992 times
8. The Source IP 172.22.212.206 appears 9898 times
9. The Source IP 172.22.210.250 appears 9679 times
10. The Source IP 172.22.203.234 appears 7075 times
11. The Source IP 172.22.217.142 appears 7021 times
12. The Source IP 172.22.209.238 appears 7006 times

13. The Source IP 172.22.217.58 appears 6990 times
14. The Source IP 172.22.150.143 appears 6782 times
15. The Source IP 172.22.206.78 appears 6743 times
16. The Source IP 172.22.150.225 appears 6557 times
17. The Source IP 172.22.98.176 appears 6160 times
18. The Source IP 172.22.225.198 appears 6090 times
19. The Source IP 172.22.224.238 appears 6068 times
20. The Source IP 172.22.100.230 appears 5725 times

Top 20 Destination IP Addresses

1. The Destination IP 172.22.160.109 appears 9992 times
2. The Destination IP 216.19.133.116 appears 2041 times
3. The Destination IP 172.132.71.130 appears 2012 times
4. The Destination IP 24.91.199.203 appears 1833 times
5. The Destination IP 63.21.61.147 appears 1729 times
6. The Destination IP 172.141.108.212 appears 1636 times
7. The Destination IP 172.169.147.76 appears 1580 times
8. The Destination IP 142.103.36.176 appears 1533 times
9. The Destination IP 66.24.125.138 appears 1489 times
10. The Destination IP 24.19.99.230 appears 1450 times
11. The Destination IP 63.14.172.15 appears 1425 times
12. The Destination IP 194.251.249.182 appears 1414 times
13. The Destination IP 66.30.167.225 appears 1401 times
14. The Destination IP 142.177.198.96 appears 1365 times
15. The Destination IP 24.6.245.220 appears 1324 times
16. The Destination IP 24.113.23.115 appears 1309 times
17. The Destination IP 24.183.99.210 appears 1299 times
18. The Destination IP 24.181.62.57 appears 1292 times
19. The Destination IP 65.33.209.215 appears 1153 times
20. The Destination IP 199.17.65.223 appears 1144 times

These were the top TCP flag based attacks which were found in the SnortS*.txt (9 files) Snort Scan preprocessor files were used to detect network scans. Each line contains a summary of the number of attacks, how many sources used this type of attack, and destinations that received them.

	# Alerts	# Sources	# Destinations
UDP scan	**454374**	**692**	**82747**
TCP **S***** scan	**54114**	**200**	**13873**
TCP **SF**** scan	17114	13	16246
TCP 21S***** scan	**780**	**124**	**133**

As for the Out of speck data:
There were 4287 instances where the SF flags were set along with a number of other abnormal flags.
grep SF OOS*.txt |wc

There were 43637 instances of just a Syn flag set
grep S***** OOS* |wc

There were grep 401 instances of port 6346 indicating a gnutella-svc
OOS*.txt |wc

45 Instances of a static TTL: 241
TCP sequence number the same.
02/12-04:57:50.433457 194.217.242.35:30975 -> 172.22.253.24:20
TCP TTL:241 TOS:0x0 ID:44050 DF
21SFRPAU Seq: 0x78FF0014 Ack: 0x78FF0014 Win: 0x14
TCP Options => EOL EOL EOL EOL EOL EOL SackOK

52 Instances of a static TTL:114
02/12-05:08:18.545748 194.222.96.40:30973 -> 172.22.60.14:20
TCP TTL:114 TOS:0x0 ID:59142 DF
21*FRPAU Seq: 0x78FD0014 Ack: 0x78FD0014 Win: 0x14
TCP Options => EOL EOL EOL EOL EOL EOL

Analysis Process

I began my analysis process by first building a Redhat Linux 7.0 system running on a P550 with 256MB of RAM and running SnortSnarf Versions v052001.1. Thinking this would be capable of compiling the data, I combined all of the alert files into one file and then ran SnortSnarf. Like many other students, I ran into memory and format problems. I decided after many attempts to move to a bigger, faster system. We happened to have a backup Compaq proliant server with dual 933Mhz, 2GB of RAM and lots of disk space to help compile this data. I once again tried to compile all of the data but once again, it failed. I then realized I needed to configure the data into a format that SnortSnarf would be able to read. I wrote a script (listed below) that would do a search and replace on every instance of MY.NET and change it to a 172.22 address. I also had to remove by hand all of the headers in all of the files. Once this process was complete I was able to compile all of the Ufiles and Afiles. What once took 2 days now took less than 1 hour. The rest of the files that contained application layer data and for some reason, all of the OOS files, would have to be done by a manual process of grep, awk and word count (wc).

Here is the script to change all instances of MY.NET to a reserved address subnet. However, before running this script I had to go in and manually remove the headers within all of the files. I used VI to do this. Then I ran the script and converted all of the MY.NET references to 172.22 references.

```
#!/bin/sh
for u in 'UMB*'
do
cat $u |sed 's/MY.NET/172.22/g'>$u.txt
done
```

I also did the same kind of thing for the Afiles, and Sfiles

```
#!/bin/sh
for a in 'SnortA*'
do
cat $a |sed 's/MY.NET/172.22/g'>$a.txt
done
```

This was how I ran Snortsnarf and compiled the data.

After changing all the MY.NET to a more Snortsnarf friendly format I wrote the following script to compile the data.

```
#!/bin/sh
#
#
# This scrip was written by David Sarmanian to compile the snort logs
for my practicle SANS exam

# The file to be processed
#snortlog=Ufiles
log1=UMBCNI3.txt
log2=UMBCNI4.txt
log3=UMBCNI5.txt
log4=UMBCNI2.txt
log5=UMBCNI60.txt
log6=UMBCNI61.txt
log7=UMBCNI25.txt
log8=UMBCNI27.txt
log9=UMBCNI31.txt
log10=UMBCNI30.txt
log11=UMBCNI35.txt
log12=UMBCNI58.txt
log13=UMBCNI59.txt
log14=UMBCNI57.txt
log15=UMBCNI54.txt
log16=UMBCNI55.txt
log17=UMBCNI52.txt
log18=UMBCNI53.txt
log19=UMBCNI51.txt
log20=UMBCNI46.txt
log21=UMBCNI47.txt
log22=UMBCNI44.txt
```

```
log23=UMBCNI43.txt
log24=UMBCNI28.txt
log25=UMBCNI26.txt

# Now go to the data and run the snortsnarf.pl command to process the
data
cd /var/www/html/snort/ufiles
echo "Compiling the data"
/usr/local/bin/snortsnarf.pl -d compall $log1 $log2 $log3 $log4 $log5
$log6 $log7 $log8 $log9 $log10 $log11 $log12 $log13 $log14 $log15
$log16 $log17 $log18 $log19 $log21 $log21 $log22 $log23 $log24 $log25

echo "The data has been compiled"
# End of File
```

This script compiled all of the UMBCNI logs and produced table 2. I also wrote similar scripts for Table one and the Afiles.

For the rest of the data I used grep, sort, awk and wc to find out how many instances of FIN-SYN and some of the other abnormal flags were set in the SnortSfiles. I did a sort and grep with wc to format the data. Then I imported all of it into Wxcel where I added up all of the instances of each type of TCP flag.

I did the same type of calculations for the top source and destination addresses.
As a reference port, I used some of the same strategies that my references used when they processed their data. Primarily similar grep and sort commands to make sure my calculations were correct.

References:

Bayerkohler, Marc. SANS Intrusion Detection Practical
http://www.sans.org/y2k/practicle/marc_bayerkohler_GCIA.html

Asadoorian, Paul. Intrusion Detection In Depth
http://www.sans.org/y2k/practical/Paul_Asadoorian_GIAC.doc

Bell, Mike. GCIA Practical
http://www.sans.org/y2k/practical/Mike_Bell_GCIA.doc

Last Updated: July 21st, 2017

Upcoming Training

SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
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 Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 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 Fall 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 Berlin 2017	Berlin, Germany	Oct 23, 2017 - Oct 28, 2017	Live Event
SANS Seattle 2017	Seattle, WA	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
SANS San Diego 2017	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Paris November 2017	Paris, France	Nov 13, 2017 - Nov 18, 2017	Live Event
Community SANS Pensacola SEC503	Pensacola, FL	Nov 27, 2017 - Dec 02, 2017	Community SANS
SIEM & Tactical Analytics Summit & Training	Scottsdale, AZ	Nov 28, 2017 - Dec 05, 2017	Live Event
SANS Cyber Defense Initiative 2017	Washington, DC	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS Security East 2018	New Orleans, LA	Jan 08, 2018 - Jan 13, 2018	Live Event
SANS Las Vegas 2018	Las Vegas, NV	Jan 28, 2018 - Feb 02, 2018	Live Event
SANS Dallas 2018	Dallas, TX	Feb 19, 2018 - Feb 24, 2018	Live Event
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced