Global Information Assurance Certification Paper

Intrusion Detection In Depth

GCIA Practical Assignment

Version 3.0

Matthew Fiddler

Assignment 1 – Describe the State of Intrusion Detection

Passive OS Fingerprinting using p0f

Understanding Passive OS Fingerprinting:
Passive OS detection is a method of attempting to identify a remote hosts's operating system using a set of unique fingerprints. These fingerprints are captured promiscuously on a network and analyzed against a database of known fingerprints. While not 100% accurate, passive OS detection can give a good indication of host OS types coming into your network, guests and attackers alike.

p0f – The Tool
p0f, written by Michal Zalewski, is a Linux command-line tool that is run promiscuously on the network. p0f works by comparing SYN packets against it's own internal database to attempt to determine the remote host OS type. Additionally p0f uses TTL values and OS predictions to estimate the proximity of the host (in number of hops). While there are many other OS detection tools available, p0f is unique in that it is totally silent, p0f does not need to generate a single packet.

p0f – The Fingerprints:
p0f uses a set of known TCP/IP flags as it's fingerprint database and with the help of BUGTRAQ contributions the database has currently grown to 136 unique OS fingerprints.

TCP/ IP Flags used by p0f for Passive OS Detection
Below is the format used internally by p0f. Briefly I will describe the unique fields used by p0f for OS Detection/Fingerprinting:

```
# Format:
#
# wwww:ttt:mmm:D:W:S:N:I:OS Description
#
# wwww - window size
# ttt  - time to live
# mmm  - maximum segment size
# D    - don't fragment flag  (0=unset, 1=set)
# W    - window scaling (-1=not present, other=value)
# S    - sackOK flag (0=unset, 1=set)
# N    - nop flag (0=unset, 1=set)
# I    - packet size (-1 = irrevelant)
#
```

Window size:
The buffer size that indicates the amount of bytes that can be transmitted before an ACK is needed.

Sample Default OS Window Sizes:

Winkdow Size	Host OS
5840	Linux 2.4.1-14
8760	Solaris 2.6 or 2.7

24820	SunOS 5.8
32120	Linux 2.2.x
32768	Mac OS X

Time To Live:
The time to live (TTL) value is set initially set by the operating system. As packets traverse the internet, their TTL value is decremented by 1 after each router hop. TTL's can help to identify a host by adding another unique value to the fingerprint. Additionally, since p0f is determining OS types by SYN packets, we can also use the TTL to determine the number of hops we are away from this host.

Sample Default OS TTL's:

TTL	Host OS
255	Solaris 2.6 or 2.7, Cisco IOS
128	Windows 2000
64	Linux 2.0, 2.2, 2.4
32	Windows CE 3.0 (Ipaq 3670)

For example, the below trace is an initial SYN packet destined for a FTP server. Assuming an initial TTL of 128 we can that this host is 5 hops away from us. While this can be confirmed with a traceroute back to the host, doing so would not be passive and would potentially alert a hostile remote host of our interest in them.

```
01/18-13:20:26.423134 xxx.xxx.xxx.4:15502 -> 64.245.58.81:21
TCP TTL:124 TOS:0x0 ID:41396 IpLen:20 DgmLen:44 DF
******S* Seq: 0x143967  Ack: 0x0  Win: 0x2000  TcpLen: 24
TCP Options (1) => MSS: 1460
0x0000: 00 D0 79 74 38 00 08 00 20 B0 1A B6 08 00 45 00  ..yt8... .....E.
0x0010: 00 2C A1 B4 40 00 7C 06 B5 EE C7 DE 64 04 40 F5  .,..@.|.....d.@.
0x0020: 3A 51 3C 8E 00 15 00 14 39 67 00 00 00 00 60 02  :Q<.....9g....`.
0x0030: 20 00 5A DF 00 00 02 04 05 B4 00 00              .Z.........
```

Maximum Segment Size:
The MSS is the size sent from the recipient to the sender indicating the maximum TCP segment size that can be received.

Sample Default OS MSS:

MSS	Host OS
536	Windows 9x
1024	Alcatel (Xylan) OmniStack 5024
1460	Windows NT, Mac OS9, BSD, Linux

Don't Fragment Flag (0=unset, 1=set):
The Don't Fragment Flag is used by applications to tell routers that under no circumstances may fragmented packets be received by this host. While the Don't Fragment Flag of 0 (May Fragment) is considered the default, many operating systems default to a value of 1, making the DF flag very valuable in OS Detection.

Sample Default DF Settings:

DF Flag	Host OS
0	Cisco IOS 2611/11.3(2)XA4, C2600/12.0(5)T1, 4500/12.0(9), 3640/12.1(2), 3620/12.0(8)
0	Linux 2.2.19
0	CacheOS 3.1 on a CacheFlow 6000
1	Windows XP Pro, Windows 2000 Pro
1	Windows 9x or NT4
1	Mac OS 7.x-9.x

Window Scaling (-1=not present, other=value):
Allows a host to advertise a window value larger than 64kilobytes (default window size) up to a maximum of one gigabyte.

Sample Window Scaling settings:

WS	Host OS
0	Windows 98
75	Windows ME
112	FreeBSD 4.0-STABLE, 3.2-RELEASE (4)
245	Alcatel (Xylan) OmniStack 5024

Additionally, many OS stacks set the following flag settings which helps to further differentiate their OS from others.
SackOK flag (0=unset, 1=set)
NOP flag (0=unset, 1=set)
Packet Size (-1 = irrevelant)

p0f – An Anaysis:
Let's look at a sample SYN packet and apply the fingerprint field matching technique used by p0f:

```
01/18-13:20:26.423134 199.222.100.4:15502 -> 64.245.58.81:21
TCP TTL:124 TOS:0x0 ID:41396 IpLen:20 DgmLen:44 DF
******S* Seq: 0x143967  Ack: 0x0  Win: 0x2000  TcpLen: 24
TCP Options (1) => MSS: 1460
0x0000: 00 D0 79 74 38 00 08 00 20 B0 1A B6 08 00 45 00  ..yt8... .....E.
0x0010: 00 2C A1 B4 40 00 7C 06 B5 EE C7 DE 64 04 40 F5  .,..@.|.....d.@.
0x0020: 3A 51 3C 8E 00 15 00 14 39 67 00 00 00 00 60 02  :Q<.....9g....`.
0x0030: 20 00 5A DF 00 00 02 04 05 B4 00 00              .Z.........
```

If we use our table of fingerprints (p0f.fp) from p0f 1.8 we identify the following key fields:

Fingerprint Table (p0f.fp)	
Field	Value
Window size	8192
Time To Live	124
Maximum Segment Size	1460
Don't Fragment Flag (0=unset, 1=set)	1
Window Scaling (-1=not present, other=value)	0
SackOK flag (0=unset, 1=set)	0
NOP flag (0=unset, 1=set)	0
Packet Size (-1 = irrevelant)	44

This converted to p0f format = 8192:128:1460:1:0:0:0:44

A quick grep of the p0f fingerprint database reveals that this SYN packet "probably" originated from a Windows NT 4.0 host. Additionally since a Windows NT 4.0 box has a default initial TTL of 128 we take our recorded TTL of 124, subtract this from the initial TTL of the OS (128) and add one additional hop for our local router to determine the hop count of 5.

(Initial OS TTL) – (Observerd TTL) + (1) = Hop Count
(128) – (124) + (1) = 5

p0f at Work:

```
usage: p0f [ -f file ] [ -i device ] [ -o file ]
           [ -s file ] [ -vKUt ] [ 'filter rule' ]
 -f file   read fingerprint information from file
 -i device read packets from device
 -s file   read packets from file
 -o file   write output to file (best with -vt)
 -v        verbose mode
 -U        do not display unknown signatures
 -K        do not display known signatures
 -t        add timestamps
```

For the following traces p0f was run on a Slackware 8.0 box running kernel version 2.2.19, tcpdump version 3.6. Only the initial SYN's are being reported.

Trace 1: Telnet from Slackware Linux Host, Kernel 2.2.13

p0f Output:

```
root@dragon:/tmp/p0f-1.8# ./p0f
p0f: passive os fingerprinting utility, version 1.8
(C) Michal Zalewski <lcamtuf@gis.net>, William Stearns <wstearns@pobox.com>
p0f: file: '/etc/p0f.fp', 139 fprints, iface: 'eth0', rule: 'all'.
192.168.1.100 [1 hops]: Debian/Caldera Linux 2.2.x (check)
```

TCPDump Output:

```
16:05:08.272451 192.168.1.100.4262 > 192.168.1.2.23: S 3907891727:3907891727(0) win 16060 <mss
1460,sackOK,timestamp 11275843[|tcp]> (DF) [tos 0x10]  (ttl 64, id 18377, len 60)
0x0000   4510 003c 47c9 4000 4006 6f2c c0a8 0164        E..<G.@.@.o,...d
0x0010   c0a8 0102 10a6 0017 e8ed b20f 0000 0000        ................
0x0020   a002 3ebc caea 0000 0204 05b4 0402 080a        ..>.............
0x0030   00ac 0e43 0000                                 ...C..
```

Trace 2: Telnet from Solaris 2.7

p0f Output:

```
root@dragon:/tmp/p0f-1.8# ./p0f
p0f: passive os fingerprinting utility, version 1.8
(C) Michal Zalewski <lcamtuf@gis.net>, William Stearns <wstearns@pobox.com>
p0f: file: '/etc/p0f.fp', 139 fprints, iface: 'eth0', rule: 'all'.
199.222.100.4 [20 hops]: Solaris 2.6 or 2.7 (1) *
```

TCPDump Output:

```
16:19:40.085117 199.222.100.4.36111 > 192.168.1.2.23: S [tcp sum ok] 4266888852:4266888852(0) win 8760 <mss
1460> (DF) (ttl 236, id 56427, len 44)
0x0000   4500 002c dc6b 4000 ec06 c4d2 c7de 6404        E..,.k@.......d.
0x0010   c0a8 0102 8d0f 0017 fe53 8e94 0000 0000        .........S......
0x0020   6002 2238 6e52 0000 0204 05b4 0000             `."8nR........
```

Trace 3: Telnet from HP/UX 11.0

P0f Output:

```
root@dragon:/tmp/p0f-1.8# ./p0f
p0f: passive os fingerprinting utility, version 1.8
(C) Michal Zalewski <lcamtuf@gis.net>, William Stearns <wstearns@pobox.com>
p0f: file: '/etc/p0f.fp', 139 fprints, iface: 'eth0', rule: 'all'.
xxx.xxx.xxx.4: UNKNOWN [32768:45:1460:1:180:0:1:48].
```

TCPDump Output:

```
16:22:37.544927 xxx.xxx.xxx.4.36784 > 192.168.1.2.23: S [tcp sum ok] 3334327479:3334327479(0) win 32768 <mss
1460,wscale 0,nop> (DF) (ttl 45, id 4353, len 48)
0x0000   4500 0030 1101 4000 2d06 4f3a c7de 6404        E..0..@.-.O:..d.
0x0010   c0a8 0102 8fb0 0017 c6bd ccb7 0000 0000        ................
0x0020   7002 8000 f453 0000 0204 05b4 0303 0001        p....S..........
```

Trace #3 was unable to accurately identify the remote OS type. While we know that the remote host was running HP/UX 11.0, we could compare our results against some other HP/UX 11.0 boxes and if the results concur, an addition could be made to the p0f.fp database. A new entry in p0f.fp would look like this:

```
32768:64:1460:1:180:0:1:48:HP/UX 11.0
```

Note that in the above database entry we modified the reported TTL to a value of 64. This is because our p0f promiscuous node, reported the arriving TTL as 45, and we can assume an approximate hopcount of 19, thus a value of 64. While a traceroute back to the host would reveal the “actual” hopcount, our attempt here is to remain stealthy.

Conclusion:

It must be noted that passive OS fingerprinting is not 100% foolproof. Individual users can tweak kernel and stack settings to mask their OS, also if a user is using an application proxy, it is possible that the fingerprint will not be that of the host but instead will reflect the default settings of the proxy.

While not 100% accurate, when coupled with a sites IDS logs, passive OS fingerprinting can aid the analyst in determining the remote attackers OS type, all without them knowing.

References:

TCP/IP Illustrated Volume 1; W. Richard Stevens

Internet Core Protocols; Eric A. Hall

http://www.stearns.org/p0f/ - passive OS fingerprinting tool

http://project.honeynet.org/papers/finger/ - Know Your Enemy: Passive Fingerprinting

http://www.securiteam.com/tools/51Q020A1SY.htm – Passive OS Detection Tool

http://www.incidents.org/papers/Osfingerprinting.php - Passive OS Fingerprinting: Details and Techniques

http://www.seifried.org/security/network/20011009-passive-os-detection.html – Passive OS Detection and Source Ports

http://www.faqs.org/rfcs/rfc879.html - The TCP Maximum Segment Size and Related Topics

Assignment 2 – Network Detects

1. ICMP PING NMAP (Lumeta Network Mapping Scan)
2. SSH Port Scanning
3. IBM Mass Scan (Network Services Auditor)
4. ICMP – TIMESTAMP
5. Anomalous Traffic (SRC Port 5635 → DST Port 0)

Detect #1 – ICMP PING NMAP (Lumeta Network Mapping Scan):

```
12/05-04:10:00.606843 65.198.68.56 -> xxx.xxx.xxx.3
ICMP TTL:1 TOS:0x0 ID:49682 IpLen:20 DgmLen:32
Type:8  Code:0  ID:21990   Seq:7353  ECHO
--
12/05-04:10:00.606843 65.198.68.56 -> xxx.xxx.xxx.3
ICMP TTL:1 TOS:0x0 ID:49682 IpLen:20 DgmLen:32
Type:8  Code:0  ID:21990   Seq:7353  ECHO
--
12/05-04:10:46.894262 65.198.68.56 -> xxx.xxx.xxx.3
ICMP TTL:2 TOS:0x0 ID:30433 IpLen:20 DgmLen:32
Type:8  Code:0  ID:21990   Seq:1742  ECHO
--
12/05-04:10:46.894262 65.198.68.56 -> xxx.xxx.xxx.3
ICMP TTL:2 TOS:0x0 ID:30433 IpLen:20 DgmLen:32
Type:8  Code:0  ID:21990   Seq:1742  ECHO
--
12/06-02:18:05.633731 65.198.68.56 -> xxx.xxx.xxx.20
ICMP TTL:1 TOS:0x0 ID:60990 IpLen:20 DgmLen:32
Type:8  Code:0  ID:22961   Seq:5392  ECHO
--
12/06-02:18:42.992268 65.198.68.56 -> xxx.xxx.xxx.20
ICMP TTL:2 TOS:0x0 ID:32812 IpLen:20 DgmLen:32
Type:8  Code:0  ID:22961   Seq:1119  ECHO
--
12/06-02:19:17.742153 65.198.68.56 -> xxx.xxx.xxx.20
ICMP TTL:3 TOS:0x0 ID:2025 IpLen:20 DgmLen:32
Type:8  Code:0  ID:22961   Seq:13362  ECHO
--
```

12/07-05:21:51.950958 65.198.68.56 -> xxx.xxx.xxx.10
ICMP TTL:1 TOS:0x0 ID:34843 IpLen:20 DgmLen:32
Type:8 Code:0 ID:40428 Seq:6340 ECHO
--
12/07-05:22:33.987452 65.198.68.56 -> xxx.xxx.xxx.10
ICMP TTL:2 TOS:0x0 ID:11343 IpLen:20 DgmLen:32
Type:8 Code:0 ID:40428 Seq:6716 ECHO
--
12/07-05:23:16.840565 65.198.68.56 -> xxx.xxx.xxx.10
ICMP TTL:3 TOS:0x0 ID:54196 IpLen:20 DgmLen:32
Type:8 Code:0 ID:40428 Seq:1814 ECHO
--
12/07-05:24:02.838934 65.198.68.56 -> xxx.xxx.xxx.10
ICMP TTL:4 TOS:0x0 ID:34658 IpLen:20 DgmLen:32
Type:8 Code:0 ID:40428 Seq:2589 ECHO
--
12/07-05:24:45.580229 65.198.68.56 -> xxx.xxx.xxx.10
ICMP TTL:1 TOS:0x0 ID:11864 IpLen:20 DgmLen:32
Type:8 Code:0 ID:40428 Seq:3681 ECHO
--
12/08-03:27:58.349024 65.198.68.56 -> xxx.xxx.xxx.10
ICMP TTL:1 TOS:0x0 ID:40173 IpLen:20 DgmLen:32
Type:8 Code:0 ID:62373 Seq:10744 ECHO
--
12/08-03:28:41.234513 65.198.68.56 -> xxx.xxx.xxx.10
ICMP TTL:2 TOS:0x0 ID:17522 IpLen:20 DgmLen:32
Type:8 Code:0 ID:62373 Seq:8595 ECHO
--
12/08-03:29:23.818429 65.198.68.56 -> xxx.xxx.xxx.10
ICMP TTL:3 TOS:0x0 ID:60106 IpLen:20 DgmLen:32
Type:8 Code:0 ID:62373 Seq:10283 ECHO
--
12/08-03:30:06.387126 65.198.68.56 -> xxx.xxx.xxx.10
ICMP TTL:4 TOS:0x0 ID:37139 IpLen:20 DgmLen:32
Type:8 Code:0 ID:62373 Seq:15135 ECHO
--
12/09-02:27:48.145392 65.198.68.56 -> xxx.xxx.xxx.2
ICMP TTL:1 TOS:0x0 ID:57621 IpLen:20 DgmLen:32
Type:8 Code:0 ID:47588 Seq:1838 ECHO
--
12/09-02:28:17.987022 65.198.68.56 -> xxx.xxx.xxx.2
ICMP TTL:2 TOS:0x0 ID:21927 IpLen:20 DgmLen:32
Type:8 Code:0 ID:47588 Seq:2812 ECHO
--
12/09-02:28:46.725192 65.198.68.56 -> xxx.xxx.xxx.2
ICMP TTL:3 TOS:0x0 ID:50665 IpLen:20 DgmLen:32
Type:8 Code:0 ID:47588 Seq:300 ECHO
--
12/09-02:29:17.749278 65.198.68.56 -> xxx.xxx.xxx.2
ICMP TTL:4 TOS:0x0 ID:16153 IpLen:20 DgmLen:32
Type:8 Code:0 ID:47588 Seq:7858 ECHO
--
12/10-05:46:18.433708 65.198.68.56 -> xxx.xxx.xxx.20
ICMP TTL:1 TOS:0x0 ID:63457 IpLen:20 DgmLen:32
Type:8 Code:0 ID:56734 Seq:977 ECHO
--
12/10-05:46:59.801086 65.198.68.56 -> xxx.xxx.xxx.20
ICMP TTL:2 TOS:0x0 ID:39289 IpLen:20 DgmLen:32
Type:8 Code:0 ID:56734 Seq:10392 ECHO
--
12/10-05:47:38.197541 65.198.68.56 -> xxx.xxx.xxx.20
ICMP TTL:3 TOS:0x0 ID:12148 IpLen:20 DgmLen:32
Type:8 Code:0 ID:56734 Seq:4545 ECHO
--
12/11-02:17:54.440411 65.198.68.56 -> xxx.xxx.xxx.2
ICMP TTL:1 TOS:0x0 ID:34516 IpLen:20 DgmLen:32
Type:8 Code:0 ID:27350 Seq:3018 ECHO
--
12/11-02:18:29.212700 65.198.68.56 -> xxx.xxx.xxx.2
ICMP TTL:2 TOS:0x0 ID:3752 IpLen:20 DgmLen:32

```
Type:8  Code:0  ID:27350   Seq:9938  ECHO

12/11-02:19:03.532065 65.198.68.56 -> xxx.xxx.xxx.2
ICMP TTL:3 TOS:0x0 ID:38071 IpLen:20 DgmLen:32
Type:8  Code:0  ID:27350   Seq:4805  ECHO
--
12/11-02:19:40.339053 65.198.68.56 -> xxx.xxx.xxx.2
ICMP TTL:4 TOS:0x0 ID:9343 IpLen:20 DgmLen:32
Type:8  Code:0  ID:27350   Seq:14059  ECHO
--
12/12-02:31:46.840559 65.198.68.56 -> xxx.xxx.xxx.3
ICMP TTL:1 TOS:0x0 ID:38094 IpLen:20 DgmLen:32
Type:8  Code:0  ID:58841   Seq:6591  ECHO
--
12/12-02:32:31.991571 65.198.68.56 -> xxx.xxx.xxx.3
ICMP TTL:2 TOS:0x0 ID:17708 IpLen:20 DgmLen:32
Type:8  Code:0  ID:58841   Seq:6668  ECHO
--
12/13-02:17:37.997212 65.198.68.56 -> xxx.xxx.xxx.132
ICMP TTL:1 TOS:0x0 ID:64400 IpLen:20 DgmLen:32
Type:8  Code:0  ID:12166   Seq:11647  ECHO
--
12/13-02:18:11.847782 65.198.68.56 -> xxx.xxx.xxx.132
ICMP TTL:2 TOS:0x0 ID:32715 IpLen:20 DgmLen:32
Type:8  Code:0  ID:12166   Seq:6421  ECHO
--
12/13-02:18:45.195092 65.198.68.56 -> xxx.xxx.xxx.132
ICMP TTL:3 TOS:0x0 ID:526 IpLen:20 DgmLen:32
Type:8  Code:0  ID:12166   Seq:1219  ECHO
--
12/14-05:30:22.684761 65.198.68.56 -> xxx.xxx.xxx.3
ICMP TTL:1 TOS:0x0 ID:52339 IpLen:20 DgmLen:32
Type:8  Code:0  ID:4553   Seq:9215  ECHO
--
12/14-05:30:50.668193 65.198.68.56 -> xxx.xxx.xxx.3
ICMP TTL:2 TOS:0x0 ID:14786 IpLen:20 DgmLen:32
Type:8  Code:0  ID:4553   Seq:9964  ECHO
```

1. **Source of Trace:**
   The souce of this trace is a network that I monitor.

2. **Detect was generated by:**
   This detect was generated by Snort Version 1.8.3 (Build 88), using the default rules included with 1.8.3.

   Rule that generated this alert:
   ```
   alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING NMAP"; dsize: 0; i
   type: 8; reference:arachnids,162; classtype:attempted-recon; sid:469; rev:1;)
   ```

3. **Probability the source address was spoofed:**
   It is very unlikely that the source address was spoofed as the source is eliciting a response (ECHO REPLY) from the destination host.

4. **Description of attack:**
   In this attack, or reconnaissance probe, the attacker sends ICMP ECHO packets in an attempt to illicit a response to map our network.

5. **Attack mechanism:**

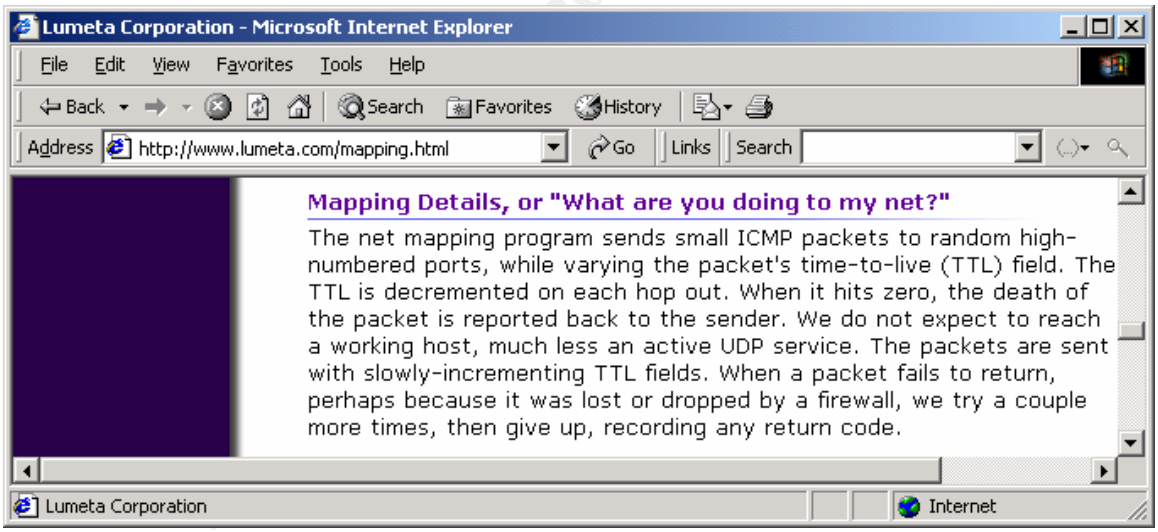
In the above trace we see ICMP ECHO requests directed with varying TTL's to hosts on our class-C DMZ subnet. The intent is to identify hosts on our network that are active and respond with an ICMP ECHO REPLY. Additionally, by varying the TTL values an attacker can attempt to map a network behind a firewall. A very similar technique is exploited with the tool "firewalk", written by Mike Schiffman. Firewalk sends TCP or UDP packets to a target host/port with a TTL that will expire just before it reaches the host. If the firewall or packet filtering device allows this traffic in, the host will respond with a TTL exceeded in transit message, indicating that the firewall or packet filter device allows this traffic destined for this host/port pair inbound. If the packet is not allowed in by the firewall, then no response will be returned and it can be assumed that this host/port pair is blocked by the firewall.

6. Correlations:

The source address in this attack is resolvable via nslookup:

Name: netmapper.research.lumeta.com
Address: 65.198.68.56

Lumeta Corporation, provides firewall analysis and network mapping services. After scouring their website I came across a link to their research page that addresses the trace seen above:
http://www.lumeta.com/mapping.html



In addition to providing a description of their network probes, Lumeta also makes available for download a map and database listing of all the hosts and networks mapped to date. I was pleased to find that our Class-C, although listed in Lumeta's database, did not include and IP addresses of active hosts on our network. Readers of this practical may want to lookup their own managed network address-space to determine if their hosts have been scanned and logged by this "Mapping Project".

http://www.usenix.org/publications/library/proceedings/lisa2000/burch/burch_html - Bill Cheswick - Lumeta Corp. Tracing Anonymous Packets to Their Approximate Source. URL:

http://packetstormsecurity.org/UNIX/audit/firewalk/firewalk-0.99.1.tar.gz – Firewalk, Network Mapping Utility

7. **Evidence of active targeting:**
   While these attacks appeared to be directly targeting hosts on my network, after reviewing Lumeta's website, I am confident that this was not a "directed" attack or probe.

8. **Severity:**
   The following formula is used to identify the severity of this and the following four traces:

   (Target Criticality + Attack Lethality) – (System Countermeasures + Network Countermeasures) = Attack Severity

   Each item is ranked from 1 to 5, with 1 being the lowest and 5 representing the worst case scenario.
   Target Criticality: 2 (Targets were random webservers in our DMZ)
   Attack Lethality: 2 (Reconnaissance or information gathering)
   System Countermeasures: 2 (Systems respond to ICMP ECHO requests)
   Network Countermeasures: 4 (Firewall blocked inbound ICMP ECHO packets to these hosts)
   Attack Severity: -2. (2 + 2) – (2 + 4) = -2

9. **Defensive recommendation:**
   While it is possible to configure an operating system to not respond to ICMP packets, ICMP is often used for network troubleshooting so it would be better to configure your Firewall, packet filter or ACL device to block inbound ICMP.

10. **Multiple choice test question:**
    ICMP type:8 code:0 is what type of ICMP Packet?
    A. Port Unreachable
    B. Time to Live exceeded in Transit
    C. Echo
    D. Echo Reply
    Answer: C (Echo)

Detect #2 - SSH Port Scanning

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

12/27-01:26:05.417384 207.175.61.96:22 -> xxx.xxx.xxx.128:22
TCP TTL:28 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x27A8CB7D  Ack: 0x24E57586  Win: 0x404  TcpLen: 20

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

```
12/27-01:26:05.430503 207.175.61.96:22 -> xxx.xxx.xxx.129:22
TCP TTL:28 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x27A8CB7D  Ack: 0x24E57586  Win: 0x404  TcpLen: 20

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

12/27-01:26:05.446856 207.175.61.96:22 -> xxx.xxx.xxx.130:22
TCP TTL:28 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x27A8CB7D  Ack: 0x24E57586  Win: 0x404  TcpLen: 20

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

12/27-01:26:05.472497 207.175.61.96:22 -> xxx.xxx.xxx.131:22
TCP TTL:28 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x1561C1D9  Ack: 0x59A82E1D  Win: 0x404  TcpLen: 20

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

12/27-01:26:05.493501 207.175.61.96:22 -> xxx.xxx.xxx.132:22
TCP TTL:28 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x1561C1D9  Ack: 0x59A82E1D  Win: 0x404  TcpLen: 20

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

12/27-01:26:05.508496 207.175.61.96:22 -> xxx.xxx.xxx.133:22
TCP TTL:28 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x1561C1D9  Ack: 0x59A82E1D  Win: 0x404  TcpLen: 20

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

1. **Source of Trace:**
   The souce of this trace is a network that I monitor.

2. **Detect was generated by:**
   SNORT Version 1.8.3 (Build 88)

3. **Probability the source address was spoofed:**
   Low. This attack is an attempt to scan for open ports, specifically port 22 (SSH). Although it is possible for the attacker to spoof the source address, he would need to be located somewhere along the return path of this traffic to sniff the response.

4. **Description of attack:**
   The attacker is attempting to determine if a host has port 22 (SSH) open. Recently exploits have been published in the wild that attack SSH remotely and return root-level user permissions if successful.

5. **Attack mechanism:**
   A couple of unique signatures in this trace make it possible to identify the tool, or a varient based on the original tool, known as synscan.c, used in this scan. Specifically this scan is very fast, all five hosts were scanned in under a second, too fast to be executed sequentially by hand on a command line. The source AND destination ports are the same and the IP ID remains static at a value of 39426. A search on google, revealed many other traces exhibiting the same characteristics of this trace, and one Incidents mailing list post,

http://lists.jammed.com/incidents/2001/10/0104.html, detailed the unique signatures addressed above. While we are able to determine that this scan was probably bred from sysnscan.c, we can not determine what specific attack might follow a successful scan of an open SSH port.

6. **Correlations:**
   MyNetWatchman's incident database included three corollary references to the attacker's source ip. All incidents were similar port 22(SSH) port scans.
   http://www.mynetwatchman.com/ - Incident #'s (2432273, 2710961, 2581270)

   http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0144

   http://lists.jammed.com/incidents/2001/10/0104.html

   http://www.sans.org/y2k/practical/donald_smith_gcia.doc - Mscan, sscan and synscan the evolution of a worm enabling vulnerability scanner that spans 2 milleniums.
   By Donald.J.Smith 5/20/2001

7. **Evidence of active targeting:**
   Because this scan was attempting to identify all SSH daemons on our class-C subnet (the above trace was snipped to include just a few hosts) it is unlikely that this is active targeting.

8. **Severity:**
   Target Criticality: 2 (webservers in our DMZ)
   Attack Lethality: 1 (SSH is not running on any hosts)
   System Countermeasures: 4 (sshd not running on hosts)
   Network Countermeasures: 4 (firewall drops inbound ssh connections)
   Attack Severity: -5. (2 + 1) – (4 + 4) = -5

9. **Defensive recommendation:**
   Defenses are adequate as inbound ssh is not currently allowed in this network. Additionally SynFin scans are logged and alerted on both the firewall and IDS.

10. **Multiple choice test question:**
    What is the purpose of the following packet?

```
12/27-01:26:05.417384 207.175.61.96:22 -> xxx.xxx.xxx.128:22
TCP TTL:28 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x27A8CB7D  Ack: 0x24E57586  Win: 0x404  TcpLen: 20
```

    A. SSH Port Scan
    B. TCP query for DNS services
    C. FTP data transfer
    D. Acknologemnet of the end of a SSH session

Answer: A (SSH Port Scan, Note that both the SYN and FIN flags are set, the source port is non ephemeral, and with a quick google search, you would find that an IP ID of 39426 is indicitive of a packet crafted with synscan.c or a varient of.)

Detect #3 – IBM Mass Scan "Network Services Auditor"

```
12/21-02:05:05.971929 129.33.73.59:50084 -> xxx.xxx.xxx.127:80
TCP TTL:240 TOS:0x0 ID:9127 IpLen:20 DgmLen:289 DF
***AP*** Seq: 0x189B6E2  Ack: 0x6678A97F  Win: 0x2238  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 36 6B 6C  GET /scripts/6kl
78 62 69 36 32 20 48 54 54 50 2F 31 2E 30 0D 0A  xbi62 HTTP/1.0..
48 6F 73 74 3A 20 31 34 39 2E 31 33 31 2E 32 30  Host: 149.131.20
2E 36 37 3A 38 30 30 30 0D 0A 55 73 65 72 2D 41  .67:8000..User-A
67 65 6E 74 3A 20 4E 65 74 77 6F 72 6B 2D 53 65  gent: Network-Se
72 76 69 63 65 73 2D 41 75 64 69 74 6F 72 2F 31  rvices-Auditor/1
2E 33 2E 31 0D 0A 41 63 63 65 70 74 3A 20 2A 2F  .3.1..Accept: */
2A 0D 0A 58 2D 4E 53 41 2D 4C 69 63 65 6E 73 65  *..X-NSA-License
3A 20 37 32 34 65 62 32 31 63 33 64 37 64 38 64  : 724eb21c3d7d8d
38 63 62 31 38 34 32 63 32 39 37 32 32 39 31 32  8cb1842c29722912
38 30 66 32 62 37 31 61 32 37 31 32 33 35 65 31  80f2b71a271235e1
31 37 32 62 63 63 30 63 66 32 37 32 33 64 31 36  172bcc0cf2723d16
38 39 61 31 37 31 63 30 66 38 34 32 39 63 33 31  89a171c0f8429c31
36 35 33 65 34 66 32 62 35 63 31 65 33 32 30 32  653e4f2b5c1e3202
32 65 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20  2e..Connection:
63 6C 6F 73 65 0D 0A 0D 0A                       close....

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

12/21-02:05:06.198018 129.33.73.59:50085 -> xxx.xxx.xxx.127:80
TCP TTL:240 TOS:0x0 ID:9134 IpLen:20 DgmLen:289 DF
***AP*** Seq: 0xBFA9938A  Ack: 0x6BB2D8DD  Win: 0x2238  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 69 69 73  GET /scripts/iis
61 64 6D 69 6E 20 48 54 54 50 2F 31 2E 30 0D 0A  admin HTTP/1.0..
48 6F 73 74 3A 20 31 34 39 2E 31 33 31 2E 32 30  Host: 149.131.20
2E 36 37 3A 38 30 30 30 0D 0A 55 73 65 72 2D 41  .67:8000..User-A
67 65 6E 74 3A 20 4E 65 74 77 6F 72 6B 2D 53 65  gent: Network-Se
72 76 69 63 65 73 2D 41 75 64 69 74 6F 72 2F 31  rvices-Auditor/1
2E 33 2E 31 0D 0A 41 63 63 65 70 74 3A 20 2A 2F  .3.1..Accept: */
2A 0D 0A 58 2D 4E 53 41 2D 4C 69 63 65 6E 73 65  *..X-NSA-License
3A 20 37 32 34 65 62 32 31 63 33 64 37 64 38 64  : 724eb21c3d7d8d
38 63 62 31 38 34 32 63 32 39 37 32 32 39 31 32  8cb1842c29722912
38 30 66 32 62 37 31 61 32 37 31 32 33 35 65 31  80f2b71a271235e1
31 37 32 62 63 63 30 63 66 32 37 32 33 64 31 36  172bcc0cf2723d16
38 39 61 31 37 31 63 30 66 38 34 32 39 63 33 31  89a171c0f8429c31
36 35 33 65 34 66 32 62 35 63 31 65 33 32 30 32  653e4f2b5c1e3202
32 65 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20  2e..Connection:
63 6C 6F 73 65 0D 0A 0D 0A                       close....

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

12/21-02:05:07.086878 129.33.73.59:50086 -> xxx.xxx.xxx.127:80
TCP TTL:240 TOS:0x0 ID:9140 IpLen:20 DgmLen:303 DF
***AP*** Seq: 0x261B59AD  Ack: 0x667E00F5  Win: 0x2238  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 69 69 73  GET /scripts/iis
61 64 6D 69 6E 2F 7A 61 70 6D 6F 69 7A 38 2E 68  admin/zapmoiz8.h
74 6D 6C 20 48 54 54 50 2F 31 2E 30 0D 0A 48 6F  tml HTTP/1.0..Ho
73 74 3A 20 31 34 39 2E 31 33 31 2E 32 30 2E 36  st: 149.131.20.6
37 3A 38 30 30 30 0D 0A 55 73 65 72 2D 41 67 65  7:8000..User-Age
6E 74 3A 20 4E 65 74 77 6F 72 6B 2D 53 65 72 76  nt: Network-Serv
69 63 65 73 2D 41 75 64 69 74 6F 72 2F 31 2E 33  ices-Auditor/1.3
2E 31 0D 0A 41 63 63 65 70 74 3A 20 2A 2F 2A 0D  .1..Accept: */*.
0A 58 2D 4E 53 41 2D 4C 69 63 65 6E 73 65 3A 20  .X-NSA-License:
37 32 34 65 62 32 31 63 33 64 37 64 38 64 38 63  724eb21c3d7d8d8c
62 31 38 34 32 63 32 39 37 32 32 39 31 32 38 30  b1842c2972291280
66 32 62 37 31 61 32 37 31 32 33 35 65 31 31 37  f2b71a271235e117
32 62 63 63 30 63 66 32 37 32 33 64 31 36 38 39  2bcc0cf2723d1689
```

```
61 31 37 31 63 30 66 38 34 32 39 63 33 31 36 35  a171c0f8429c3165
33 65 34 66 32 62 35 63 31 65 33 32 30 32 32 65  3e4f2b5c1e32022e
0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 63 6C  ..Connection: cl
6F 73 65 0D 0A 0D 0A                             ose....

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

12/21-02:05:07.390866 129.33.73.59:50087 -> xxx.xxx.xxx.127:80
TCP TTL:240 TOS:0x0 ID:9146 IpLen:20 DgmLen:302 DF
***AP*** Seq: 0xA6F4407E  Ack: 0x6BB95C40  Win: 0x2238  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 69 69 73  GET /scripts/iis
61 64 6D 69 6E 2F 64 65 66 61 75 6C 74 2E 68 74  admin/default.ht
6D 6C 20 48 54 54 50 2F 31 2E 30 0D 0A 48 6F 73  ml HTTP/1.0..Hos
74 3A 20 31 34 39 2E 31 33 31 2E 32 30 2E 36 37  t: 149.131.20.67
3A 38 30 30 30 0D 0A 55 73 65 72 2D 41 67 65 6E  :8000..User-Agen
74 3A 20 4E 65 74 77 6F 72 6B 2D 53 65 72 76 69  t: Network-Servi
63 65 73 2D 41 75 64 69 74 6F 72 2F 31 2E 33 2E  ces-Auditor/1.3.
31 0D 0A 41 63 63 65 70 74 3A 20 2A 2F 2A 0D 0A  1..Accept: */*..
58 2D 4E 53 41 2D 4C 69 63 65 6E 73 65 3A 20 37  X-NSA-License: 7
32 34 65 62 32 31 63 33 64 37 64 38 64 38 63 62  24eb21c3d7d8d8cb
31 38 34 32 63 32 39 37 32 32 39 31 32 38 30 66  1842c2972291280f
32 62 37 31 61 32 37 31 32 33 35 65 31 31 37 32  2b71a271235e1172
62 63 63 30 63 66 32 37 32 33 64 31 36 38 39 61  bcc0cf2723d1689a
31 37 31 63 30 66 38 34 32 39 63 33 31 36 35 33  171c0f8429c31653
65 34 66 32 62 35 63 31 65 33 32 30 32 32 65 0D  e4f2b5c1e32022e.
0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 63 6C 6F  .Connection: clo
73 65 0D 0A 0D 0A                                se....

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

12/21-02:05:07.646206 129.33.73.59:50088 -> xxx.xxx.xxx.127:80
TCP TTL:240 TOS:0x0 ID:9152 IpLen:20 DgmLen:302 DF
***AP*** Seq: 0xCE02E0A1  Ack: 0x66831CEA  Win: 0x2238  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 69 69 73  GET /scripts/iis
61 64 6D 69 6E 2F 75 70 36 31 65 33 6A 35 2E 64  admin/up61e3j5.d
6C 6C 20 48 54 54 50 2F 31 2E 30 0D 0A 48 6F 73  ll HTTP/1.0..Hos
74 3A 20 31 34 39 2E 31 33 31 2E 32 30 2E 36 37  t: 149.131.20.67
3A 38 30 30 30 0D 0A 55 73 65 72 2D 41 67 65 6E  :8000..User-Agen
74 3A 20 4E 65 74 77 6F 72 6B 2D 53 65 72 76 69  t: Network-Servi
63 65 73 2D 41 75 64 69 74 6F 72 2F 31 2E 33 2E  ces-Auditor/1.3.
31 0D 0A 41 63 63 65 70 74 3A 20 2A 2F 2A 0D 0A  1..Accept: */*..
58 2D 4E 53 41 2D 4C 69 63 65 6E 73 65 3A 20 37  X-NSA-License: 7
32 34 65 62 32 31 63 33 64 37 64 38 64 38 63 62  24eb21c3d7d8d8cb
31 38 34 32 63 32 39 37 32 32 39 31 32 38 30 66  1842c2972291280f
32 62 37 31 61 32 37 31 32 33 35 65 31 31 37 32  2b71a271235e1172
62 63 63 30 63 66 32 37 32 33 64 31 36 38 39 61  bcc0cf2723d1689a
31 37 31 63 30 66 38 34 32 39 63 33 31 36 35 33  171c0f8429c31653
65 34 66 32 62 35 63 31 65 33 32 30 32 32 65 0D  e4f2b5c1e32022e.
0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 63 6C 6F  .Connection: clo
73 65 0D 0A 0D 0A                                se....

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

12/21-02:05:07.898841 129.33.73.59:50089 -> xxx.xxx.xxx.127:80
TCP TTL:240 TOS:0x0 ID:9158 IpLen:20 DgmLen:297 DF
***AP*** Seq: 0x52C8557F  Ack: 0x66884993  Win: 0x2238  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 69 69 73  GET /scripts/iis
61 64 6D 69 6E 2F 69 73 6D 2E 64 6C 6C 20 48 54  admin/ism.dll HT
54 50 2F 31 2E 30 0D 0A 48 6F 73 74 3A 20 31 34  TP/1.0..Host: 14
39 2E 31 33 31 2E 32 30 2E 36 37 3A 38 30 30 30  9.131.20.67:8000
0D 0A 55 73 65 72 2D 41 67 65 6E 74 3A 20 4E 65  ..User-Agent: Ne
74 77 6F 72 6B 2D 53 65 72 76 69 63 65 73 2D 41  twork-Services-A
75 64 69 74 6F 72 2F 31 2E 33 2E 31 0D 0A 41 63  uditor/1.3.1..Ac
63 65 70 74 3A 20 2A 2F 2A 0D 0A 58 2D 4E 53 41  cept: */*..X-NSA
2D 4C 69 63 65 6E 73 65 3A 20 37 32 34 65 62 32  -License: 724eb2
31 63 33 64 37 64 38 64 38 63 62 31 38 34 32 63  1c3d7d8d8cb1842c
32 39 37 32 32 39 31 32 38 30 66 32 62 37 31 61  2972291280f2b71a
32 37 31 32 33 35 65 31 31 37 32 62 63 63 30 63  271235e1172bcc0c
66 32 37 32 33 64 31 36 38 39 61 31 37 31 63 30  f2723d1689a171c0
66 38 34 32 39 63 33 31 36 35 33 65 34 66 32 62  f8429c31653e4f2b
```

```
35 63 31 65 33 32 30 32 32 65 0D 0A 43 6F 6E 6E  5c1e32022e..Conn
65 63 74 69 6F 6E 3A 20 63 6C 6F 73 65 0D 0A 0D  ection: close...
0A                                               .

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

1. **Source of Trace:**
   The source of this trace is a network that I monitor.

2. **Detect was generated by:**
   SNORT Version 1.8.3 (Build 88)

3. **Probability the source address was spoofed:**
   Low. The attacker has already established a session with the webserver and is expecting a response to it's queries.

4. **Description of attack:**
   This attack set of hundreds of alarms on our IDS sensor (a small sampling of the alerts is presented here). The attacker is using a proxy server, identified by "**Host: 149.131.20.67:8000**" in the traces above, to scan our host's webserver for the following vulnerabilities.

   ```
   GET /scripts/6klxbi62
   GET /scripts/iisadmin
   GET /scripts/iisadmin/zapmoiz8.html
   GET /scripts/iisadmin/default.html
   GET /scripts/iisadmin/up61e3j5.dll
   GET /scripts/iisadmin/ism.dll
   ```

   Additionally the packet decode lists a license #"**X-NSA-License:724eb21c3d7d8d8cb1842c2972291280f2b71a2 71235e1172bcc0c f2723d1689a171c0f8429c31653e4f2b5c1e32022e**" NSA is IBM's "Network Services Auditor", a tool used to scan a host for known vulnerabilities. No correlation can be made to the License #, but I believe it corresponds to the software license issued to that specific version of the software. A call was placed to the admin of our website, who told us that a "Big-5" security consulting firm was hired to test the security of this site.

5. **Attack mechanism:**
   Clearly this is a very loud soup-to-nuts vulnerability scan. The attack program attempts to scan the host for files, programs, and services with known vulnerabilities. Specifically, the above traces are looking for known vulnerabilities in Microsoft's IIS server. In IIS version 4.0, the ism.dll in /scripts/iisadmin was vulnerable to remote attacks enabling the attacker to gain access to sensitive files and possible the Administrator's password. There is no indication from the IDS logs or from the Web logs that a vulnerability was detected.

6. **Correlations:**

CVE Candidate CAN-1999-1538

http://www.research.ibm.com/gsal/gsal-watson.html - IBM Watson Research - Global Security Analysis Lab

7. **Evidence of active targeting:**
While not malicious in nature, this is absolutely active targeting. A thorough search of the IDS logs for the attackers source IP revealed no other hosts on our network were attacked.

8. **Severity:**
Target Criticality: 4 (financial webserver, hence the business need for an external audit assessment)
Attack Lethality: 2 (Common vulnerability assessment scan)
System Countermeasures: 4 (Host is well patched and no know web based vulnerabilities exist.)
Network Countermeasures: 3 (System is firewalled, however port 80 web traffic is not restricted)
Attack Severity: -1. (4 + 2) – (4 + 3) = -1

9. **Defensive recommendation:**
The current defensive stance is appropriate. The host is running a mature operating system with all the latest patches and a restrictive stateful firewall is in place.

10. **Multiple choice test question:**
In the following trace, what is the attacker looking for?

```
12/21-02:05:07.898841 129.33.73.59:50089 -> xxx.xxx.xxx.127:80
TCP TTL:240 TOS:0x0 ID:9158 IpLen:20 DgmLen:297 DF
***AP*** Seq: 0x52C8557F  Ack: 0x66884993  Win: 0x2238  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 69 69 73  GET /scripts/iis
61 64 6D 69 6E 2F 69 73 6D 2E 64 6C 6C 20 48 54  admin/ism.dll HT
54 50 2F 31 2E 30 0D 0A 48 6F 73 74 3A 20 31 34  TP/1.0..Host: 14
39 2E 31 33 31 2E 32 30 2E 36 37 3A 38 30 30 30  9.131.20.67:8000
0D 0A 55 73 65 72 2D 41 67 65 6E 74 3A 20 4E 65  ..User-Agent: Ne
74 77 6F 72 6B 2D 53 65 72 76 69 63 65 73 2D 41  twork-Services-A
75 64 69 74 6F 72 2F 31 2E 33 2E 31 0D 0A 41 63  uditor/1.3.1..Ac
63 65 70 74 3A 20 2A 2F 2A 0D 0A 58 2D 4E 53 41  cept: */*..X-NSA
2D 4C 69 63 65 6E 73 65 3A 20 37 32 34 65 62 32  -License: 724eb2
31 63 33 64 37 64 38 64 38 63 62 31 38 34 32 63  1c3d7d8d8cb1842c
32 39 37 32 32 39 31 32 38 30 66 32 62 37 31 61  2972291280f2b71a
32 37 31 32 33 35 65 31 31 37 32 62 63 63 30 63  271235e1172bcc0c
66 32 37 32 33 64 31 36 38 39 61 31 37 31 63 30  f2723d1689a171c0
66 38 34 32 39 63 33 31 36 35 33 65 34 66 32 62  f8429c31653e4f2b
35 63 31 65 33 32 30 32 32 65 0D 0A 43 6F 6E 6E  5c1e32022e..Conn
65 63 74 69 6F 6E 3A 20 63 6C 6F 73 65 0D 0A 0D  ection: close...
0A                                               .
```

A. An unpatched Linux Box
B. A default NT IIS Installation
C. A SGI Indy Server
D. Solaris lpd vulnerability
Answer: B (A default NT IIS Installation)

Detect #4 - ICMP – TIMESTAMP:

```
01/03-02:06:14.082545 158.130.5.110 -> xxx.xxx.xxx.89
ICMP TTL:231 TOS:0x0 ID:50987 IpLen:20 DgmLen:84
Type:13  Code:0  TIMESTAMP REQUEST
04 3D E7 A4 00 00 00 00 68 74 74 70 3A 2F 2F 77  .=......http://w
77 77 2E 63 69 73 2E 75 70 65 6E 6E 2E 65 64 75  ww.cis.upenn.edu
2F 7E 61 6E 61 67 6E 6F 73 74 2F 63 69 6E 67 2E  /~anagnost/cing.
68 74 6D 6C 00 00 00 00                          html....

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

01/03-02:09:35.375583 158.130.5.110 -> xxx.xxx.xxx.89
ICMP TTL:231 TOS:0x0 ID:31045 IpLen:20 DgmLen:84
Type:13  Code:0  TIMESTAMP REQUEST
04 40 F9 FD 00 00 00 00 68 74 74 70 3A 2F 2F 77  .@......http://w
77 77 2E 63 69 73 2E 75 70 65 6E 6E 2E 65 64 75  ww.cis.upenn.edu
2F 7E 61 6E 61 67 6E 6F 73 74 2F 63 69 6E 67 2E  /~anagnost/cing.
68 74 6D 6C 00 00 00 00                          html....
```

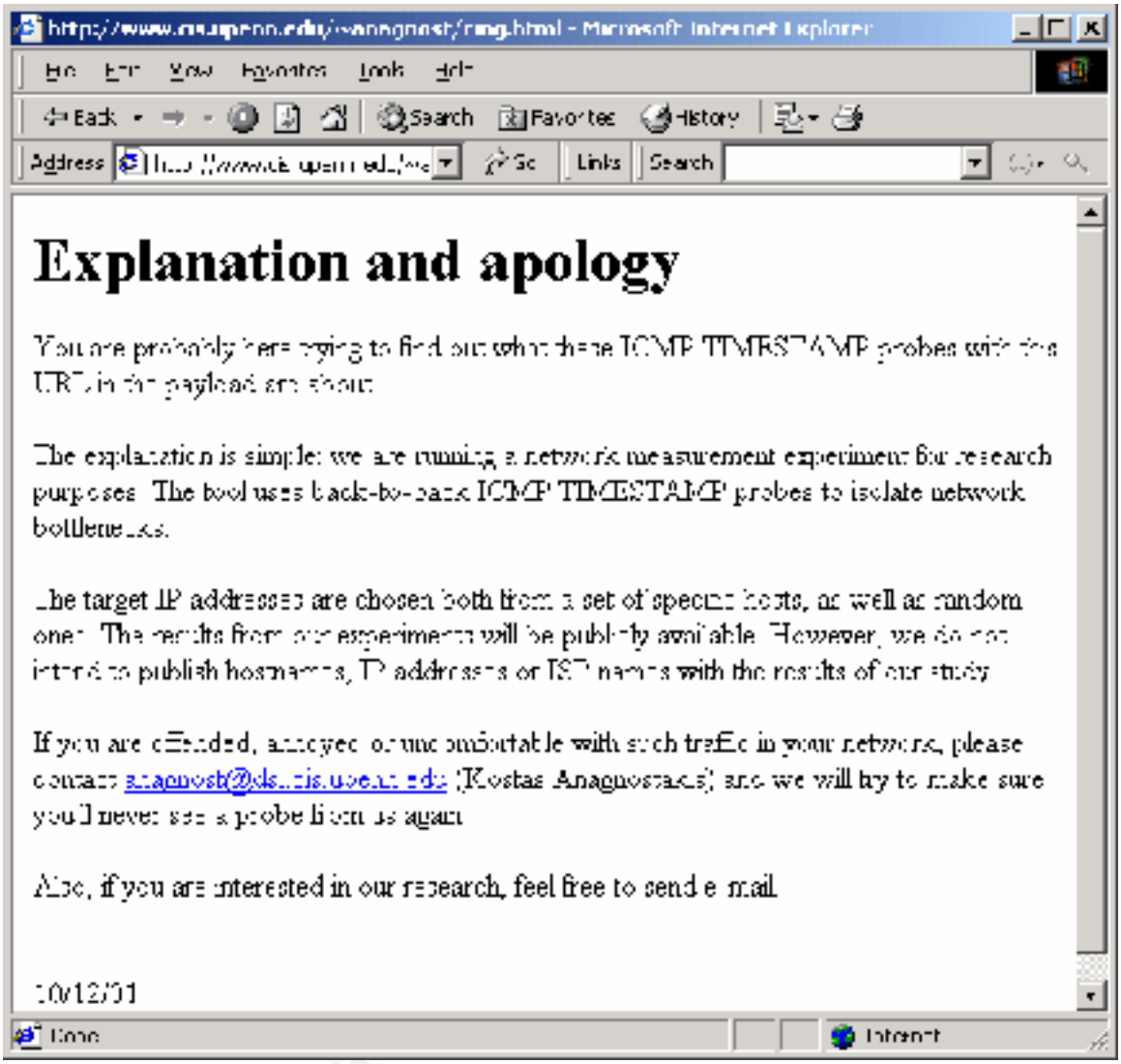
1. **Source of Trace:**
   NIDS located outside a firewall that I manage.

2. **Detect was generated by:**
   SNORT Version 1.8.3 (Build 88)

3. **Probability the source address was spoofed:**
   Unlikely, It appears as if the attacker is attempting to map our network using ICMP TIMESTAMP REQUESTs. For this reconnaissance attack to be successful, the source IP must not be spoofed, or a skilled attacker must be directly in the return path of the traffic to intercept the response.

4. **Description of attack:**
   ICMP Type 13, or TIMESTAMP requests are often used by attackers to determine whether or not a host is active on the network. Additionally, since Microsoft operating systems do not respond to Type 13 requests, (http://support.microsoft.com/default.aspx?scid=kb;EN-US;q156165),
   if an attacker receives a response, he can be sure that the host is NOT a Microsoft box.

5. **Attack mechanism:**
   Immediately these packets struck me as odd. A URL in the payload of an ICMP packet? A quick hop to the embedded URL:
   (http://www.cis.upenn.edu/~anagnost/cing.html)
   revealed the following:



Currently this page is still active and no additional information is provided.

6. **Correlations:**
   CVE Candidate CAN-1999-0524 – ICMP information such as netmask and timestamp is allowed from arbitrary hosts.

   http://www.mynetwatchman.com/ - Incident # (2370760)

7. **Evidence of active targeting:**
   The URL from the ICMP indicates that the IP addresses are chosen from both specific and from random hosts. It is my belief that our .89 host was randomly targeted.

8. **Severity:**
   Target Criticality:2 (Corporate Webserver)
   Attack Lethality: 2 (Reconnaissance or information gathering)

System Countermeasures: 2 (System responds to ICMP TIMESTAMP requests)
Network Countermeasures: 4 (Firewall blocked inbound ICMP TIMESTAMP packets to this host)
Attack Severity: -2. (2 + 2) – (2 + 4) = -2

9. **Defensive recommendation:**
   Current defenses are appropriate, firewall has been tightened down to explicitly block all inbound ICMP requests.

10. **Multiple choice test question:**
    ICMP TIMESTAMP requests are often used by attackers to determine:
    A. Latency between hosts.
    B. Maximum Segment Size of receiving host.
    C. ICMP Redirect datagram type.
    D. Host OS type.
    Answer: D (Host OS Type) Since ICMP TIMESTAMP has not been implemented in Microsoft's stack, a successful reply to an ICMP TIMESTAMP pack indicates the respond host is not a Microsoft OS.

Detect #5 – Anomalous Traffic (SRC Port 5635 → DST Port 0)

TCPDump Output

```
15:31:59.153606 63.254.25.173.32835 > xxx.xxx.xxx.17.259: R 10752:10821(69) win 0 (DF)
15:32:05.554791 63.254.25.173.5635 > xxx.xxx.xxx.17.0: FR 1526792192:1526792284(92) win 36727 urg 26181 (DF)
15:32:05.554791 63.254.25.173.5635 > xxx.xxx.xxx.17.0: FR 1526792192:1526792284(92) win 36727 urg 26181 (DF)
15:32:05.788422 63.254.25.173.5635 > xxx.xxx.xxx.17.0: FRWE 1526792192:1526792260(68) win 30892 <[bad opt]> (DF)
15:32:26.524954 63.254.25.173.5635 > xxx.xxx.xxx.17.0: SRPWE 1526792192:1526792268(76) win 22834 urg 3417 (DF)
15:32:35.026687 63.254.25.173.5635 > xxx.xxx.xxx.0: SFRPW 1526792192:1526792236(44) win 53870 <[bad opt]> (DF)
15:32:36.020559 63.254.25.173.5635 > xxx.xxx.xxx.0: SFRWE 1526792192:1526792288(96) ack 1459814577 win 48431 urg 12918 (DF)
```

1. **Source of Trace:**
   NIDS located outside a firewall that I manage.

2. **Detect was generated by:**
   Manual TCPDump query of Snort output based on destination port 0 accesses.

3. **Probability the source address was spoofed:**
   Low. This trace appears to be a remote host detection scan, the attacker would need to receive a response from the host.

4. **Description of attack:**
   At first glance this looks to be an OS detection scan. (Unique source ports "5635", common destination port "0", and a christmas tree of TCP Flags.) While the signature of this attack does not point to a specific tool or exploit, a thorough search of google reveals many other posts requesting information on similar traces (Source port 5635 → Destination port 0)

5. **Attack mechanism:**
   The actual attack mechanism here is unknown, but it can be assumed that a malicious tool is being employed to scan hosts on the Internet in a possible attempt to determine their OS Type.

6. **Correlations:**
   The following detects were reported by Matt Fearnow on the Daily Report of January, 09 2001 http://www.sans.org/y2k/010901-1300.htm

   Detect 5: Similar pattern to detect 3, with the first packet being a bad TCP header length with FIN and SYN flags set, and the second packet showing invalid transport field six seconds later. The source ports are very similar to detect 3; detect 3 source port is 32811, detect 5 is 32808; destination TCP port is 259 in both cases; and the source ports for the invalid transport field messages are both 5635 with destination port 0.

   Dec 27 14:22:43 [firewall.ip.address] %PIX-5-500003: Bad TCP hdr length (hdrlen=0, pktlen=42) from 209.255.81.199/32808 to cidr.net.addr.98/259, flags: FIN SYN , on interface outside
   Dec 27 14:22:49 [firewall.ip.address] %PIX-4-500004: Invalid transport field for protocol=6, from 209.255.81.199/5635 to cidr.net.addr.98/0

   Detect 15: Possible attempt at OS fingerprint. Risk: low Jan 01 21:56:59 [firewall.ip.address] %PIX-4-500004: Invalid transport field for protocol=6, from 63.254.150.21/5635 to cidr.net.addr.98/0

   It is interesting to note that Detect # 15 above identified the attacker originating from the same Class-B subnet range that we observed in our traces.

7. **Evidence of active targeting:**
   With very little information known about the actual "intent" of the attack, it must be assumed that our host was actively targeted.

8. **Severity:**
   Target Criticality: 2 (Corporate Webserver)
   Attack Lethality: 4 (It appears that this is a reconnaissance probe, so I normally would list the Attack Lethality as a 2 or 3, however with so little information correlated I have raised the lethality to a 4)
   System Countermeasures: 4 (System is up to date with all current patches applied)
   Network Countermeasures: 4 (Firewall silently dropped port 259 and port 0 requests
   Attack Severity: -2 (2 + 4) – (4 + 4) = -2

9. **Defensive recommendation:**
   Defenses are appropriate, systems are hardened and the local firewall prevented these reconnaissance attacks.

10. **Multiple choice test question:**
    The following trace is anomolous because?
    15:32:05.554791 63.254.25.173.5635 > 199.222.100.17.0: FR 1526792192:1526792284(92) win 36727 urg 26181 (DF)
    15:32:05.554791 63.254.25.173.5635 > 199.222.100.17.0: FR 1526792192:1526792284(92) win 36727 urg 26181 (DF)

15:32:05.788422 63.254.25.173.5635 > 199.222.100.17.0: FRWE 1526792192:1526792260(68) win 30892 <[bad opt]> (DF)
15:32:26.524954 63.254.25.173.5635 > 199.222.100.17.0: SRPWE 1526792192:1526792268(76) win 22834 urg 3417 (DF)
15:32:35.026687 63.254.25.173.5635 > 199.222.100.17.0: SFRPW 1526792192:1526792236(44) win 53870 <[bad opt]> (DF)
15:32:36.020559 63.254.25.173.5635 > 199.222.100.17.0: SFRWE 1526792192:1526792288(96) ack 1459814577 win 48431 urg 12918 (DF)

A. A static source port of 5635
B. A destination port of 0
C. Anomolous TCP flag combinations
D. All of the above

Answer: D (All of the above)

Assignment 3 – "Analyze This"

Below are the results of our findings during our five day audit of GIAC University. During the period of January 4th through January 8th 2002 the following logs were generated and analyzed.

Alerts, OOS, and Scans from January 4th-8th 2002

Alerts	OOS	Scans
alert.020104	oos_Jan.4.2002	Scans.020104
alert.020105	oos_Jan.5.2002	Scans.020105
alert.020106	oos_Jan.6.2002	Scans.020106
alert.020107	oos_Jan.7.2002	Scans.020107
alert.020108	oos_Jan.8.2002	Scans.020108

Detects:

An initial analysis of the log data resulted in the following summary of alerts identified at GIAC University:

Alerts	Alert Description
33556	ICMP traceroute
22409	connect to 515 from inside
22373	spp_http_decode: IIS Unicode attack detected
15419	MISC Large UDP Packet
13338	SNMP public access
2362	INFO MSN IM Chat data
1915	INFO - ICQ Access
1848	High port 65535 udp - possible Red Worm – traffic
1285	ICMP Router Selection
1078	SMB Name Wildcard
784	ICMP Fragment Reassembly Time Exceeded
586	ICMP Echo Request L3retriever Ping
412	Watchlist 000220 IL-ISDNNET-990517
329	ICMP Destination Unreachable (Communication Administratively Prohibited)
189	FTP DoS ftpd globbing
181	WEB-MISC Attempt to execute cmd
150	Null scan!

138	SCAN Proxy attempt
99	spp_http_decode: CGI Null Byte attack detected
96	WEB-CGI scriptalias access
82	ICMP Echo Request Nmap or HPING2
82	EXPLOIT x86 NOOP
65	ICMP Echo Request Windows
58	TCP SRC and DST outside network
48	Possible trojan server activity
36	ICMP Destination Unreachable (Protocol Unreachable)
22	INFO FTP anonymous FTP
21	INFO Possible IRC Access
18	INFO Inbound GNUTella Connect accept
17	Incomplete Packet Fragments Discarded
15	INFO - Possible Squid Scan
14	INFO Napster Client Data
14	EXPLOIT x86 setuid 0
13	WEB-MISC 403 Forbidden
13	High port 65535 tcp - possible Red Worm - traffic
12	WEB-IIS _vti_inf access
12	FTP passwd attempt
11	WEB-FRONTPAGE _vti_rpc access
10	ICMP Echo Request Cisco Type.x
9	WEB-IIS view source via translate header
8	Attempted Sun RPC high port access
7	IDS552/web-iis_IIS ISAPI Overflow ida nosize
7	EXPLOIT x86 setgid 0
6	SCAN FIN
6	Port 55850 udp - Possible myserver activity - ref. 010313-1
5	x86 NOOP - unicode BUFFER OVERFLOW ATTACK
5	WEB-CGI formmail access
5	MISC traceroute
5	INFO Inbound GNUTella Connect request
5	Back Orifice
4	Watchlist 000222 NET-NCFC
4	Tiny Fragments - Possible Hostile Activity
4	SCAN XMAS
4	SCAN Synscan Portscan ID 19104
4	NMAP TCP ping!
4	EXPLOIT x86 stealth noop
4	EXPLOIT NTPDX buffer overflow
3	WEB-IIS Unauthorized IP Access Attempt
3	SYN-FIN scan!
3	ICMP Echo Request CyberKit 2.2 Windows
2	WEB-MISC compaq nsight directory traversal
2	SUNRPC highport access!
2	MISC PCAnywhere Startup
2	INFO Outbound GNUTella Connect accept
1	X11 outgoing

1	WEB-CGI redirect access
1	TFTP - Internal UDP connection to external tftp server
1	TFTP - External UDP connection to internal tftp server
1	Queso fingerprint
1	IDS50/trojan_trojan-active-subseven
1	FTP CWD - possible warez site

To better help you understand the impact of these alerts, we have compiled a listing of the top 10 Alert definitions:

Top 10 Alert Definitions:

1. ICMP traceroute:
Snort reported a total of 33556 unique ICMP traceroute alerts. 33548 of these originated from MY.NET.5.202 destined for MY.NET.5.1. A traceroute is used to determine the number of hops and specific route a packet will take to reach a host. It is extremely odd that an internal host would be performing a traceroute against another internal host on the same subnet (MY.NET.5.X). If we reference the actual Snort rule used for ICMP tracerourte,

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP traceroute
";ttl:1;itype:8; reference:arachnids,118;)
```

we see that this rule will trigger on any ICMP ECHO Request (Type 8) with a TTL of 1. While it is possible that an actual traceroute was launched between the hosts, it is highly unlikely that 33548 traceroutes were executed over a 5 day period. Except for a handful of portscans and only 9 alerts (all SNMP public), there was no other anomalous activity seen directed at or from this host. We recommend that GIAC University investigate this machine for possible network corruption.

2. connect to 515 from inside:
Port 515 is commonly associated with a UNIX printer port and has been historically compromised for root level access. Of the 22409 alerts generated, all except one was directed at host MY.NET.150.198. The source IP's were a fairly even distribution of hosts in the internal MY.NET network, indicating that MY.NET.150.198 is probably a valid UNIX print server. Please confirm the existence of this print server.

3. spp_http_decode: IIS Unicode attack detected:
The IIS Unicode attack alert is a very common "False Positive" This alert attempts to identify hostile traffic by interpreting unicode data as an attempt to obfuscate an attack. Based on the varying distribution of source and destination addresses these alerts appear to be false positives. GIAC University may want to investigate removing this alert from their configuration.
http://archives.neohapsis.com/archives/snort/2001-08/0528.html

4. MISC Large UDP Packet:
Initially, the only information I could find on "MISC Large UDP Packet" alerts indicated that large windows install bases would have a large falsepositive hit rate for this alert based on NetBIOS traffic running on ports 137 and 138:
http://archives.neohapsis.com/archives/snort/2001-02/0080.html
However, a quick grep of all of the MISC Large UDP Packet alerts for port 137 or 138 yielded no matches.

The only other correlation I could find detailed a problem with the MISC Large UDP Packet alert on the Snort-Users mailing list:
http://archives.neohapsis.com/archives/snort/2000-06/0335.html
This post identified a problem in the rule syntax whereby quotes surrounding the byte size of the packet were erroneously triggering the alert, generating false positives.

To accurately identify the source of these 15419 alerts we would need to acquire full packet traces for detailed investigation.

5. SNMP public access:
SNMP is typically used for remote network management and has been historically plagued with a multitude of vulnerabilities. A quick search of CVE returns the following entries/candidates of varying severity:

CVE-1999-029 CVE-1999-0472 CVE-2000-022 CVE-2000-0379
CVE-2000-0515 CVE-2000-1058 CAN-1999-0186 CAN-1999-0254
CAN-1999-0499 CAN-1999-0516 CAN-1999-0517 CAN-1999-0615

CAN-1999-0792 CAN-1999-0815 CAN-1999-1042 CAN-1999-1126
CAN-1999-1245 CAN-1999-1335 CAN-1999-1460 CAN-1999-1513
CAN-2000-0147 CAN-2000-0885 CAN-2000-0955 CAN-2000-1157
CAN-2000-1192 CAN-2001-0046 CAN-2001-0236 CAN-2001-0352
CAN-2001-0380 CAN-2001-0470 CAN-2001-0487 CAN-2001-0514
CAN-2001-0552 CAN-2001-0564 CAN-2001-0566 CAN-2001-0711
CAN-2001-0840 CAN-2001-0888

Currently of the 13328 alerts destined for port 161 (SNMP) all source and destination hosts are internal to GIAC University. It is possible that SNMP is being employed for network management at the university, however if this is the case only authorized hosts should be making SNMP requests.

6. INFO MSN IM Chat data:
MSN IM is a protocol used to chat between computers on the internet. While not malicious in and of itself, chat programs are often employed by the underground community to exchange correspondence. All IM Chat alerts identified at GIAC University are communicating with known registered chat servers at Hotmail.
MS Hotmail (NETBLK-HOTMAIL)
1065 La Avenida
Mountain View, CA 94043

US

Netname: HOTMAIL
Netblock: **64.4.0.0 - 64.4.63.255**

Coordinator:
Myers, Michael (MM520-ARIN) icon@HOTMAIL.COM
650-693-7072

Domain System inverse mapping provided by:

NS1.HOTMAIL.COM 216.200.206.140
NS3.HOTMAIL.COM 209.185.130.68

Record last updated on 09-Jan-2001.
Database last updated on 28-Jan-2002 19:56:48 EDT.

Please verify with your acceptable use policy whether or not IM Chat is an acceptable use of the Internet at GIAC University.

7. INFO - ICQ Access

Much like AOL IM, and MSN IM, ICQ Access is not suspicious in nature. What may be suspicious, although GIAC needs to confirm access with their AUP, is the amount of chat activity that is taking place. Between January 4th and January 8th a total of 4277 alerts indicating Internet chat activity took place. Below is a list of all internal host addresses currently engaging in Chat activity. This includes both MSN IM and ICQ Access:

Chat Count	Internal Host
233	MY.NET.150.232
213	MY.NET.150.241
149	MY.NET.153.113
124	MY.NET.153.45
81	MY.NET.153.125
72	MY.NET.153.193
51	MY.NET.153.46
27	MY.NET.153.108
26	MY.NET.150.143
25	MY.NET.88.181
25	MY.NET.88.165
22	MY.NET.153.119
18	MY.NET.88.163
16	MY.NET.150.165
10	MY.NET.150.246
6	MY.NET.153.112
1	MY.NET.153.126

8. High port 65535 udp - possible Red Worm - traffic

The Red Worm, or more commonly referred to as the Adore Worm commonly binds a trojan backdoor to UDP port 65535 of the infected host. By tickling the host's port with an appropriately sized ICMP packet, the compromised host will open a backdoor on port 65535.
http://www.sans.org/y2k/adore.htm

http://groups.google.com/groups?hl=en&selm=3AE3D2E9.D65479D5%40bell-bird.com.au – Lengthy Usenet thread detailing an analysis of the Adore Worm

9. ICMP Router Selection

ICMP Router Selection or Router Discovery is a method of querying a multicast address of 224.0.0.2 to determine available routes and to query for available routers. The 1285 Alerts generated by the internal MY.NET hosts all were directing to 224.0.0.2 and are therefore deemed appropriate traffic. This alert rule should be considered for removal due to the high false positive hit rate.

http://www.microsoft.com/windows2000/en/server/help/default.asp?url=/windows2000/en/server/help/sag_RRAS-Ch2_14.htm

http://www.isi.edu/in-notes/iana/assignments/multicast-addresses

11. SMB Name Wildcard:

SMB Name Wildcard is triggered whenever a host performs a NetBIOS name lookup of a remote host, using Windows natively or using Samba or CIFS. Due to numerous vulnerabilities discovered on Windows hosts, it is recommended that NetBIOS traffic be restricted to internal hosts. Our analysis of GIAC University has determined that all SMB Name Wildcard traffic is in fact internal traffic and therefore is not deemed malicious. It is our recommendation that the SMB Name Wildcard rule be modified to only alert on SMB traffic to or from "external" hosts. This will highly reduce the rate of false positives for this alert.

http://www.sans.org/y2k/061500.htm

http://archives.neohapsis.com/archives/snort/2000-08/0289.html

OOS (Out of Spec) Traffic Analysis:

OOS or Out of Specification packets are those packets which are anomalous by nature and do not conform to RFC specifications. Traditionally OOS packets are either crafted packets, misconfigured or broken network equipment (ie. routers).

Our analysis of OOS packets observed at GIAC University identified the following Top 10 Source IP addresses of OOS packets.

Top 10 OOS by SRC IP

Count	SRC IP
16	144.122.42.38
10	195.132.240.41
9	24.158.117.251
8	130.104.19.73
6	4.61.46.216
5	192.116.55.2
3	66.121.247.51
1	68.41.218.102

1	66.68.58.90
1	62.198.104.68

While we can't be sure of malicious intent, these hosts definitely require further investigation:

OOS Source IP - 144.122.42.38:

Middle East Technical University (NET-METU-NET)
METU Computer Center Inonu Bulvari - ODTU
Ankara, 06531
TR

Netname: METU-NET
Netblock: 144.122.0.0 - 144.122.255.255

Coordinator:
METU Hostmaster (MH2-ORG-ARIN) hostmaster@METU.EDU.TR
+90 312 2103330
Fax- +90 312 2101120

Domain System inverse mapping provided by:

NS1.METU.EDU.TR 144.122.199.90
NS2.METU.EDU.TR 144.122.199.93
NS1-AUTH.SPRINTLINK.NET 206.228.179.10
AUTH60.NS.UU.NET 198.6.1.181

Record last updated on 27-Oct-1998.
Database last updated on 28-Jan-2002 19:56:48 EDT.

A nslookup of **144.222.42.38** yielded the following:

Name: 1207.odtukent.metu.edu.tr
Address: 144.122.42.38

While 144.222.42.38 appeared 16 times in our OOS logs, this host also appeared in the scan and alert logs approximately 200 more times. All scans were directed at MY.NET.88.162 with the majority directed at port 1214. TCP and UDP port 1214 is commonly associated with the filesharing application Kazaa where users connect to servers that facilitate the sharing of files (music/video/etc.) over the Internet. There have been several postings on underground websites detailing how to "hack" Kazaa and other Peer to Peer (p2p) applications. One specific link, http://www.hackers.com/new/viewarticle.php?aid=46, details how to scan for Kazaa hosts on port 1214, and shows step by step how to subvert the Kazaa application to directly access a users files.

It is quite probable that MY.NET.88.162 is engaging in p2p filesharing, and is unwittingly illiciitng access to both his/her computer and GIAC's network. P2p applications can consume large amounts of bandwith, so it is our recommendation that GIAC consider blocking all p2p application functionality at the University.

As these were only scans and not actual host accesses, it appears as 144.222.42.38 was attempting to portscan and possible perform OS detection on our host.

OOS Source IP - 195.132.240.41

inetnum: 195.132.0.0 - 195.132.255.255
netname: FR-CYBERCABLE-960620
descr: LYONNAISE COMMUNICATIONS
PROVIDER Local Registry
country: FR
admin-c: LC220-RIPE
tech-c: LC224-RIPE
status: ALLOCATED PA
mnt-by: RIPE-NCC-HM-MNT
mnt-lower: AS6678-MNT
mnt-routes: AS6678-MNT
changed: hostmaster@ripe.net 19960620
changed: hostmaster@ripe.net 20020108
source: RIPE
route: 195.132.0.0/16
descr: NOOS
descr: Lyonnaise Communications
descr: Paris, FRANCE
origin: AS6678
mnt-by: AS6678-MNT
changed: ripe@euroconnect.fr 20010413
source: RIPE
role: Lyonnaise Communications Administrative Role Account
address: Lyonnaise Communications
address: 20 place des vins de France
address: 75 614 PARIS Cedex 12
address: FRANCE
e-mail: hostmaster@noos.net
admin-c: FG3404-RIPE
admin-c: PJ187-RIPE
admin-c: DT3675-RIPE
admin-c: DG2553-RIPE
admin-c: MM266-ARIN
admin-c: OM28-RIPE
admin-c: JD208-RIPE
tech-c: LC224-RIPE
nic-hdl: LC220-RIPE
notify: hostmaster@noos.net
mnt-by: AS6678-MNT
changed: marc-olivier.mehu@noos.fr 20011130
changed: marc-olivier.mehu@noos.fr 20011203
source: RIPE
role: Lyonnaise Communications Technical Role Account
address: Lyonnaise Communications
address: 20 place des vins de France
address: 75 614 PARIS Cedex 12
address: FRANCE
e-mail: hostmaster@noos.net
admin-c: LC220-RIPE
tech-c: PJ187-RIPE
tech-c: DT3675-RIPE
tech-c: MM266-ARIN
tech-c: OM28-RIPE
tech-c: JD208-RIPE
nic-hdl: LC224-RIPE
notify: hostmaster@noos.net
mnt-by: AS6678-MNT
changed: marc-olivier.mehu@noos.fr 20011130
changed: marc-olivier.mehu@noos.fr 20011203
source: RIPE

An nslookup performed against 195.132.240.41 resolved to the local cablemodem user host in France:

```
Name:    r240m41.cybercable.tm.fr
Address:  195.132.240.41
```

195.132.240.41 also performed extensive portscans / OS detection against the same host, MY.NET.88.162, primarily directed again at Kazaa's port 1214.

OOS Source IP - 24.158.117.251

```
Charter Communications, Inc. (NETBLK-CHARTER-NET-2BLK) CHARTER-NET-2BLK
                              24.158.0.0 - 24.158.255.255
Charter Communications (NETBLK-24-158-117-NET-TN) 24-158-117-NET-TN
                              24.158.117.0 - 24.158.117.255

To single out one record, look it up with "!xxx", where xxx is the
handle, shown in parenthesis following the name, which comes first.

The ARIN Registration Services Host contains ONLY Internet
Network Information: Networks, ASN's, and related POC's.
Please use the whois server at rs.internic.net for DOMAIN related
Information and whois.nic.mil for NIPRNET Information.


Name:    kpt-c-24-158-117-251.chartertn.net
Address:  24.158.117.251
```

24.158.117.251, unlike the previous two OOS source IP addresses, was found in the scan and alert logs scanning MY.NET.150.204 however the common thread continues to be port 1214.

OOS Source IP - 130.104.19.73:

```
   Universite Catholique de Louvain (NET-UCLOUVAIN)
     Place de l'Universite, 1
     Louvain-la-Neuve, B-1348
     BE

     Netname: UCLOUVAIN
     Netblock: 130.104.0.0 - 130.104.255.255

     Coordinator:
       Fontaine, Alain  (AF194-ARIN)  fontaine@sri.ucl.ac.be
       +32 10 472625 (FAX) +32 10 472650

     Domain System inverse mapping provided by:

     NS1.SRI.UCL.AC.BE          130.104.1.1
     NS2.SRI.UCL.AC.BE          130.104.1.2
     NS3.SRI.UCL.AC.BE          130.104.254.1
     NS.BELNET.BE              193.190.198.10 193.190.198.2
     NS2.KULNET.KULEUVEN.AC.BE    134.58.127.1

     Record last updated on 22-Nov-1999.
     Database last updated on  28-Jan-2002 19:56:48 EDT.
```

130.104.19.73 resolves to the following name:

```
Name:    19-73.CampusNet.ucl.ac.be
Address:  130.104.19.73
```

This host, along with (195.132.240.41 and 144.122.42.38) scan and alert logs was actively scanning GIAC's internal host MY.NET.88.162.

OOS Source IP - 4.61.46.216:

GENUITY (NET-GNTY-4-0)
3 Van de Graaff Dr.
Burlington, MA 01803
US

Netname: GNTY-4-0
Netblock: 4.0.0.0 - 4.255.255.255
Maintainer: GNTY

Coordinator:
Soulia, Cindy (CS15-ARIN) csoulia@genuity.net
800-632-7638

Domain System inverse mapping provided by:

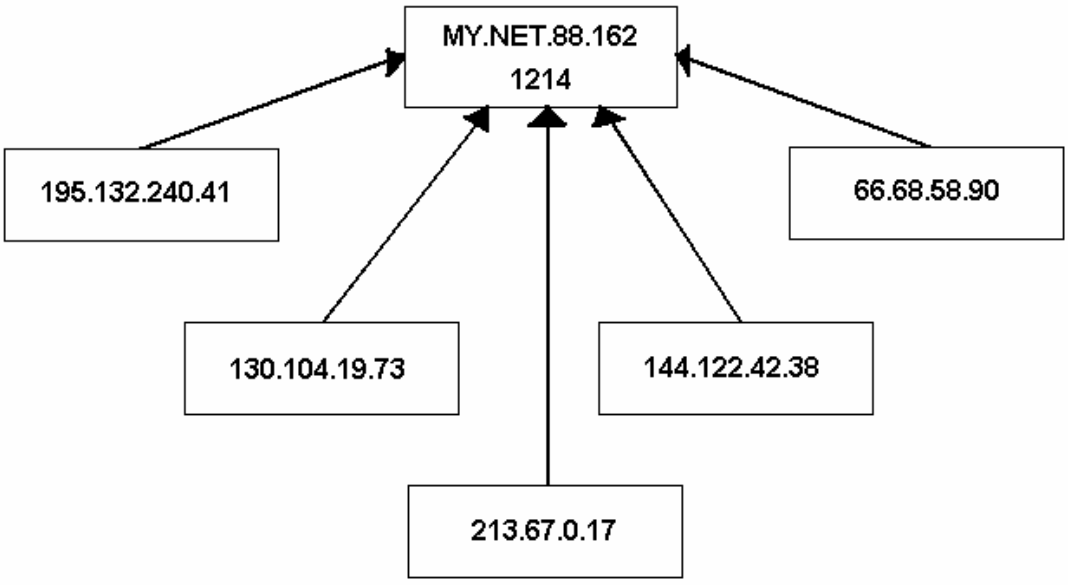
NIC.NEAR.NET 192.52.71.4
VIENNA1-DNS-AUTH1.BBNPLANET.COM 4.1.16.4
NIC3.BARRNET.NET 131.119.245.6

Record last updated on 24-Sep-2001.
Database last updated on 28-Jan-2002 19:56:48 EDT.

4.61.46.216 was not resolvable via DNS, but was actively scanning our internal host MY.NET.150.143

Based on the above OOS data, we recommend that GIAC University review their logs on the following hosts:
MY.NET.150.143
MY.NET.88.162
MY.NET.150.204

To further validate the port 1214 accesses at GIAC University. We have generated a Link Graph of OOS data to detail the scope of the Kazaa activity.



After identifying MY.NET.88.162 as a potential host for further investigation, an explicit search was performed against the alert logs for any instance where this

host was referenced. The following alert logs were identified as possibly malicious:

```
alert.020106:01/06-04:17:30.354187  [**] EXPLOIT x86 setuid 0 [**] 24.178.184.42:1214 -
> MY.NET.88.162:4823
alert.020106:01/06-04:17:57.465872  [**] EXPLOIT x86 setuid 0 [**] 24.178.184.42:1214 -
> MY.NET.88.162:4823
```

This alert is generated when the Snort sensor detects specific machine code indicating a possible compromise due to a buffer overflow (using x86 nop's). This alert has a high occurrence of false positive, but care must be taken to address this host for possible compromise.

Top Talkers:

In an attempt to normalize 5 days worth of data we have identified the Top Talkers both internally and externally.

Top 10 Scans by SRC IP :

Count	SRC IP
369031	MY.NET.60.43
49237	MY.NET.6.49
46390	MY.NET.6.50
26618	MY.NET.6.52
25077	MY.NET.6.51
24324	MY.NET.6.45
23472	MY.NET.150.143
16012	MY.NET.6.48
12359	MY.NET.150.209
8300	MY.NET.6.60

Top 10 Scans by Protocol

Count	Proto
808932	UDP
152700	SYN
107	NULL
73	VECNA
50	NOACK
50	INVALIDACK
47	UNKNOWN
10	SYNFIN
4	FIN
3	XMAS
3	NMAPID
3	FULLXMAS
1	SPAU

Top 10 Scans by DST Port

Count	Port

112680	80
65944	7001
48424	53
44935	6970
34505	7000
24084	6257
20902	0
13353	7003
12194	1214
11612	6699

Top 10 Scans by DST IP

Count	DST IP
26873	MY.NET.1.3
20266	MY.NET.1.4
14837	MY.NET.153.189
14720	MY.NET.153.142
12868	MY.NET.153.203
12772	MY.NET.153.204
12351	MY.NET.60.43
12225	MY.NET.153.143
11924	MY.NET.153.209
11586	MY.NET.6.45

Top 10 OOS by SRC IP

Count	SRC IP
16	144.122.42.38
10	195.132.240.41
9	24.158.117.251
8	130.104.19.73
6	4.61.46.216
5	192.116.55.2
3	66.121.247.51
1	68.41.218.102
1	66.68.58.90
1	62.198.104.68

Top 10 OOS by DST IP

Count	DST IP
36	MY.NET.88.162
13	MY.NET.150.204
8	MY.NET.150.143
5	MY.NET.153.206
1	MY.NET.153.148
1	MY.NET.150.220

Analysis Process:

The bulk of the analysis for this section of the practical was performed using the following command line UNIX tools:

sed, grep, awk, cut, sort, uniq, cat.

Here is an example of how I gathered totals for the Top 10 Alert Source IP Addresses:

```
cat alert.0112*|grep -v "spp_portscan:"|cut -d ']' -f 3|cut -d ' ' -f 2|cut -d ':' -f 1
|sort |uniq -c |sort -r |head –10
```

Whois lookups were performed using the following perl script, “jwhois.pl” by Josh Grubman:

```
#!/usr/bin/perl
# jwhois 1.3 - the universal whois client - 5/21/99
# if you find this script useful, drop me a line and let me know
# - josh grubman <jg@false.net>

### The IO::Socket module lets us do this without a whois binary. This
### provides us with Win32 compatibility. This has been tested with
### ActiveState perl build 516e, it may or may not work with older
### versions and should have no problems with a current release.

use IO::Socket;

### The -h flag will force a query against a specified server. This
### provides a legacy-style whois interface.

if ("$ARGV[0]" =~ "\-h") {
  $sock = IO::Socket::INET->new("$ARGV[1]:43") || die $!;
  print $sock "$ARGV[2] \n";
  while (<$sock>) {
    print $_;
  }
  close $sock;
  exit;
}

$hostname="$ARGV[0]";

### Handles which end in -DOM, -ORG, or two/three letters followed by
### a numeric string are queried as InterNIC handles.

if ($hostname =~ /\-dom/i || $hostname =~ /\-org/i)  {
  $sock = IO::Socket::INET->new("rs.internic.net:43") || die $!;
  print $sock "$hostname \n";
  while (<$sock>) {
    print $_;
  }
  close $sock;
  exit;
}

### Figure out what TLD we're looking at and query the appropriate
### registry based on a response from whois-servers.net

@tld = split (/\./,$hostname);
@tld = reverse @tld;
if (! ($tld[0] =~ /\d{1,3}/) ) {
  $sock = IO::Socket::INET->new("$tld[0].whois-servers.net:43") || die &fail;
  print $sock "$hostname \n";
  while (<$sock>) {
    print $_;
  }
  close $sock;
exit;
```

```
}

### Query ARIN for network assignments. If a network has been subassigned
### to another authority, query the appropriate whois server

### Add additional servers to this hash as needed - these are used when
### ARIN's records indicate a netblock has been "further assigned"

%whois_servers = ("RIPE"=>"whois.ripe.net","APNIC"=>"whois.apnic.net");

$sock = IO::Socket::INET->new("whois.arin.net:43") || die &fail;
print $sock "$hostname \n";
while (<$sock>) {
  $extraquery = $1 if (/Contact information can be found in the (\S+)\s+database/);
  push(@lines,$_);
}
if ($extraquery) {
  undef(@lines);
  $sock = IO::Socket::INET->new("$whois_servers{$extraquery}:43") || die $!;
  print $sock "$hostname\n";
  while (<$sock>) {
  push(@lines,$_);
  }
}
foreach $line (@lines) {
  print $line;
}

### Error message which indicates that no valid whois server was found

sub fail {
print <<EOF;

no information found.

jwhois currently supports whois queries for network assignments, ASNs,
and most top level domains. TLD information is not available from some
international registries who do not provide a public whois server.

EOF
exit;
}
```

Last Updated: August 17th, 2017

Upcoming Training

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
Baltimore Fall 2017 - SEC503: Intrusion Detection In-Depth	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
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
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
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 Munich December 2017	Munich, Germany	Dec 04, 2017 - Dec 09, 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 Northern VA Spring - Tysons 2018	Tysons, VA	Mar 17, 2018 - Mar 24, 2018	Live Event
SANS 2018	Orlando, FL	Apr 03, 2018 - Apr 10, 2018	Live Event
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced