Global Information Assurance Certification Paper

GIAC LevelTwo Intrusion Detection In Depth
Practical Assignment for SANS Baltimore 2001
Version 2.9

Kevin Liston

Table of Contents

Quick overview of log formats used in these analyses:

SNORT (http://www.snort.org)

Marty Roesch's SNORT operates in three basic modes: sniffer, logger, and NIDS. The following is an example output from SNORT in sniffer mode (called as `snort -dv`.) The output differs slightly between IP protocols, the most common are TCP, UDP, and ICMP.

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

06/11-20:56:00.366377 11.22.33.44:33124 -> 111.122.133.144:80
TCP TTL:48 TOS:0x0 ID:43935 IpLen:20 DgmLen:378
***AP*** Seq: 0xEE794A2D  Ack: 0xE6C68890  Win: 0xB68  TcpLen: 32
TCP Options (3) => NOP NOP TS: 1072614 1295875663
47 45 54 20 2F 74 6F 6B 79 6F 2F 49 6D 61 67 65  GET /tokyo/Image
73 2F 50 69 63 5F 42 75 74 74 6F 6E 5F 4D 65 6E  s/Pic_Button_Men
75 5F 33 45 2E 67 69 66 20 48 54 54 50 2F 31 2E  u_3E.gif HTTP/1.
30 0D 0A 41 63 63 65 70 74 3A 20 2A 2F 2A 0D 0A  0..Accept: */*..
52 65 66 65 72 65 72 3A 20 68 74 74 70 3A 2F 2F  Referer: http://
77 77 77 2E 66 63 74 6D 2E 63 6F 2E 6A 70 2F 74  www.fctm.co.jp/t
6F 6B 79 6F 2F 46 72 6F 6E 74 4A 2E 68 74 6D 0D  okyo/FrontJ.htm.
0A 41 63 63 65 70 74 2D 4C 61 6E 67 75 61 67 65  .Accept-Language
3A 20 6A 61 0D 0A 41 63 63 65 70 74 2D 45 6E 63  : ja..Accept-Enc
6F 64 69 6E 67 3A 20 67 7A 69 70 2C 20 64 65 66  oding: gzip, def
6C 61 74 65 0D 0A 55 73 65 72 2D 41 67 65 6E 74  late..User-Agent
3A 20 4D 6F 7A 69 6C 6C 61 2F 34 2E 30 20 28 63  : Mozilla/4.0 (c
6F 6D 70 61 74 69 62 6C 65 3B 20 4D 53 49 45 20  ompatible; MSIE
35 2E 30 3B 20 57 69 6E 64 6F 77 73 20 4E 54 29  5.0; Windows NT)
0D 0A 48 6F 73 74 3A 20 77 77 77 2E 66 63 74 6D  ..Host: www.fctm
2E 63 6F 2E 6A 70 0D 0A 58 72 6F 78 79 2D 43 6F  .co.jp..Xroxy-Co
6E 6E 65 63 74 69 6F 6E 3A 20 4B 65 65 70 2D 41  nnection: Keep-A
6C 69 76 65 0D 0A 43 61 63 68 65 2D 43 6F 6E 74  live..Cache-Cont
72 6F 6C 3A 20 6D 61 78 2D 73 74 61 6C 65 3D 30  rol: max-stale=0
0D 0A 50 72 61 67 6D 61 3A 20 6E 6F 2D 63 61 63  ..Pragma: no-cac
68 65 0D 0A 0D 0A                                he....

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

06/11-20:56:00.367159 11.22.33.44:33099 -> 111.122.133.144:80
TCP TTL:48 TOS:0x0 ID:43939 IpLen:20 DgmLen:52
***A**** Seq: 0xEE3754F9  Ack: 0xE6B97049  Win: 0xB68  TcpLen: 32
TCP Options (3) => NOP NOP TS: 1072614 1295875648

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

Here is an example of SNORT's output (the IP numbers have been changed to protect the innocent.) The format is as follows:

```
<timestamp> <source-ip>:<source-port> -> <destination-ip>:<destination-port>
<protocol> <time-to-live> <type-of-service> <session-ID> <IP length> <datagram length>
<flags> <sequence number> <ack number> <window size> <TCP length>
<TCP options>
<packet payload>
```

The above example illustrates a couple of sample flag settings (ACK and PUSH in the first, a simple ACK in the second,) and TCP options (timestamp in this case.) The payload is in HEX/ASCII format and begins in the data field of the TCP packet.

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

06/11-20:56:01.070682 11.22.33.44:137 -> 111.122.133.144:137
UDP TTL:127 TOS:0x0 ID:18470 IpLen:20 DgmLen:78
Len: 58
AD C0 00 10 00 01 00 00 00 00 00 00 20 43 4B 41  ............ CKA
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 00 00 21  AAAAAAAAAAAAA..!
00 01                                            ..

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

The above is an example UDP packet with the following format:

```
<timestamp> <source-ip>:<source-port> -> <destination-ip>:<destination-port>
<protocol> <time-to-live> <type-of-service> <session-ID> <IP length> <datagram length>
<UDP length>
<packet payload>
```

An example ICMP capture looks like:

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

06/11-19:14:15.207497 11.22.33.44 -> 111.122.133.144
ICMP TTL:242 TOS:0x0 ID:28759 IpLen:20 DgmLen:28
Type:8  Code:0  ID:0   Seq:0  ECHO

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

Where the format is:

```
<timestamp> <source-ip> -> <destination-ip>
<protocol> <time-to-live> <type-of-service> <session ID> <IP length> <datagram length>
<icmp type> <icmp code> <icmp options> <human-readable icmp type name>
```

In this example we have a ping echo request so it has ICMP options of ID and sequence number.

Checkpoint Firewall-1

Checkpoint's Firewall-1 is commonly used. The source used in these analyses comes from ASCII archive files. They use the following format:

```
num;date;time;orig;type;action;alert;i/f_name;i/f_dir;proto;src;dst;service;s_port;len;rul
e;icmp-type;icmp-code;xlatesrc;xlatedst;xlatesport;xlatedport;reason:;port:;sys_msgs
```

Each log entry is enumerated in the file. When the logs are rotated, the numbering restarts. An example entry would appear as:

```
8;26Jun2001;22:58:38;FW.MY.NET.1;log;accept;;eth-
s1p1c0;inbound;udp;166.93.1.3;12.27.38.36;domain;domain;61;24;;;;;;;;
```

Here, we have event numbered 8 in this particular logfile, date and time are obvious, and FW.MY.NET.1 is the firewall reporting the entry. Type is log in this case, other type is control, which logs events such as log rotation, or rule-pushes. The action given by the rule-base is to accept the packet, it could reject, or deny the packet. Here, there is no alert, this field is user-defined and could contain something like page-out or email-alert. The packet arrived on interface eth-s1p1c0, and is inbound. The example packet is UDP. The format is the same for all protocols, but in the case of UDP or TCP, the icmp-type and icmp-code field will be unused; in the case of ICMP, service and s_port will be unused. The source of the packet is logged in src, the destination in dst. The destination is logged in the service field, and the source port is logged in s_port. The packet length is len. Rule is the rule number that was triggered by the packet. The

xlatesrc, xlatedst, xlatesport and xlatedport are used in the firewall is NATing. Reason, port, and sys_msgs are informational fields populated by the Checkpoint engine to explain the reason for a drop, reject, or other rule activation.

Cisco Access Control List

Cisco equipment with Access Control List capability can log using the syslog facility. This is the standard practice, since routers and switches usually lack the disk-space and backup capabilities of servers. The syslog format appears as:

```
Date Time Hostname SequenceNumber: SyslogFacility: ActivatedACL ActionTaken
TrafficProtocol SourceIP(SourcePort) -> DestinationIP(DestinationPort), NumberOfPackets
```

For TCP or UDP, for ICMP the format is:

```
Date Time Hostname SequenceNumber: SyslogFacility: ActivatedACL ActionTaken
TrafficProtocol SourceIP -> DestinationIP(ICMPcode/ICMPtype), NumberOfPackets
```

An example TCP packet blocked by ACL list 101. The %SEC-6-IPACCESSLOGP is the syslog logging facility used.

```
Apr 16 18:57:09: %SEC-6-IPACCESSLOGP: list 101 denied tcp 212.236.6.2(3213) ->
DMZ.NET.169.48(111), 1 packet
```

A note about log obfuscation:

The detects within this document have been obfuscated and logs sanitized to protect the client, and any innocent victims involved in these incidents. Machines launching the attack, where the probability of IP spoofing is low, are not protected.

Assignment 1

Network Detects

Detect #1: Smurf Variant

Every trace has a story. This capture came from a client's inherited network. They were complaining that customers were having trouble getting to their website. SNORT was employed in sniffer-mode to get a feel for what was going on. A flood of stimuli was coming into the client's network in pairs similar to:

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

06/11-19:14:15.206722 INTENDED.VICTIM.NET.224 -> MY.CLIENTS.NET.0
ICMP TTL:242 TOS:0x0 ID:28759 IpLen:20 DgmLen:28
Type:8  Code:0  ID:0   Seq:0  ECHO

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

06/11-19:14:15.207497 INTENDED.VICTIM.NET.224 -> MY.CLIENTS.NET.255
ICMP TTL:242 TOS:0x0 ID:28759 IpLen:20 DgmLen:28
Type:8  Code:0  ID:0   Seq:0  ECHO

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

These stimuli unfortunately resulted in the following replies in the form of:

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

06/11-19:14:15.328435 MY.CLIENTS.NET.3 -> INTENDED.VICTIM.NET.224
ICMP TTL:255 TOS:0x0 ID:15185 IpLen:20 DgmLen:28 DF
Type:0  Code:0  ID:0  Seq:0  ECHO REPLY

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

06/11-19:14:15.328458 MY.CLIENTS.NET.249 -> INTENDED.VICTIM.NET.224
ICMP TTL:254 TOS:0x0 ID:61522 IpLen:20 DgmLen:28 DF
Type:0  Code:0  ID:0  Seq:0  ECHO REPLY

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

06/11-19:14:15.328483 MY.CLIENTS.NET.123 -> INTENDED.VICTIM.NET.224
ICMP TTL:254 TOS:0x0 ID:15262 IpLen:20 DgmLen:28 DF
Type:0  Code:0  ID:0  Seq:0  ECHO REPLY

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

06/11-19:14:15.328506 MY.CLIENTS.NET.114 -> INTENDED.VICTIM.NET.224
ICMP TTL:254 TOS:0x0 ID:19942 IpLen:20 DgmLen:28 DF
Type:0  Code:0  ID:0  Seq:0  ECHO REPLY

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

06/11-19:14:15.328530 MY.CLIENTS.NET.115 -> INTENDED.VICTIM.NET.224
ICMP TTL:254 TOS:0x0 ID:9825 IpLen:20 DgmLen:28 DF
Type:0  Code:0  ID:0  Seq:0  ECHO REPLY

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

06/11-19:14:15.328554 MY.CLIENTS.NET.120 -> INTENDED.VICTIM.NET.224
ICMP TTL:254 TOS:0x0 ID:50568 IpLen:20 DgmLen:28 DF
Type:0  Code:0  ID:0  Seq:0  ECHO REPLY
```

=+

We see a steady stream of pings to the broadcast and network (which serves as a broadcast address in older implementations of the IP stack,) address of the client's network, which apparently came from INTENDED.VICTIM.NET.224, among others. It appeared that INTENDED.VICTIM.NET.224 was suffering a smurf attack and the client was being used as a smurf-amplifier.

The victim machines were found to be on two different university networks.

I searched for the client's networks on netscan.org and turned up that the network had been identified as a smurf-amplifier.

1. Source of the trace:

These data were captured from a client's network.

2. Detect was generated by:

This capture was made using Snort 1.7 in sniffer-mode on a firewall machine in the client's network.

3. Probability the source address was spoofed:

It is almost certain that the source addresses were spoofed. Otherwise the sender would be creating a self-inflicted denial-of-service. Non-spoofed packets could be used to scan for smurf-amplifiers, but you would expect to see only a handful of packets, not a flood.

4. Description of the attack:

The attacker crafted ICMP Echo Request packets with the victim's IP as the source, and one of my client's broadcast and network addresses. Later on in the attack, spoofed packets arrived with the apparent-source of x.x.255.255, another smurf-amplifier network.

Use of spoofed ICMP Echo requests for denial-of-service is documented in:

CA-1998-01: http://www.cert.org/advisories/CA-1998-01.html
CVE-1999-0513: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0513
IIS X-Force: http://xforce.iss.net/alerts/vol-1_num-5.php#smurf
http://cs.baylor.edu/~donahoo/NIUNet/smurf.html
http://www.pentics.net/denial-of-service/white-papers/smurf.cgi

5. Attack Mechanism:

The attacker leverages many machines against the victim by forging an ICMP Echo Request packet with the victim's IP number as the source and sending the request to the broadcast address of a network that will reply to broadcast-directed pings. The machines on the network will reply to this forged Echo Request with their own Echo Reply, thus amplifying the attack. This enables an attacker with a low bandwidth connection to effectively ping-flood a victim with a much larger bandwidth capacity.

6. Correlations:

Al Veach cited a smurf attack in his practical (Northcutt, Cooper, Fearnow, Frederick, p 214.) He only detected spoofed packets to x.x.x.255, and not x.x.x.0, so I feel that a different tool was employed.

7. Evidence of active targeting:

The attacker probably used an online resource like netscan.org or powertech.po/smurf to perform his reconnaissance for him. Since the number of victim IP numbers was low, he was certainly firing-for-effect directly at them. It was certainly not a broad scan of victims.

8. Severity:

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

As a smurf-amplifier, the target is the client's routers and network resources, both of which are critical, so I give it a 5.

The lethality is medium; large networks can handle this load (the client was being used as a relay for sometime before noticing degradation of service, so I rate it as 3.

Arguably, the system countermeasures are low, since they will reply to these pings, so I rate the system countermeasures as 2.

The network countermeasures were *nil*, so I evaluated it as 1.

This yields an Attack Severity level of (5+3)-(2+1) = 5. It is something that should be remedied as soon as possible to help out the victim of the attack.

As a bonus evaluation, calculating from the point of view of the victim of the smurf-flood:

(5+5)-(3+[1-5]) = range of 6 to 2

The Attack Severity depends on what ICMP filtration you have in place and where it is placed. If they have no ICMP filtering they would have a 1, if they have their upstream blocking ICMP at their border routers, then it's a 5.

9. Defensive recommendation:

A network used as a smurf-amplifier can disable directed-broadcast traffic at their routers. It is good practice to disable this capability on your network as a precaution. In the case of the client, the router was not capable of filtering traffic and handling the load, so a firewall rule was used as a temporary measure until the router was upgraded.

A recipient of a smurf attack has few options but to request their ISP block all ICMP traffic to their network.

Further information involving Cisco equipment is available here:
http://www.cisco.com/warp/public/707/22.html

10. Multiple-choice test question:

```
VICTIM.NET.255.255 -> SMURF.RELAY.NET.255
ICMP TTL:242 TOS:0x0 ID:1234 IpLen:20 DgmLen: 28
Type:8  Code:0  ID:0   Seq:0  ECHO
```

If directed broadcast is enabled at SMURF.RELAY.NET.0/24, and at VICTIM.NET.0.0/16, what will be the ultimate result from the above stimulus packet?

A) A broadcast storm.
B) SMURF.RELAY.NET will suffer a denial of service from VICTIM.NET
C) VICTIM.NET will suffer a denial of service from SMURF.RELAY.NET
D) SMURF.RELAY.NET will ignore the ECHO request.

Answer: C. At the worst case, VICTIM.NET will receive a large number of ICMP replies from SMURF.REALY.NET machines, but the machines on VICTIM.NET will not respond since ICMP can not create more ICMP traffic (excluding ECHO requests, of course.)

Detect #2: Smurf Variant 2 with Fraggle-- Papasmurf

This detect is from the same network and situation as Detect #1, but with different technique and different targeted victims.

Although the analysis for this smurf-relay-attack would be very similar to Detect #1, the tool employed is different. The tool used in Detect #1 sends an empty payload, while this one adds 64 null-bytes, increasing the load upon the victim.

First a flood of stimuli into the client's network and the intended response:

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

06/11-20:56:01.408977 y.y.y.141 -> x.x.x.0
ICMP TTL:238 TOS:0x0 ID:16623 IpLen:20 DgmLen:92
Type:8  Code:0  ID:0   Seq:0  ECHO
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................

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

06/11-20:56:01.378796 y.y.y.141 -> x.x.x.255
ICMP TTL:238 TOS:0x0 ID:16623 IpLen:20 DgmLen:92
Type:8  Code:0  ID:0   Seq:0  ECHO
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................

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

06/11-20:56:01.378852 x.x.x.3 -> y.y.y.141
ICMP TTL:255 TOS:0x0 ID:2878 IpLen:20 DgmLen:92 DF
Type:0  Code:0  ID:0  Seq:0  ECHO REPLY
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................

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

These packets were arriving as well:

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

06/11-20:56:01.397834 y.y.y.141:50745 -> x.x.x.255:7
UDP TTL:238 TOS:0x0 ID:16623 IpLen:20 DgmLen:92
Len: 72
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................

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

06/11-20:56:01.416145 y.y.y.141:33050 -> x.x.x.0:7
UDP TTL:238 TOS:0x0 ID:16623 IpLen:20 DgmLen:92
Len: 72
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................

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

These stimuli generated no response.
The victim IP numbers resolved to dialup accounts in Canada and Italy.

1. Source of the trace:

These data were captured from a client's network.

2. Detect was generated by:

This capture was made using Snort 1.7 on a firewall machine in the client's network.

3: Probability the source address was spoofed:

These captured packets are certainly spoofed, it is the spoofed source address on the stimuli packets that determine the victim (although it must be noted that non-spoofed packets would be used to locate smurf-amplifier and fraggle-amplifier networks.) The volume of the packets would indicate that this is an attack and not a scan.

4. Description of attack:

In this attack not only do we see a smurf attack but a fraggle attempt. This smurf-relay-attack contains an added 64-null-byte payload, which differs it from the smurf-relay-attack in Detect #1.
Use of spoofed ICMP Echo requests for denial-of-service is documented in:

CA-1998-01: http://www.cert.org/advisories/CA-1998-01.html
CVE-1999-0513: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0513
IIS X-Force: http://xforce.iss.net/alerts/vol-1_num-5.php#smurf
http://cs.baylor.edu/~donahoo/NIUNet/smurf.html
http://www.pentics.net/denial-of-service/white-papers/smurf.cgi

The fraggle attack is a smurf with a UDP twist. Here the attacker crafts a packet with the victim's IP number as the source of the packet. The destination is the echo service port (UDP port 7,) on the relay-victim's broadcast address. The theory is that servers on the relay network will respond to the stimulus by either an echo response to the spoofed source, or an ICMP Port Unreachable message to the spoofed source. In the case of my client, there was no response to this stimulus (the servers in this network had the Echo service disabled.)
The fraggle attack described in the developer's own words:

http://www.rootshell.com/archive-j457nxiqi3gq59dv/199803/fraggle.c.html

Seeing these two attacks in tandem makes me think that the tool used was papasmurf
http://packetstorm.securify.com/new-exploits/papasmurf.c

5. Attack mechanism:

Via smurf, the attacker leverages the smurf-amplifier's network resources against the victim to essentially flood their connection to the Internet with ICMP traffic, thus denying service. This is accomplished by crafting an ICMP Echo Request that appears to come from the victim that is sent to the broadcast address of a network that allows directed-broadcast packets. This results in every machine on the smurf-amplifier network to send an ICMP Echo Reply to the victim. This geometrically increases the amount of malicious traffic that the attacker can throw at the victim.
Fraggle is the UDP variant of smurf. In this attack, the attacker again leverages a relay-network against the victim, this time it uses the Echo service, via UDP port 7 instead of an ICMP request.

The echo service is often shutoff on servers, and many networks block ICMP at their border, thus the papasmurf is employed (probably fed a long list of vulnerable relay-networks,) to cover both bases.

6. Correlations:

A quick-identification guide on determining a smurf from a fraggle, from a pappasmurf is available at: http://www.securityportal.com/list-archive/fw1/1999/Feb/0160.html

7. Evidence of targeting:

The attacker probably used an online resource like netscan.org or powertech.po/smurf to perform his reconnaissance for him. Since the number of victim IP numbers was low, he was certainly firing-for-effect directly at them. It was certainly not a broad scan of victims.

8. Severity:

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

As a smurf-amplifier, the target is the client's routers and network resources, both of which are critical, so I give it a 5.

The lethality is medium; large networks can handle this load (the client was being used as a relay for sometime before noticing degradation of service, so I rate it as 3.

Arguably, the system countermeasures are low, since they will reply to these pings, so I rate the system countermeasures as 2.

The network countermeasures were *nil*, so I evaluated it as 1.

This yields an Attack Severity level of (5+3)-(2+1) = 5. It is something that should be remedied as soon as possible to help out the victim of the attack.

The fraggle attack is calculated similarly, but in the case of the client, their system and network countermeasures were higher. At the time of the attack, I calculate the severity to be:

(5+3)-(5+1) = 2

In this case it is servere enough that it should motivate changes in the firewall or router to block access to broadcast numbers from the Internet.

9. Defensive recommendation:

A network used as a smurf-amplifier can disable directed-broadcast traffic at their routers. It is good practice to disable this capability on your network as a precaution. In the case of the client, the router was not capable of filtering traffic and handling the load, so a firewall rule was used as a temporary measure until the router was upgraded.

A recipient of a smurf attack has few options but to request their ISP block all ICMP traffic to their network.

Further information involving Cisco equipment is available here:

http://www.cisco.com/warp/public/707/22.html

10. Multiple choice test question:

The papsmurf attack is a combination of what Denial of Service attacks?

A) Trinoo and Smurf
B) Smurf and WinNuke
C) Smurf and Fraggle
D) Ping of Death and Smurf

Answer: C.

Detect #3: UNICODE Exploit Attempt

A client's reverse-proxy is running SNORT in NUIS mode. Their border NIDS system detected a URL request that contained CMD.EXE, yet the SNORT running on the proxy did not log an alert. An audit of the rules on the proxy-server indicated that the SNORT rule-base would miss CMD.EXE requests via HTTP, except in very specific circumstances. A search through their old detects turned up a couple of instances of CMD.EXE requests (listed below.) The rule-base was promptly updated, and SNORT restarted. The successful captures were:

```
[**] WEB-prefix-get // [**]
04/14-15:56:35.343563 200.222.172.88:21097 -> MY.NET.33:80
TCP TTL:107 TOS:0x0 ID:8515  DF
*****PA* Seq: 0xC0AB1DA1   Ack: 0xEDBF0A2F   Win: 0x2398
47 45 54 20 2F 2F 49 49 53 41 44 4D 50 57 44 2F  GET //IISADMPWD/
2E 2E C0 AF 2E 2E C0 AF 2E 2E C0 AF 2E 2E C0 AF  ................
2E 2E C0 AF 2E 2E C0 AF 2E 2E C0 AF 2E 2E C0 AF  ................
2F 77 69 6E 6E 74 2F 73 79 73 74 65 6D 33 32 2F  /winnt/system32/
63 6D 64 2E 65 78 65 3F 2F 63 20 64 69 72 20 48  cmd.exe?/c dir H
54 54 50 2F 31 2E 31 0D 0A 41 63 63 65 70 74 3A  TTP/1.1..Accept:
20 69 6D 61 67 65 2F 67 69 66 2C 20 69 6D 61 67   image/gif, imag
65 2F 78 2D 78 62 69 74 6D 61 70 2C 20 69 6D 61  e/x-xbitmap, ima
67 65 2F 6A 70 65 67 2C 20 69 6D 61 67 65 2F 70  ge/jpeg, image/p
6A 70 65 67 2C 20 61 70 70 6C 69 63 61 74 69 6F  jpeg, applicatio
6E 2F 76 6E 64 2E 6D 73 2D 70 6F 77 65 72 70 6F  n/vnd.ms-powerpo
69 6E 74 2C 20 61 70 70 6C 69 63 61 74 69 6F 6E  int, application
2F 76 6E 64 2E 6D 73 2D 65 78 63 65 6C 2C 20 61  /vnd.ms-excel, a
70 70 6C 69 63 61 74 69 6F 6E 2F 6D 73 77 6F 72  pplication/mswor
64 2C 20 2A 2F 2A 0D 0A 41 63 63 65 70 74 2D 4C  d, */*..Accept-L
61 6E 67 75 61 67 65 3A 20 70 74 2D 62 72 0D 0A  anguage: pt-br..
41 63 63 65 70 74 2D 45 6E 63 6F 64 69 6E 67 3A  Accept-Encoding:
20 67 7A 69 70 2C 20 64 65 66 6C 61 74 65 0D 0A   gzip, deflate..
55 73 65 72 2D 41 67 65 6E 74 3A 20 4D 6F 7A 69  User-Agent: Mozi
6C 6C 61 2F 34 2E 30 20 28 63 6F 6D 70 61 74 69  lla/4.0 (compati
62 6C 65 3B 20 4D 53 49 45 20 35 2E 35 3B 20 57  ble; MSIE 5.5; W
69 6E 64 6F 77 73 20 39 38 3B 20 57 69 6E 20 39  indows 98; Win 9
78 20 34 2E 39 30 29 0D 0A 48 6F 73 74 3A 20 77  x 4.90)..Host: w
77 77 2E XX XX XX XX XX XX XX XX XX XX XX XX XX  ww.XXXXXXXXXXXXX
XX XX XX XX 2E 63 6F 6D 0D 0A 43 6F 6E 6E 65 63  XXXX.com..Connec
74 69 6F 6E 3A 20 4B 65 65 70 2D 41 6C 69 76 65  tion: Keep-Alive
0D 0A 0D 0A                                      ....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
[**] WEB-prefix-get // [**]
04/14-15:56:36.752565 200.222.172.88:21097 -> MY.NET.33:80
TCP TTL:107 TOS:0x0 ID:8529  DF
*****PA* Seq: 0xC0AB1F67   Ack: 0xEDBF0BA4   Win: 0x2223
47 45 54 20 2F 2F 6D 73 61 64 63 2F 2E 2E C0 AF  GET //msadc/....
2E 2E C0 AF 2E 2E C0 AF 2E 2E C0 AF 2E 2E C0 AF  ................
2E 2E C0 AF 2E 2E C0 AF 2E 2E C0 AF 2F 77 69 6E  ............/win
6E 74 2F 73 79 73 74 65 6D 33 32 2F 63 6D 64 2E  nt/system32/cmd.
65 78 65 3F 2F 63 20 64 69 72 20 48 54 54 50 2F  exe?/c dir HTTP/
31 2E 31 0D 0A 41 63 63 65 70 74 3A 20 69 6D 61  1.1..Accept: ima
67 65 2F 67 69 66 2C 20 69 6D 61 67 65 2F 78 2D  ge/gif, image/x-
78 62 69 74 6D 61 70 2C 20 69 6D 61 67 65 2F 6A  xbitmap, image/j
70 65 67 2C 20 69 6D 61 67 65 2F 70 6A 70 65 67  peg, image/pjpeg
2C 20 61 70 70 6C 69 63 61 74 69 6F 6E 2F 76 6E  , application/vn
64 2E 6D 73 2D 70 6F 77 65 72 70 6F 69 6E 74 2C  d.ms-powerpoint,
20 61 70 70 6C 69 63 61 74 69 6F 6E 2F 76 6E 64   application/vnd
2E 6D 73 2D 65 78 63 65 6C 2C 20 61 70 70 6C 69  .ms-excel, appli
63 61 74 69 6F 6E 2F 6D 73 77 6F 72 64 2C 20 2A  cation/msword, *
2F 2A 0D 0A 41 63 63 65 70 74 2D 4C 61 6E 67 75  /*..Accept-Langu
```

```
61 67 65 3A 20 70 74 2D 62 72 0D 0A 41 63 63 65  age: pt-br..Acce
70 74 2D 45 6E 63 6F 64 69 6E 67 3A 20 67 7A 69  pt-Encoding: gzi
70 2C 20 64 65 66 6C 61 74 65 0D 0A 55 73 65 72  p, deflate..User
2D 41 67 65 6E 74 3A 20 4D 6F 7A 69 6C 6C 61 2F  -Agent: Mozilla/
34 2E 30 20 28 63 6F 6D 70 61 74 69 62 6C 65 3B  4.0 (compatible;
20 4D 53 49 45 20 35 2E 35 3B 20 57 69 6E 64 6F   MSIE 5.5; Windo
77 73 20 39 38 3B 20 57 69 6E 20 39 78 20 34 2E  ws 98; Win 9x 4.
39 30 29 0D 0A 48 6F 73 74 3A 20 77 77 77 2E XX  90)..Host: www.X
XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX  XXXXXXXXXXXXXXXX
2E 63 6F 6D 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E  .com..Connection
3A 20 4B 65 65 70 2D 41 6C 69 76 65 0D 0A 0D 0A  : Keep-Alive....

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

A little investigation into the attacking IP yields a Brazillian network:

```
$ whois 200.222.172.88@whois.arin.net
[whois.arin.net]
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

   Record last updated on 11-Apr-2001.
   Database last updated on 16-Jul-2001 23:04:52 EDT.

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.
```

$ whois 200.222.172.88@whois.nic.br
[whois.nic.br]

% Copyright registro.br
% The data below is provided for information purposes
% and to assist persons in obtaining information about or
% related to domain name and IP number registrations
% By submitting a whois query, you agree to use this data
% only for lawful purposes.
% 2001-07-18 00:02:22 (BRT -03:00)

inetnum: 200.222/16
aut-num: AS7738
abuse-c: CGR13
owner: Tele Norte Leste Participagues S.A.
ownerid: 002.558.134/0001-58
responsible: Marcello Lugon

address: Rua Lauro Muller, 116, 21 andar
address: 22299-900 - Rio de Janeiro - RJ
phone: (021) 279-3240
owner-c: MAL516
tech-c: ART3

1. Source of the trace:

These alerts we gathered from a client's network

2. Detect was generated by:

SNORT 1.6 running in NIDS-mode on a client's proxy server generated these alerts.

3: Probability the source address was spoofed:

The stimuli packets were probably not spoofed, the attacker is interested in gathering information about a system vulnerability, and will want to collect the results. The packets could be spoofed, but the attacker would have to have a sensor on the target network (the TTL of 107 would indicate that this possibility is unlikely,) or a sensor on 200.222.172.88's network. The latter is a possibility.

4. Description of attack:

The intended victim platform is Microsoft's IIS. Here the attacker attempts to execute the `dir` command on the web-server via `cmd.exe`. It attempts to traverse up the directory-tree out of the IIS space and into the system space via `../../../../../../../../winnt/system32/`. It hides the slashes by using their UNICODE equivalent of `C0 AF`.

In the case of a vulnerable machine, the attacker would have received the directory of `c:\` on the web-server. In the case of a patched machine, the attacker would receive a 404 or 500 error. Fortunately, for the client, they were already patched. The patch was released in 10/2000, and this attack was launched 4/2001.

5. Attack mechanism:

This attack attempts to exploit the "web server folder directory traversal" vulnerability by encoding a relative reference to a file the attacker wishes to read, or to a program the attacker wishes to execute. Programs executed in this manner will have the privileges of the IUSR_*machinename* account.

By design, IIS will not traverse out of the IIS web-space via relative links. Should the attacker encode the relative-link, via UNICODE, the requested file can be read, or the program executed. Effective encoding of the slash character depends on what languages are supported on the server—more languages opens up more possibilities, and increases risk.

CERT: http://www.kb.cert.org/vuls/id/111677
Microsoft: http://www.microsoft.com/technet/security/bulletin/MS00-078.asp
Bugtraq ID: 1806 http://www.securityfocus.com/bid/1806
ArachNIDS ID: IDS452 http://whitehats.com/info/IDS452
Packetstorm: http://packetstormsecurity.org/0010-exploits/iis-unicode.txt
SANS GIAC: http://www.sans.org/y2k/unicode.htm

6. Correlations:

The Honeynet Project scan-of-the-month number 12 was a UNICODE exploit very similar to this attempt. See http://project.honeynet.org/scans/scan12/ for their top analyses. Their attacker uses:

```
47 45 54 20 2F 6D 73 61 64 63 2F 2E 2E C0 AF 2E  GET /msadc/.....
```

```
2E 2F 2E 2E C0 AF 2E 2E 2F 2E 2E C0 AF 2E 2E 2F  ./....../....../
77 69 6E 6E 74 2F 73 79 73 74 65 6D 33 32 2F 63  winnt/system32/c
6D 64 2E 65 78 65 3F 2F 63 2B 64 69 72 2B 63 3A  md.exe?/c+dir+c:
```

Their detects differ slightly from this detect in that they use some un-encoded /'s in their request and the URL encoding of the `dir` command call differs.

Could this have been the sadmind/IIS worm? The two attempts arrive within a second of one another. http://www.incidents.org/archives/intrusions/msg00526.html cites the sadmind/IIS worm's UNICODE attack to look like:

```
[**] spp_http_decode: IIS Unicode attack detected [**]
05/31-03:07:46.427163 0:D0:58:26:BC:70 -> 0:1:2:39:B0:43 type:0x800 len:0xA5
209.3.45.50:2932 -> a.b.c.1:80 TCP TTL:112 TOS:0x0 ID:53639 IpLen:20
DgmLen:151 DF
***AP*** Seq: 0x7D9264DA  Ack: 0x4FE6C970  Win: 0x4000  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 25 63 30 25 61 66 2E 2E 25  c0%af..%c0%af..%
63 30 25 61 66 2E 2E 25 63 30 25 61 66 2E 2E 25  c0%af..%c0%af..%
63 30 25 61 66 2E 2E 25 63 30 25 61 66 2E 2E 25  c0%af..%c0%af..%
63 30 25 61 66 2E 2E 25 63 30 25 61 66 2F 77 69  c0%af..%c0%af/wi
6E 6E 74 2F 73 79 73 74 65 6D 33 32 2F 63 6D 64  nnt/system32/cmd
2E 65 78 65 3F 2F 63 25 32 30 64 69 72 0D 0A     .exe?/c%20dir..
```

It appears to use a different UNICODE value to encode the slashes and it uses /scripts to disguise itself, not msadc or IISADMPWD. Therefore, this is not likely to be the sadmind/IIS worm at work.

7. Evidence of targeting:

No other evidence of this IP address, or this exploit appeared in the logs on the client's proxy. Other web-sites use this proxy and the target web-site is high-profile. It appears that this was a targeted attempt, and not a scan.

8. Severity:

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

The target in this case, handled e-commerce, so it is a critical system, rating the Target Criticality as 5. Had the system not been patched, this would have resulted in a system compromise, again rating a 5 for the severity of this risk. In the case of the client, they are patched and immune to the attack, earning a 5 for System countermeasures. The firewalls let HTTP traffic right through, and the SNORT rules were insufficient to catch similar attacks, earning only a 1 for Network Countermeasures.

$$(5+5)-(5+1) = 4$$

This is a potentially severe attack, the analysts in the Honeynet Project list ways in how this vulnerability can be exploited to load Trojans on the compromised system.

9. Defensive recommendation:

It is recommended that all IIS servers have the appropriate security patches applied:

Microsoft IIS 4.0:

http://www.microsoft.com/ntserver/nts/downloads/critical/q269862/default.asp

Microsoft IIS 5.0:

http://www.microsoft.com/windows2000/downloads/critical/q269862/default.asp

If your SNORT rules do not already have a general CMD.EXE detection rule add this rule to your rule-base:

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

10. Multiple choice test question:

```
04/14-15:56:36.752565 200.222.172.88:21097 -> MY.NET.33:80
TCP TTL:107 TOS:0x0 ID:8529  DF
*****PA* Seq: 0xC0AB1F67   Ack: 0xEDBF0BA4   Win: 0x2223
47 45 54 20 2F 2F 6D 73 61 64 63 2F 2E 2E C0 AF  GET //msadc/....
2E 2E C0 AF 2E 2E C0 AF 2E 2E C0 AF 2E 2E C0 AF  ................
2E 2E C0 AF 2E 2E C0 AF 2E 2E C0 AF 2F 77 69 6E  ............/win
6E 74 2F 73 79 73 74 65 6D 33 32 2F 63 6D 64 2E  nt/system32/cmd.
65 78 65 3F 2F 63 20 64 69 72 20 48 54 54 50 2F  exe?/c dir HTTP/
31 2E 31 0D 0A 41 63 63 65 70 74 3A 20 69 6D 61  1.1..Accept: ima
67 65 2F 67 69 66 2C 20 69 6D 61 67 65 2F 78 2D  ge/gif, image/x-
78 62 69 74 6D 61 70 2C 20 69 6D 61 67 65 2F 6A  xbitmap, image/j
70 65 67 2C 20 69 6D 61 67 65 2F 70 6A 70 65 67  peg, image/pjpeg
2C 20 61 70 70 6C 69 63 61 74 69 6F 6E 2F 76 6E  , application/vn
64 2E 6D 73 2D 70 6F 77 65 72 70 6F 69 6E 74 2C  d.ms-powerpoint,
20 61 70 70 6C 69 63 61 74 69 6F 6E 2F 76 6E 64   application/vnd
2E 6D 73 2D 65 78 63 65 6C 2C 20 61 70 70 6C 69  .ms-excel, appli
63 61 74 69 6F 6E 2F 6D 73 77 6F 72 64 2C 20 2A  cation/msword, *
2F 2A 0D 0A 41 63 63 65 70 74 2D 4C 61 6E 67 75  /*..Accept-Langu
61 67 65 3A 20 70 74 2D 62 72 0D 0A 41 63 63 65  age: pt-br..Acce
70 74 2D 45 6E 63 6F 64 69 6E 67 3A 20 67 7A 69  pt-Encoding: gzi
70 2C 20 64 65 66 6C 61 74 65 0D 0A 55 73 65 72  p, deflate..User
2D 41 67 65 6E 74 3A 20 4D 6F 7A 69 6C 6C 61 2F  -Agent: Mozilla/
34 2E 30 20 28 63 6F 6D 70 61 74 69 62 6C 65 3B  4.0 (compatible;
20 4D 53 49 45 20 35 2E 35 3B 20 57 69 6E 64 6F   MSIE 5.5; Windo
77 73 20 39 38 3B 20 57 69 6E 20 39 78 20 34 2E  ws 98; Win 9x 4.
39 30 29 0D 0A 48 6F 73 74 3A 20 77 77 77 2E XX  90)..Host: www.X
XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX  XXXXXXXXXXXXXXXX
2E 63 6F 6D 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E  .com..Connection
3A 20 4B 65 65 70 2D 41 6C 69 76 65 0D 0A 0D 0A  : Keep-Alive....
```

This capture is an example of a:

A) MSADC exploit
B) Buffer-Overflow
C) UNICODE exploit
D) Web-Proxy Scan

Answer: C. Although msdac appears in the request, it is not the MSADC exploit. There's not enough encoded hex to put shell-code into, so it's not a buffer overflow.

Detect #4: DNS (udp/53) Scan

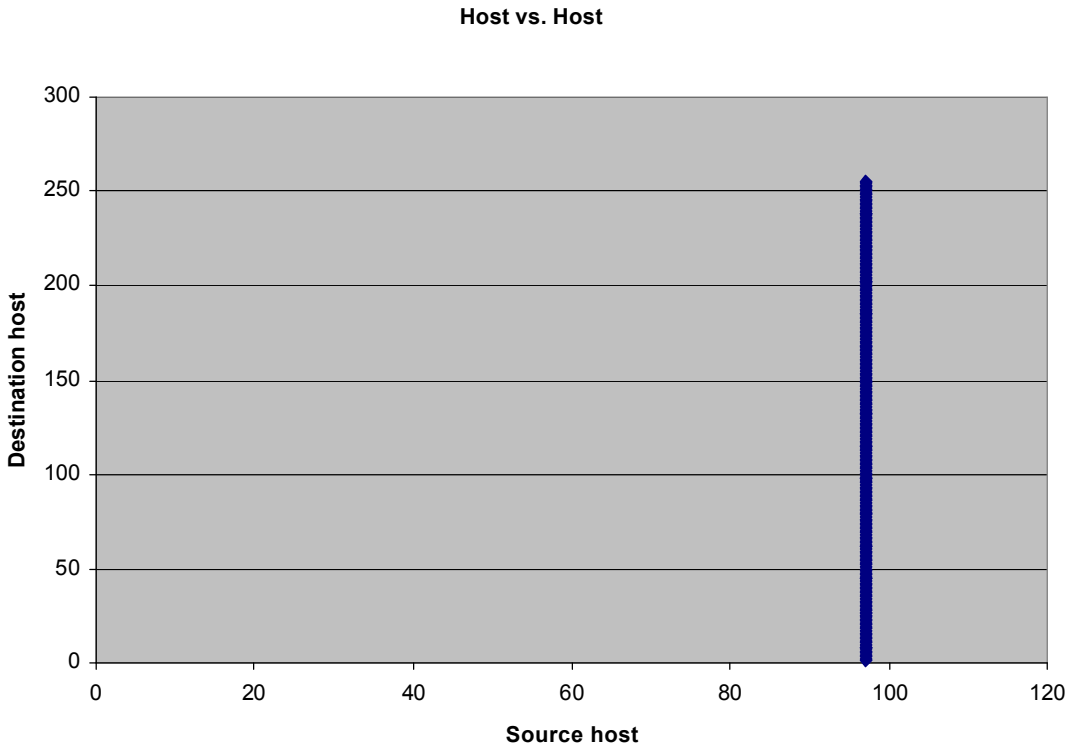
This series of Checkpoint Firewall-1 archive log entries were detected by simple data-reduction scripts currently in-development for a client. An entire day's-worth of log activity was filtered on domain traffic, and graphed. Vertical patterns on inbound traffic can indicate scanning activity. The host, 203.43.135.97 was identified as a potential so filters were applied on this IP which yielded the following log entries:

```
38820;11Jul2001; 5:09:27;FW.MY.NET.220;log;drop;;eth-
s1p1c0;inbound;tcp;203.43.135.97;MY.NET.1;domain-tcp;domain-tcp;40;72;;;;;;;;;
38829;11Jul2001; 5:06:33;FW.MY.NET.221;log;drop;;eth-
s1p1c0;inbound;tcp;203.43.135.97;MY.NET.2;domain-tcp;domain-tcp;40;72;;;;;;;;;
38832;11Jul2001; 5:06:33;FW.MY.NET.221;log;drop;;eth-
s1p1c0;inbound;tcp;203.43.135.97;MY.NET.4;domain-tcp;domain-tcp;40;72;;;;;;;;;
38837;11Jul2001; 5:09:27;FW.MY.NET.220;log;drop;;eth-
s1p1c0;inbound;tcp;203.43.135.97;MY.NET.3;domain-tcp;domain-tcp;40;72;;;;;;;;;
38838;11Jul2001; 5:09:27;FW.MY.NET.220;log;drop;;eth-
s1p1c0;inbound;tcp;203.43.135.97;MY.NET.5;domain-tcp;domain-tcp;40;72;;;;;;;;;
38848;11Jul2001; 5:06:33;FW.MY.NET.221;log;drop;;eth-
s1p1c0;inbound;tcp;203.43.135.97;MY.NET.6;domain-tcp;domain-tcp;40;72;;;;;;;;;
38849;11Jul2001; 5:06:33;FW.MY.NET.221;log;drop;;eth-
s1p1c0;inbound;tcp;203.43.135.97;MY.NET.8;domain-tcp;domain-tcp;40;72;;;;;;;;;
```

… 246 similar log entries later…

```
39545;11Jul2001; 5:06:38;FW.MY.NET.221;log;drop;;eth-
s1p1c0;inbound;tcp;203.43.135.97;MY.NET.254;domain-tcp;domain-tcp;40;72;;;;;;;;;
39546;11Jul2001; 5:09:32;FW.MY.NET.220;log;drop;;eth-
s1p1c0;inbound;tcp;203.43.135.97;MY.NET.255;domain-tcp;domain-tcp;40;72;;;;;;;;;
```

Within a 3-second window, 203.43.135.97 attempts a TCP connection to an entire /24 subnet within the client's organization. A graph of the requests mapping the source IP versus the Destination IP appears below.



Researching the identity of the attacker, 203.43.135.97, turned up an Australian IP. ARIN pointed me to APNIC, who lead me to AUNIC:

```
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)  [No mailbox]
      +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

                        Search the APNIC Whois database
Search results for '203.43.135.97'
inetnum             203.40.0.0 - 203.47.255.255
netname             TELSTRAINTERNET2-AU
descr               Telstra Internet
descr               Locked Bag 5744
descr               Canberra
descr               ACT 2601
```

```
country             AU
admin-c             GH169-AP, inverse
tech-c              GH169-AP, inverse
remarks             ** Conversion note - reference 'GH29-AU' changed to 'GH169-
AP'
remarks             Record imported from AUNIC as part of AUNIC->APNIC
migration
remarks             Please see http://www.apnic.net/db/aunic/
mnt-by              MAINT-AU-GH169-AP, inverse
changed             register@aunic.net 19961120
changed             register@aunic.net 20000105
changed             aunic-transfer@apnic.net 20010525
source              APNIC


person              Geoffrey Huston, inverse
address             Telstra Internet
address             Locked Bag 5744
address             Canberra
address             ACT 2601
phone               +61 6 248 6165
e-mail              gih@telstra.net, inverse
nic-hdl             GH169-AP, inverse
remarks             This data originated from AUNIC, and was copied as part of
remarks             the AUNIC to APNIC migration.
http://www.apnic.net/db/aunic/
remarks             Original nic-hdl in AUNIC: GH29-AU
mnt-by              MAINT-AU-GH169-AP, inverse
changed             register@aunic.net 19951128
changed             aunic-transfer@apnic.net 20010523
source              APNIC
```

Which we confirm from www.aunic.net:

```
inetnum:     203.40.0.0 - 203.47.255.255
netname:     TELSTRAINTERNET2-AU
descr:       Telstra Internet
descr:       Locked Bag 5744
descr:       Canberra
descr:       ACT 2601
country:     AU
admin-c:     GH29-AU
tech-c:      GH29-AU
remarks:     Created 19961120
changed:     nobody@aunic.net 20000105
source:      AUNIC

person:      Geoffrey Huston
address:     Locked Bag 5744
address:     Canberra
address:     ACT 2601
address:     AU
phone:       +61 6 248 6165
e-mail:      gih@telstra.net
nic-hdl:     GH29-AU
remarks:     (Organisation) Telstra Internet
remarks:     Created 19951128
changed:     nobody@aunic.net 19951128
source:      AUNIC
```

The IP resolves back to the DNS server of a Business Products supplier in Austrailia, which is possibly compromised.

1. Source of the trace:

These logs were gathered from a client's border firewall.

2. Detect was generated by:

This detect was made by client's 3rd-party-managed NIDS, the firewall logs were analyzed to collaborate and determine impact. The logs came from Checkpoint Firewall-1 logging to a management server and stored in the archive section—they were not the product of the `fw log` command.

3: Probability the source address was spoofed:

It is unlikely that the source address was spoofed. The intent is to locate DNS servers within the client's network. A spoofed-source scenario is possible. In this scenario the attacker needs a sensor with in the client's network, or in the spoofed-host's network. By selecting a DNS-server to source the DNS scan, the attacker could elude some attention since an analyst my quickly write it off as legitimate traffic.

4. Description of attack:

Here the scanner/attacker sends a SYN (exact flag settings are unknown) connection from port 53 to port 53 on every member of a /24 subnet. Every packet is dropped by rule 72 on the firewall except for MY.NET.251 that is dropped by rule 5. The 40byte length of the packets indicates that the incoming packet has no data in the TCP datagram.

5. Attack mechanism:

This is a fast and furious scan across the client's network. It is sequential, high-speed and definitely automated. The scan attempts a SYN connection to port 53 on the target machines. It is unclear if the attack would attempt to identify the version of the DNS server or launch an exploit. The client had 53/tcp blocked in this instance.

6. Correlations:

Paul Asadoorian made a similar capture in his GCIA practical. His Checkpoint Firewall-1 logs closely match. Source port, destination port, packet length all match. From firewall logs alone, I can only speculate the flag-settings, but it is safe to assume that SYN is one of the flags that were set.

7. Evidence of targeting:

This is likely an untargeted scan at the client. The scanner blindly searches through the address-space, yet does not visit other addresses registered to the client.

8. Severity:

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

DNS servers are critical resources (rating a 5,) the scan does no harm, but can yield information so its Lethality is rated as 2. There are DNS servers in that range, they are current and/or patched, but we don't know what exploit the scanner is going to launch rating system countermeasures 4,) the firewall blocked all of this unsolicited 53/TCP traffic successfully, yielding 5 for network countermeasures.

$$(5+2) – (4+5) = -2$$

This is not a sever attack, but it can indicate possible future activity against the clients' DNS servers. It certainly should be reported to the owners of the source, since they may have a compromised machine.

9. Defensive recommendation:

DNS servers are critical resources and high-profile targets. Be certain that you are running a recent and patched version of DNS (be that BIND, Microsoft or other.) Lock down the configuration to restrict unauthorized zone-transfers, or forwarding of requests. Understand how DNS works and when it uses UDP, versus TCP. Employ statefull, protocol-aware firewall software to protect your DNS servers.

10. Multiple choice test question:

```
38820;11Jul2001; 5:09:27;FW.MY.NET.220;log;drop;;eth-
s1p1c0;inbound;tcp;203.43.135.97;MY.NET.1;domain-tcp;domain-tcp;40;72;;;;;;;;;
38829;11Jul2001; 5:06:33;FW.MY.NET.221;log;drop;;eth-
s1p1c0;inbound;tcp;203.43.135.97;MY.NET.2;domain-tcp;domain-tcp;40;72;;;;;;;;;
38832;11Jul2001; 5:06:33;FW.MY.NET.221;log;drop;;eth-
s1p1c0;inbound;tcp;203.43.135.97;MY.NET.4;domain-tcp;domain-tcp;40;72;;;;;;;;;
38837;11Jul2001; 5:09:27;FW.MY.NET.220;log;drop;;eth-
s1p1c0;inbound;tcp;203.43.135.97;MY.NET.3;domain-tcp;domain-tcp;40;72;;;;;;;;;
…
```

This type of log entry would indicate:

A) 203.43.135.97 is a busy DNS server.
B) UDP scan of MY.NET by 203.43.135.97.
C) Sequential scan of MY.NET for DNS servers from 203.43.135.97
D) Both B and C.

Answer: C

Detect #5: Overnight Scans

```
 Apr 16 18:57:09: %SEC-6-IPACCESSLOGP: list 101 denied tcp 212.236.6.2(3213) ->
DMZ.NET.169.48(111), 1 packet
 Apr 16 18:57:12: %SEC-6-IPACCESSLOGP: list 101 denied tcp 212.236.6.2(3223) ->
DMZ.NET.169.57(111), 1 packet
 Apr 16 19:02:25: %SEC-6-IPACCESSLOGP: list 101 denied tcp 212.236.6.2(3219) ->
DMZ.NET.169.54(111), 1 packet
 Apr 16 22:12:10: %SEC-6-IPACCESSLOGP: list 101 denied tcp 210.71.174.26(53) ->
DMZ.NET.169.48(53), 1 packet
 Apr 16 22:36:24: %SEC-6-IPACCESSLOGP: list 101 denied tcp 63.214.90.206(2797) ->
DMZ.NET.169.48(1080), 1 packet
 Apr 16 22:41:30: %SEC-6-IPACCESSLOGP: list 101 denied tcp 211.54.39.50(3224) ->
INET.T1.CON.26(111), 1 packet
 Apr 16 22:41:32: %SEC-6-IPACCESSLOGP: list 101 denied tcp 63.214.90.206(2812) ->
DMZ.NET.169.63(1080), 1 packet
 Apr 17 00:24:48: %SEC-6-IPACCESSLOGP: list 101 denied tcp 212.33.60.225(4190) ->
DMZ.NET.169.48(53), 1 packet
 Apr 17 00:30:36: %SEC-6-IPACCESSLOGP: list 101 denied tcp 212.33.60.225(4205) ->
DMZ.NET.169.63(53), 2 packets
 Apr 17 01:13:30: %SEC-6-IPACCESSLOGP: list 101 denied tcp 62.122.22.238(23) ->
DMZ.NET.169.48(23), 1 packet
 Apr 17 02:04:29: %SEC-6-IPACCESSLOGP: list 101 denied tcp 212.187.228.216(2687) ->
DMZ.NET.169.51(111), 1 packet
 Apr 17 02:04:32: %SEC-6-IPACCESSLOGP: list 101 denied tcp 212.187.228.216(2689) ->
DMZ.NET.169.53(111), 1 packet
 Apr 17 02:09:39: %SEC-6-IPACCESSLOGP: list 101 denied tcp 212.187.228.216(2698) ->
DMZ.NET.169.62(111), 1 packet
 Apr 17 03:08:19: %SEC-6-IPACCESSLOGP: list 101 denied tcp 128.121.2.143(3317) ->
DMZ.NET.169.48(19216), 1 packet
 Apr 17 03:13:41: %SEC-6-IPACCESSLOGP: list 101 denied tcp 128.121.2.143(3317) ->
DMZ.NET.169.48(19216), 1 packet
 Apr 17 03:14:26: %SEC-6-IPACCESSLOGP: list 101 denied tcp 217.58.40.250(1552) ->
INET.T1.CON.26(111), 1 packet
 Apr 17 04:02:22: %SEC-6-IPACCESSLOGP: list 101 denied tcp 216.119.50.81(1572) ->
DMZ.NET.169.48(1080), 1 packet
 Apr 17 04:02:24: %SEC-6-IPACCESSLOGP: list 101 denied tcp 216.119.50.81(1704) ->
DMZ.NET.169.63(1080), 1 packet
 Apr 17 04:02:27: %SEC-6-IPACCESSLOGP: list 101 denied tcp 216.119.50.81(1891) ->
DMZ.NET.169.48(1080), 1 packet
 Apr 17 04:02:30: %SEC-6-IPACCESSLOGP: list 101 denied tcp 216.119.50.81(2055) ->
DMZ.NET.169.63(1080), 1 packet
 Apr 17 04:02:52: %SEC-6-IPACCESSLOGP: list 101 denied tcp 216.119.50.81(2841) ->
DMZ.NET.169.48(1080), 1 packet
```

Who are the players in this little drama?

Host	Hostname	Registered To	Activity
212.236.6.2		cybertron.at	111/tcp scan
210.71.174.26		Asiacast.net	53/tcp blocked
63.214.90.206	dialup-63.214.90.206.Dial1.Boston1.Level3.net.	Level3.net	1080/tcp scan
211.54.39.50		Korea, kornet.net	111/tcp scan of ISP
212.33.60.225	cm60-225.liwest.at	Liwest.at	53/tcp scan
62.122.22.238	62-122-22-238.flat.galactica.it.	Galactica.it	23/tcp blocked
212.187.228.216		Level3.com	111/tcp scan
128.121.2.143		Verio.net	19216/tcp blocked

217.58.40.250		Interbusiness.it	111/tcp scan of ISP
216.119.50.81		Jps.net	1080/tcp scan

This is not a detect of a single event, but an example of how to triage and evaluate the amount of exposure from reconnaissance probes. We don't have logs from what penetrated the firewall, nor do we have logs of any replies from DMZ.NET machines.

A quick glance over the incident table once can eliminate 62.122.22.238 as a threat. There is only one attempt to port 23 on one machine in the DMZ. Odds are good that this is a wrong number. Another class of scans that can probably be ignored are the scans from 217.58.40.250 and 211.54.39.50 to port 111. These machines appear to be checking for port 111 on DMZ.NET's connection to their ISP. 217.58.40.250 and 211.54.39.50 are probably scanning INET.T1.CON.0/24, not DMZ.NET specifically.

The connection from Verio's 128.121.2.143 to port 19216/tcp looks innocent enough, it hits only one machine, and one port, and the two attempts appear to be a retry attempt—all nice an innocent-looking until you do a google.com search on 19216/tcp and turn up a number of reports about NT servers being compromised and Serv-U ftp server is setup to distribute warez. This one deserves some later attention.

Moving on, we look at the exposure from the 111/tcp scans at DMZ.NET. At the time of this detect, incidents.org had ranked port 111 as one of the top 5 most-scanned-for ports. It's still up in that range, and caused mostly by the sadmind/IIS worm.

Source IP	Source Port	Destination IP	Delta Source Port	Delta Dest IP
212.236.6.2	3213	DMZ.NET.169.48	N/a	N/a
212.236.6.2	3219	DMZ.NET.169.54	6	6
212.236.6.2	3223	DMZ.NET.169.57	4	3
212.187.228.216	2687	DMZ.NET.169.51	N/a	N/a
212.187.228.216	2698	DMZ.NET.169.53	2	2
212.187.228.216	1552	DMZ.NET.169.62	9	9

There is a very clear covariance of source port and destination IP. The network events also indicate a sequential scan. We can infer from the logs that .49, .50, .52, .55,.56, .58, .59, .60, and .61 are not protected by the Access Control Lists. DMZ hosts less than .48 and greater than .62 are also potentially at risk (assuming that the logs reported are incomplete.)

Using the same covariance calculation we see that the 53/tcp scan from 212.33.60.225 potentially penetrated to DMZ.NET.169.49 through .62.

The attempt from asiacast.net's 210.71.174.26 is probably benign, at the very least it was unsuccessful and they appeared to move on to another network without further probes (at least from the reported log evidence.)

The next port to consider is 1080/tcp. The scanners are probably looking for wingates or socks servers to use as proxies for their surfing, scanning, or attacking. Using the covariance technique, level3.net's 63.214.90.206 appears to be performing a sequential scan of DMZ.NET.169. DMZ.NET.169.49 through .62 are potentially vulnerable, or at least they are unprotected by the Cisco Access Lists. Jps.net's 216.119.50.81, on the other hand, does not show a covariant pattern. It appears to be either oscillating through the list two to three times, or there are two to three scanning processes running in parallel on 212.187.228.216. One would need more observations, or more granularity (such as sequence number and ID numbers,) make a better guess. A curious coincidence between the two scans is that only .48 and .63 are protected. Either they are the only hosts in DMZ.NET.169 getting scanned (implying that both scanners are sharing the same list of targets,) or that the Access Control Lists are protecting only those servers.

1. Source of the trace:

From the April 25th, 2001 GIAC detect: http://www.sans.org/y2k/042501.htm

2. Detect was generated by:

These logs come from a Cisco router employing access control lists. This detect was reported by Micheal Dwyer to handler@incidents.org.

3: Probability the source address was spoofed:

It appears that all of these packets were scan stimuli. Thus the odds are low that they are spoofed, since the intent is to capture the result of the scan. It is important to note that there exists the scenario where the scanner has a sensor in the scan-source's network, and spoofs the stimuli and captures the result.

4. Description of attack:

Here we see a number of horizontal scans, and a probable wrong number to port 23. We have scans looking for 53/tcp (DNS,) 111/tcp (portmapper,) and 1080/tcp. The DNS scans could be precursors to exploit attempts, or later vulnerability scans. The portmapper scans a more than likely sadmind/IIS worms. The 1080/tcp scans are probably searching for open proxies.

5. Attack mechanism:

Nearly all of these scans were sequential, horizontal, SYN scans. They operate by sequentially attempting a TCP connection to their target port. Some software will perform half-scans, and not complete the TCP handshake. These logs contain only those that were blocked, so it is unclear if these were full or half-scans.

6. Correlations:

The connection between tcp/19216 and warez-sites was made from:
http://www.splash.co.za/print.php?sid=4

DNS vulnerabilities can be found at:
CERT CA-2000-03: http://www.cert.org/advisories/CA-2000-03.html

Information about the sadmind/IIS worm:
CERT CA-2001-11: http://www.cert.org/advisories/CA-2001-11.html
Bugtraq ID 866: http://www.securityfocus.com/vdb/bottom.html?vid=866
CVE-1999-0977: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0977

Information about Wingate and proxy scanning:
CERT VN-1998-03: http://www.cert.org/vul_notes/VN-98.03.WinGate.html

7. Evidence of targeting:

Each of these scans appears to be random trolling. The scanners are in the early stages of looking for targets.

8. Severity:

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

On average these scans are looking for serves, but not critical servers, rating a 3. The reconnaissance is potentially successful, but not crippling, yielding a 2 for lethality. There is not enough information about the services running, or the patch level of the services, so we have to guess and give them a median 3 for system countermeasures. The Access Control Lists were capable of blocking some, but not

all of the probe attempts (from inference,) yielding a 3.

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

This attack deserves some attention, but from the log information the alert status of DMZ.NET.169 does not need to be raised.

9. Defensive recommendation:

It is recommended that internal scanning and audits of the probed ports should be executed against DMZ.NET.169. The risk posture of the network needs to be established. As a precaution, patches to any Solaris machines need to be applied (to protect against the sadmind/IIS worm,) and any NT web servers need to be patched against UNICODE exploits (for the same reasons.) All DNS servers should be upgraded and patched, and they should be configured to not respond to requests for their software version. All socks servers should be audited to verify that they are not open proxies.

10. Multiple choice test question:

```
Apr 16 18:57:09: %SEC-6-IPACCESSLOGP: list 101 denied tcp 212.236.6.2(3213) ->
DMZ.NET.169.48(111), 1 packet
Apr 16 18:57:12: %SEC-6-IPACCESSLOGP: list 101 denied tcp 212.236.6.2(3222) ->
DMZ.NET.169.57(111), 1 packet
```

In the above log entries, the source port and destination IP both differ by 9. This is an example of what kind of relation:

A) Equality relation
B) Proximiy relation
C) Covariance relation
D) Both B and C.

Answer: D. The events occur with in the same minute and the source port and destination IP vary by the same amount.

Assignment 2

Traffic Analysis, Data Mining, and Anomaly Detection in Network Intrusion Detection

Introduction

The ideal Network Intrusion Detection System will efficiently and effectively classify network traffic between benign and belligerent. A great deal of research and work in Network Intrusion Detection involves the development of attack signatures. Protected network traffic is fed into a matching-engine and alerts are generated when the network traffic matches a given attack signature. An intrusion analyst uses these alerts to determine how to allocate analysis resources.

"Traffic analysis is the branch of signal intelligence analysis that deals with the study of external characteristics of signal communications. The information was used: (1) to effect interception, (2) to aid cryptanalysis, (3) to rate the level and value of intelligence in the absence of the specific message contents, and (4) to improve the security in the communication nets." (Nichols, p71.) The performance of a Network Intrusion Detection System can be more effective if it includes not only signature matching but also traffic analysis. By using traffic analysis, anomalous traffic is identified as a potential intrusion, no signatures are involved in the process, so it more likely to detect new attacks for which signatures have yet to be developed.

Traffic analysis deals not with the payload of a message, but its other characteristics such as source, destination, routing, length of the message, time it was sent, the frequency of the communication, etc. Traffic payload is not always available for analysis, the traffic may be encrypted, VPN links may be flowing through your monitoring area, or it may simply be against policy to analyze packet payload. For the purposes of Network Intrusion Detection we gather these characteristics either from the actual network traffic itself (via a method such as `tcpdump`,) or by log-files from network sensors such as firewalls or routers (Lee and Stolfo.) These data are processed for visualization or by data mining techniques to generate alerts and provide useful information to the analyst to aid them in the decision on how to allocate analysis resources.

The Aim of Network Intrusion Detection

The intent of a Network Intrusion Detection system is to guide the analyst towards network-events that are malicious. The two major approaches are misuse detection and anomaly detection. Pattern-matching solutions primarily use misuse detection. They employ a library of signatures of misuse, which are used to match against network traffic. The weaknesses of these systems are: variants, false positives, false negatives, and data overload. Since they rely on signatures, a new variant of an attack can be created to evade detection. Additionally, the signatures themselves can create false positives if they are not written correctly, or if the nature of the attack is difficult to isolate from normal traffic characteristics. A signature-based system cannot detect attacks for which is has no signature—they don't react well to the unknown. Data overload can occur when a sensor, or an analyst is presented with too much information to analyze effectively (Phung.)

Traffic analysis performed on network traffic can mitigate the limitations of signature-based systems, because they are anomaly detectors. It is important to note that they cannot replace signature-bases systems; ideally a analyst would have both tools at his disposal. A system based on traffic analysis can detect attack variants because it is not looking for the pattern of the attack, but triggering on the anomalous nature of the connection (either from a strange IP, or to a strange port, or of an odd packet length or flag-setting.) False positives are also a weakness of anomaly detection, but if the alerts from both methods can be correlated first, the relevance of the alert will improve. The strength of anomaly detection is its low rate of false negatives. New attacks, for which signatures have not been developed for the signature-based system to trigger on, will be anomalous by nature. An anomaly-based detection system might not catch the latest IIS UNICODE exploit, but the behavioral change of the compromised system will get its attention. Reducing data overload is accomplished by data-mining and visualization techniques. Abstracting the data and

presenting it visually to the analyst can detect anomalies and patterns that the heuristics of the traffic analysis system could not.

A Multi-dimensional Model

A given packet can be broken down into a number of fields such as protocol, source IP, destination IP, ports used and flag settings (in the case of TCP or UDP,) or message type (in the case of ICMP,) and length. A field is either numerical, or can be converted to a numerical range via a mapping function (e.g. mapping IP numbers to integer ID numbers.) If the information captured by a packet has *n* fields, then a packet can be expressed as a vector of *n* elements (i.e. an *n*-tuple) in an *n*-dimensional vector space. This gives a unique spatial-representation of each event and creates a context for the network activity (Girardin, p. 4.) Within this *n*-dimension vector space events can be correlated, compared, and visualized. In order to satisfy the need to communicate clearly to the analyst, one can employ visualization and data mining to produce useful graphs and alerts.

The Quarry

A Network Intrusion Detection System provides the analyst with not only information about what has happened, but potentially what is *about* to happen. If an analyst is able to detect an attacker's reconnaissance of the protected network, and the alert arrives in time, an attack can potentially be thwarted. An attacker will probe your defenses, and this will leave traces that can be detected. At this point the analyst is much like a tracker, attempting to infer the intent behind the signs left (Carss, P 22.) Scans have footprints, which can be defined as the set of port and IP combinations scan is targeting (Stainford, Hoagland, and McAlerney pgs 2-4.) They characterize a horizontal scan as one that searches a group of IP numbers for a single port, and a vertical scan as a single IP being scanned for multiple ports. Other footprint geometries can be described, such as box scanning—a combination of vertical and horizontal. When plotting destination port versus destination IP numbers these patterns become prominent. The size of the footprint can be calculated from the sum of the IP/port combinations used in the scan. This size can serve as a metric on how difficult it will be to detect a scan. Clearly an NMAP scan on a server will be easier to detect than a scan for port 53 on 4 machines in the network.

The response to a scan's stimuli also forms a detectable footprint. SYN-ACK, responses from open TCP ports, ICMP port unreachable messages from UDP inverse-scans, drops or rejects logged by a firewall, can be used to detect a scan and evaluate how much information the attacker gained from the scan. These are all responses to the scan, much like disturbed pebbles or broken foliage tell the passing of your quarry.

The attacker will often employ some sort of deception to disguise their reconnaissance scan. Some methods of deception can successfully elude simple port-scan detectors. An attacker can alter the scanning software to randomize the hosts scanned, or slow the scan down to cover a larger time-window, or randomize the period between probes. They can blur the signature of a single scan packet by randomizing non-essential fields such as source port of the scan, the sequence or ack number, or IP id. These techniques can evade simple port-scan detection software based on signatures, sequential scan detection, or exceeding *x*-events over a *y*-second threshold. Furthermore, the true source scan can be disguised by hiding with-in forged scans, or employing distributed scanning. A scan hiding within a smoke-screen of other scans does little to disguise that scanning is going on—in fact it draws a lot of attention to the scanning-event, but it can protect the identity of the source. Distributed scanning, on the other hand, can be difficult to detect if a system is simply looking at events correlating source IP, destination IP, and destination port (Stainford, Hoagland, and McAlerney, pgs. 4-5.)

One special method employed by some scanners to avoid detection is the *stealth-scan*. This is a bit of a misnomer, since from the point of view of most Intrusion Detection systems these scans, in the words of SNORT's author Marty Roesch, "are more like sore-thumb scans." These scans operate by using illegal flag-settings that can evade some simple packet filters. Perhaps *penetration-scan* is better label for the technique.

Anomaly Detection Tools

Traffic analysis is performed through visualization and data mining techniques. Recall that network events can be represented as vectors in an *n*-dimensional vector space. The dimension of this vector space can be reduced in intelligent ways in an attempt to highlight detectable patterns. A simple method of dimension reduction would reduce the vector space down to destination IP, and destination port, and this reduced space would be visualized as a scatter-plot. From this plot an analyst can visually detect horizontal, and vertical scan footprints. Another simple dimension reduction method would reduce the space down to the source and destination IP of the events. This plot would illustrate which machines are communicating with each other. As the data space is reduced, information is lost, so it is important that a number of reduction and visualization methods are employed, in order to give a more complete picture to the analyst. An analyst could use more sophisticated mapping techniques, such as neural-network-computed self-organizing maps (Girardin,) or spicules (Vert, Frincke, and McConnell,) in an attempt to gather a higher-resolution picture of the status of the network. These visualization techniques can work from captured network traffic, or network equipment logs. In addition to visualization, the results of data mining can be compared to heuristics to detect patterns or anomalies. Mark Prager describes a technique of reducing firewall logs to detect scan and DoS scans (Pragger.) Additional tools for generating alerts from log files are available from http://www.enteract.com/~lspitz/intrusion.html.

SPICE/SPADE

Silicon Defense's SPICE (Stealthy Portscan and Intrusion Correlation Engine) project is a DARPA-sponsored development-effort whose aim is to build a better mousetrap capable of detecting stealthy port scans. SPICE consists of two components, an anomaly sensor and a correlation engine. SPADE is the anomaly detector, which acts as a plug-in preprocessor to SNORT. The correlation engine is still under development.

Each packet coming into the anomaly detector is assigned a anomaly score A(x). This score is calculated from the negative log of the probability of the event, P(x). I.e., A(x) = - log(P(x)) (Staniford, Hoagland, and McAllerney, P 4.) A sharp eye would note that their anomaly score is close, if not equivalent to the calculation of a signal's Entropy (Shannon, p 13.)

The calculation of P(x) is based on observed network traffic, since network traffic, as a signal, is non-ergodic, and thus not subject to universal computation of probability distributions for all possible signals (Pierce p57-59.) SPADE uses four methods of calculating P(x): P(destination IP, destination port), P(source IP, destination IP, destination port), P(source IP, source port, destination IP, destination port), and a Bayes network approximation of P(source IP, source port, destination IP, destination port.) From observation, a packet to port 80 on a web-server will be more probable than say, port 37337 to the same server. The higher P(x) is, the lower A(x) will be. If A(x) exceeds the provided threshold, SPADE will generate an alert.

In their published paper on SPADE, they measure the performance of the system using efficiency (ratio of true positives to all positives,) and effectiveness (ratio of true positives to all trues.) From their results, it appeared that the most effective and efficient method was P(destination ip, destination port), or joint-2 method, which is now the default probability-setting for SPADE. It was found that filtering the calculation to include only the protected network further improved efficiency and effectiveness (Staniford, Hoagland, and McAllerney, P 13.) The anomaly threshold could be lowered when filtering was used since the probabilities were matched against local IP/port pairs, which is a smaller space to consider than the rest of the Internet.

The settings for SPADE need tuning to match the protected network. One tunable factor is the alert-threshold. This can be set manually and tuned by the analyst, or SPADE itself can be set to set its own threshold level. In default learning mode, SPADE will monitor network traffic for 24 hours. Then it will calculate the threshold level required to create 200 alerts in that monitoring period. The length of the monitoring period and the number of alerts to generate are selectable.

As it runs, SPADE will generate a probability table of observed network traffic. This table contains critical information and should be protected and backed-up. If this file is lost, SPADE will need to be

retrained, exposing your network as the history is rebuilt and alerts are not generated. When operating in survey-mode, SPADE will generate reports on observed probability distributions of the network traffic. SPADE can be placed into statistical mode if one wishes to view the probability tables on a regular basis.

Currently, SPADE simply generates alerts on packets whose anomaly score (as calculated by SPADE,) exceeds the anomaly threshold level. These alerts are logged along with the other SNORT alerts. It is the correlation engine, which is still under development, which promises to detect the stealthiest of port scans.

The correlation engine is fed alerts from the anomaly detector. The alerts contain the event, and the anomaly score. The correlation engine will keep an event in memory based on its anomaly score. The higher the score, the more anomalous the event, thus the longer it will keep its state. The correlation engine then attempts to link the events into groups to possibly link rare events to a single cause. Links between a given pair of events are calculated by a series of heuristic functions. There are four basic heuristic functions, and the connection between two events is scored as a combination of the heuristic functions' results. If the source IP or the destination port or network are the same in the two events, a given heuristic would fire. A second heuristic would look for events close to each other in time, or in the *n*-dimensional space (a Euclidean difference.) A third would link two events that were off by one IP number, or one source port number, or one destination IP. Another heuristic would detect covariance relations (such as increases of one in destination IP and destination port.) The connection function would be a combination of these heuristic outputs. The correlation engine would build a graph linking related events, and alert the analyst of these correlations (Staniford, Hoagland, and McAllerney, P 8.)

Future Steps

Once SPICE itself is released, Silicon Defense intends to apply the tool to detect and track worms and DDoS attacks, in addition to stealthy port scans (Staniford, Hoagland, and McAllerney, P 15.) In the field of traffic analysis and data mining there is plenty of room for work in Intrusion Detection Fusion, where the logs and alerts from different NIDS systems, firewalls, and routers are synthesized into one data-space for analysis (Girardin, P 12.)

Conclusion

There is still much work to be done in the field of anomaly-based detection. Misuse detection based on signature matching has limitations; these limitations can be mitigated through the use of anomaly-based detection. The fusion of both misuse-detection and anomaly-detection techniques will result in a more effective and efficient Network Intrusion Detection System.

References

Carss, Bob. The SAS Guide to Tracking. New York: Lyons Press, 2000.

Girardin, Luc. "An eye on network intruder-administrator shootouts." URL: http://www.usenix.org/event/detection99/full_papers/girardin/girardin_html/index.html (June 6, 2001)

Lee, Wenke and Stolfo, Salvatore J. "Data Mining Approaches for Intrusion Detection." URL: http://www.cs.columbia.edu/~wenke/papers/usenix/usenix.html (July 9, 2001)

Nichols, Randall K. ICSA Guide to Cryptography. New York: McGraw-Hill, 1999. 71-75.

Phung, Manh. "Data Mining in Intrusion Detection." Intrusion Detection FAQ. October 24, 2000. URL: http://www.sans.org/newlook/resources/IDFAQ/data_mining.htm (July 9, 2001)

Pierce, John R. An Introduction to Information Theory: Symbols, Signals and Noise, Second Revised Edition. New York: Dover Publications, Inc., 1980.

Prager, Mark. "Firewall Log-Checking Techniques." Sys Admin. August 2001: 33-37.

Shannon, Claude E. "A Mathematical Theory of Computation" October 1948. URL: http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf (July 19, 2001)

Staniford, Stuart, Hoagland, James A., and McAlerney, Joseph M. "Practical Automated Detection of Stealthy Portscans." URL: http://www.silicondefense.com/pptntext/spice-ccs2000.pdf (July 19, 2001)

Vert, Greg, Frincke, Deborah A., and McConnell, Jesse C. "A Visual Mathematical Model for Intrusion Detection." URL: http://citeseer.nj.nec.com/vert98visual.html (July 19, 2001)

Assignment 3

Inventory

The data to be analyzed consisted of fast alert files, port scan logs , and out-of-spec logs from March 1st, through July 8th. Additionally, there were eleven summaries of port scans, and two correlated port scan log files.

Overview Attack Summary

An overview of detected attacks, sources and destinations was taken from the log files from June 3rd through July 8th. Broken down by week, and an overall summary appear below.

Top ten attacks June 3-June 9

UDP SRC and DST outside network	587048
Possible trojan server activity	19570
External RPC call	3924
High port 65535 udp - possible Red Worm - traffic	2145
Watchlist 000220 IL-ISDNNET-990517	2014
SMB Name Wildcard	938
connect to 515 from outside	660
Queso fingerprint	650
WinGate 1080 Attempt	441

Top ten attacks June 10-June 16

UDP SRC and DST outside network	528754
SYN-FIN scan!	14349
Watchlist 000220 IL-ISDNNET-990517	4645
Port 55850 tcp - Possible myserver activity - ref. 010313-1	4344
External RPC call	4272
SMB Name Wildcard	1693
Queso fingerprint	1578
Possible trojan server activity	1214
WinGate 1080 Attempt	987

Top ten attacks June 17-June 23

UDP SRC and DST outside network	313367
Watchlist 000220 IL-ISDNNET-990517	127896
Possible trojan server activity	10482
SYN-FIN scan!	4221
External RPC call	3513
SMB Name Wildcard	2131
connect to 515 from outside	1962
Port 55850 tcp - Possible myserver activity - ref. 010313-1	1849
Watchlist 000222 NET-NCFC	423

Top ten attacks June 24-June 30

UDP SRC and DST outside network	543878
Possible trojan server activity	53728
Watchlist 000220 IL-ISDNNET-990517	18933
WinGate 1080 Attempt	8751
High port 65535 tcp - possible Red Worm - traffic	5194
Tiny Fragments - Possible Hostile Activity	4460
External RPC call	4203
connect to 515 from outside	2990
SMB Name Wildcard	680

Top ten attacks July 1-July 8

UDP SRC and DST outside network	37118
Possible trojan server activity	15956
SYN-FIN scan!	8526
Watchlist 000220 IL-ISDNNET-990517	6497
External RPC call	2524
connect to 515 from outside	2209
SMB Name Wildcard	1232
Queso fingerprint	945
WinGate 1080 Attempt	673

Top Attacks detected June 3-July 8:

UDP SRC and DST outside network	2010165
Watchlist 000220 IL-ISDNNET-990517	159985
Possible trojan server activity	100950
SYN-FIN scan!	27253
External RPC call	18436
WinGate 1080 Attempt	11249
connect to 515 from inside	8105
Port 55850 tcp - Possible myserver activity - ref. 010313-1	6890
SMB Name Wildcard	6674
High port 65535 tcp - possible Red Worm - traffic	5714

We will analyze the impact, risks, and defensive recommendations based on the total alert counts from June 3rd to July 8th. There is additional analysis of worm activity and Out-of-Spec packets over the whole data range of March 1st through July 8th.

UDP SRC and DST outside network

This alert triggers when UDP traffic is detected where both the source and destination IP do not belong in the protected network. Most of these alerts were generated by Multicast traffic, RFC 1918 traffic, and Microsoft Windows workstations.

Multicast IP numbers lie within 224.0.0.0 through 239.255.255.255.

RFC 1918 addresses (ftp://ftp.isi.edu/in-notes/rfc1918.txt) include: 10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16.

A Microsoft Windows machine will allocate itself a 169.254.0.0/16 number if it does not receive a reply to its DHCP request. This behavior accounts for the UDP traffic to 169.254.255.255.

Once the multicast, RFC 1918, and windows DHCP-orphans were filtered out, only 3253 alerts remain, 3177 of which were UDP port 137 requests from various ipt.aol.com addresses to 130.132.143.43 (acs-wins.its.yaled.edu) and 130.132.143.42 (pantheon-po01.its.yale.edu.) Perhaps caused by visiting faculty.

A similar pattern occurs with umaryland.edu, and Maryland Department of Transportation.

Crafted packet spoofing can cause alerts of this nature. Egress filtering should be put in place to make sure that RFC1918 traffic remains local only, and that spoofed packets cannot leave the network. In order to limit the amount of false positives, include RFC1918 and multicast addresses into the protected address list on the SNORT servers.

Watchlist 000220 IL-ISDNNET-990517

This alert will trigger with any traffic coming from 212.179.0.0/16. No alerts contain traffic from MY.NET to 212.179.0.0/26, either the traffic is being blocked, or the sensor's rules are looking only for the source address to match 212.179.0.0/26. The rule does not identify if the traffic is UDP or TCP, so the analysis of the alerts is mostly conjecture. The port-to-port footprint of the traffic from 212.179.0.0/26 with possible uses of the ports is listed below. It indicates that traffic from 217.179.0.0/26 is not being blocked.

Top Ten Source-Port/Destination-Port Pairs:

ISDNNET	MY.NET	# alerts	Possible nature of traffic
3697	1234	94489	Connection from ISDNNET to port 1234 in MY.NET to exploit Ultors trojan, or searching for Ultors trojan
3620	1234	32635	Connection from ISDNNET to port 1234 in MY.NET to exploit Ultors trojan, or searching for Ultors trojan
38550	4241	14369	vrml
62903	4236	2965	vrml
1200	4020	2893	SCOL or NoBackO trojan
31611	41003	1268	unknown ports
23206	1214	1038	Kazaa
1049	4734	698	unknown ports
23088	1372	419	fujitsu config protocol
10070	1214	365	Kazaa

Assuming worst-case intentions from ISDNNET (they are, after all on a watch-list for a reason,) port 1234 is known to be to shell port opened up by the Ultors Trojan. All traffic involving destination port 1234 is between 212.179.58.200 (no reverse map,) and MY.NET.150.220. In the June 3rd to July 8th window, two connections were made, the first at 06/18 0356, the second started at 06/18 1114. The sessions ran concurrently. MY.NET.150.220 should definitely be examined for evidence of a service running at 1234.

The source-destination table of the vrml ports indicates only that a service is running on these MY.NET services:

ISDNNET		MY.NET	Port	# packets
212.179.38.71	clnt-38071.bezeqint.net	MY.NET.97.13	4236	10
212.179.56.5	Unresolved	MY.NET.97.176	4236	252
212.179.56.5	Unresolved	MY.NET.97.44	4236	2987
212.179.79.2	pc.creoscitex.co.il	MY.NET.218.198	4241	14369

NoBackO, a reported anti-BackOrifice program will open ports on 1200 and 1201. There is traffic between 212.179.47.70 (fr-c47070.bezeqint.net) and MY.NET.97.175. There are not other sessions between MY.NET.97.175 and 212.179.0.0/16.

Kazaa is a peer-to-peer file-sharing program similar to Napster or Guntella. It can increase a network's risk since it offers a new vector for the introduction of malicious software, and it can consume a lot of bandwidth. Based on just the Watchlist captures, kazaa servers are running on:

MY.NET.104.111
MY.NET.130.135
MY.NET.150.133
MY.NET.150.220
MY.NET.150.225
MY.NET.153.163
MY.NET.217.154
MY.NET.217.18
MY.NET.218.126
MY.NET.218.154
MY.NET.218.178
MY.NET.218.234
MY.NET.219.50
MY.NET.70.77
MY.NET.70.97
MY.NET.75.145

It is not clear from the ports 23088 and 1372, which is the server and which is the client. 212.179.41.235 (fr-c41218.bezequint.net) is communicating with MY.NET.97.165. Research indicates that port 1372 is used by Fujitsu for a configuration protocol (Neohapsis Port List,) this could be the case (if so, there should be some concern, or 1372 could simply be an ephemeral port picked by MY.NET.97.165 to connect to 212.179.41.235 port 23088. It is recommended that MY.NET.97.165 be checked for a service running on port 1372.

The connection between 212.179.72.226 (unresolved) port 31611 and MY.NET.97.210 port 41003 is still an unknown. As a precaution, it is recommended that MY.NET.97.210 be checked over.

The connection between 212.179.41.216 (fr-c41216.bezequint.net) port 1049 and MY.NET.156.55 port 4734 is still an unknown. If pressed for a guess, one could reason that fr-c41216 appears to be a dialup, so it's quite likely that it is a windows machine, and starting up new connections above 1024. So it is likely that it is connecting to a service on MY.NET.156.55, on port 4734. Thus MY.NET.156.55 should be examined.

There is also telnet and smtp traffic. There either telnet attempts or sessions (the rules don't capture return-traffic,) between 212.179.80.74 (PT712074.bezeqint.net,) and MY.NET.60.11, and MY.NET.60.39. This machine also appears to answer an ident request from MY.NET.60.11.

A summary inbound SMTP traffic:

ISDNNET	MY.NET	# packets
212.179.45.241	MY.NET.253.18	5
212.179.46.193	MY.NET.253.41	9
212.179.46.193	MY.NET.6.47	7
212.179.54.34	MY.NET.145.9	14
212.179.72.53	MY.NET.253.114	5
212.179.72.53	MY.NET.253.41	9
212.179.72.53	MY.NET.253.42	25
212.179.72.53	MY.NET.253.43	11

A similar analysis of ISDNNET activity can be correlated with Paul Asadoorian's (337) analysis at: http://www.sans.org/y2k/practical/Paul_Asadoorian_GIAC.doc, and P.J. Goodwin's (305) at http://www.sans.org/y2k/practical/PJ_Goodwin_GCIA.doc.

Possible trojan server activity

This rule triggers on traffic to port 27374, the port commonly used by the SubSeven tojan. There were 160791 alerts of this activity between March 1st and July 8th. MY.NET.70.38 is the top internal source

of this traffic. The following internal servers have generated notable alert traffic (in order of traffic volume:)

MY.NET.15.214
MY.NET.146.95
MY.NET.202.26
MY.NET.228.50
MY.NET.202.34
MY.NET.208.142
MY.NET.97.200
MY.NET.98.123
MY.NET.60.16
MY.NET.98.110
MY.NET.105.120
MY.NET.98.228

These are top-talkers, many more internal hosts have generated traffic, but less than 100 packets in the full sampling period. A network scan for 27374 is recommended to ascertain the true infection rate.

Good virus-scanning policy is the best defense against the SubSeven and Trojans like it.

SYN-FIN scan!

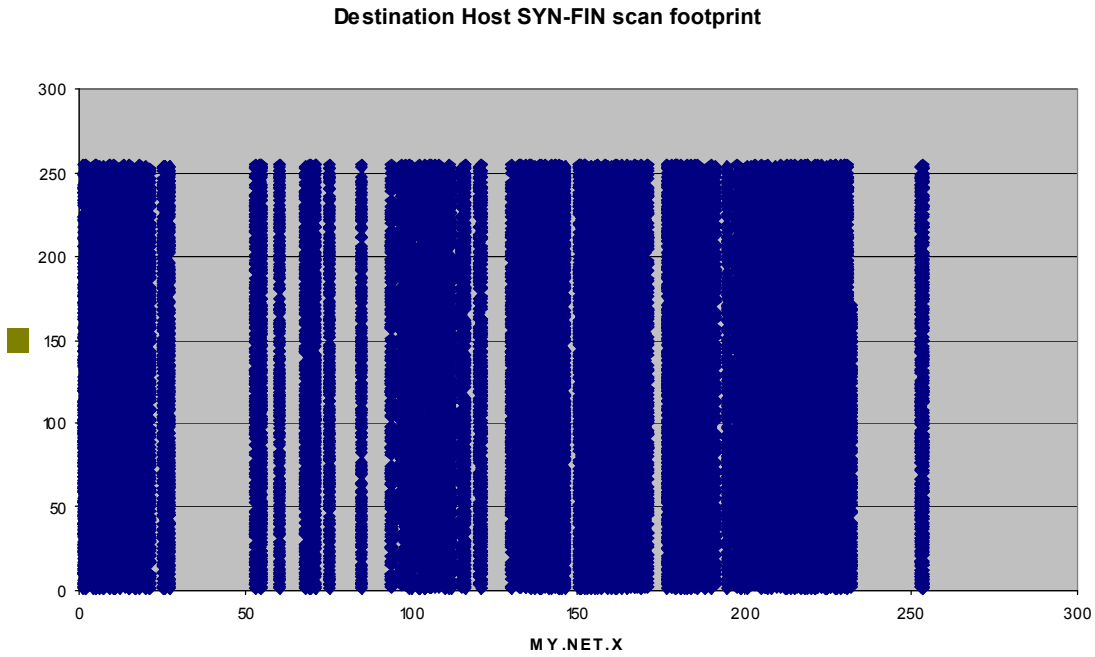
This rule generates an alert when a packet enters the network with both the SYN and FIN flag set, this flag combination is illegal and will not appear in normal traffic. Packets are crafted with this flag setting, and used to penetrate certain packet-filtering firewalls or cisco access lists. This is a reconnaissance-technique and could indicate a future attack.

Over the March 1st through July 8th window, there were 49,981 SYN-FIN event recorded, sourcing from 50 IP numbers, scanning 27,202 destinations.

The top ten external hosts scanning the MY.NET network:

211.240.28.66
213.255.24.48
211.178.63.4
61.13.106.35
210.160.190.244
132.248.100.200
210.96.75.129
61.11.252.117
206.139.131.244
64.0.153.38

Between these machines, they've scanned a sizable percentage of MY.NET. Their top target was port 21 (ftp,) by far at 26971 events in the June 3rd to July 8th window. Port 111 (portmapper) had 269 events in the same timeframe.



As a defense against, SYN-FIN packets should be blocked at the border routers. This is cut down the amount of information a potential attacker can gain from "stealth" scanning.

Paul Asadoorian had a similar analysis at http://www.sans.org/y2k/practical/Paul_Asadoorian_GIAC.doc .

External RPC call

When an external host attempts to connect to port 111 on the protected net, this alert will fire. Port 111 is the portmapper service which will inform the requester what ports RPC services are running on. Portmapper is used to identify which ports are open to attack on a host. This is likely reconnaissance activity, and could indicate a future attack. The snort rules do not capture the response of the probed host, so the true risk posture of the network is difficult to identify from these logs.

The top ten external hosts scanning port 111:

host	hostname	# of scans
202.98.10.70	unknown	1304
61.143.127.86	unknown	1243
134.198.26.42,1229	gem300.chem.uofs.edu	1229
211.152.241.1	unknown	1176
129.49.65.82	pi.msrc.sunysb.edu	800
212.209.79.162	unknown	759
24.147.14.159	h0000949291ad.ne.mediaone.net	734
128.95.12.195	rogue.bchem.washington.edu	651
129.186.213.89	smith.ansci.iastate.edu	614
211.23.6.234	unknown	432

Inbound port 111 traffic could be blocked at the border routers which would curb a fair amount of RPC exploits from entering the network. The network policy of the University might require that this be left open. In that case, the next best defense is to make sure that port 111 is protected by tcpwrappers to limit access to authorized hosts. Also, make sure that all RPC services offered are patched and updated.

Paul Asadoorian had a similar analysis at http://www.sans.org/y2k/practical/Paul_Asadoorian_GIAC.doc.

WinGate 1080 Attempt

WinGate is a web-proxy program with a history of being wide open for external use. It resides on port 1080. When external traffic is destined to the protected network on port 1080, this rule is activated. The top internal servers being visited are:

MY.NET.157.5
MY.NET.100.203
MY.NET.60.16
MY.NET.60.39
MY.NET.60.38
MY.NET.60.8
MY.NET.217.142
MY.NET.218.162
MY.NET.97.223
MY.NET.217.58

The number of alerts for the internal hosts range from 50 to 97. It is unlikely that these machines are being exploited as external web proxy/relays. Most of the traffic is in a scan format, single packets to multiple machines. A quick scan of port 1080 on these machines is recommended, just to be sure.

The top offenders look like likely victims themselves, especially wingate.proxy.monitor.dal.net, it's possible that they are exploiting proxies to scan for more proxies.

host	hostname	# of alerts
208.151.245.252	dsl-208-151-245-252.easystreet.com	693
24.200.15.30	modemcable030.15-200-24.que.mc.videotron.ca	584
24.249.236.109	cc233016-g.owml1.md.home.com	527
24.130.201.49	we-24-130-201-4.we.mediaone.net	504
62.54.255.94	dsdn-3e36ff53.pool.mediaWays.net	476
208.11.228.86	pm2port16.newbethlehem.pcidu.com	331
24.141.61.183	d141-61-183.home.cgocable.net	289
142.169.139.36	ts1-26.f1733.quebectel.com	272
217.10.143.54	wingate.proxy.monitor.dal.net	268
63.193.150.61	adsl-63-193-150-61.dsl.lsan03.pacbell.net	248

WinGate itself can be configured with access lists to control its use. It is recommended that any web proxies be locked down to MY.NET at the very least.

Further information about the WinGate 1080 detect is available from Paul Asadoorian's detects at http://www.sans.org/y2k/practical/Paul_Asadoorian_GIAC.doc, and P.J. Goodwin's at http://www.sans.org/y2k/practical/PJ_Goodwin_GCIA.doc.

connect to 515 from inside

The printer daemon runs on port 515, and is a common attack destination. This alert indicates that an internal machine is connecting to a remote printer daemon. This could be legitimate traffic, or it could be the symptom of a worm such as the Ramen.

There is not a lot of traffic of this nature in the June 3rd to July 8th window. The top link is from MY.NET.100.234 to 192.35.232.241 (pixpat.austin.ibm.com.) This server should be checked for infection, but it's possible that a research project is underway with IBM, and the remote print traffic is legitimate.

There is another link between MY.NET.98.139 and 198.100.97.100(smaug.cd.hope.edu.)

There is another rule to catch inbound 515 traffic. It was not a top 10 alert in the complete June 3rd to July 8th monitoring window, but it was a top 10 in a couple of the one-week windows. 1268 internal hosts were targets of 515 traffic from the outside. The top targets:

MY.NET.137.53
MY.NET.137.115
MY.NET.137.45
MY.NET.137.97
MY.NET.137.87
MY.NET.137.40
MY.NET.137.83
MY.NET.137.65
MY.NET.137.59
MY.NET.137.170

The deviation of alerts to host is quite low, alert counts range from 15 to 19 in this sample. This indicates mostly scan activity.

The top scanners are:

host	hostname	# of alerts
64.27.27.1	unknown	1442
165.132.31.137	unknown	1207
150.183.110.179	unknown	774
202.109.72.113	unknown	622
216.139.196.151	unknown	450
210.103.58.65	unknown	419
161.184.162.126	edtn008762.hs.telusplanet.net	308
202.64.229.178	unknown	268
137.78.79.35	wkimpc.jpl.nasa.gov	253
24.251.243.247	Ci662472-a.billtwn1.ky.home.com	250

As a precaution, 515 may be blocked at the border routers. This will disable remote printing off-campus which may or may not agree with the networks acceptable use, or security policy.

Additional information about 515 traffic is available from Paul Asadoorian's detects at http://www.sans.org/y2k/practical/Paul_Asadoorian_GIAC.doc, and P.J. Goodwin's at http://www.sans.org/y2k/practical/PJ_Goodwin_GCIA.doc.

Port 55850 tcp - Possible myserver activity - ref. 010313-1

The process called myserver running on a linux machine listens on port 55850. This alert will indicate either a scan for exploited systems, or actual exploitation activity. There is also the risk of many false positives since 55850 is a valid source port. Breaking down the source port/destination port pairs from the alerts in the June 3rd to July 8th monitoring window the top alerts are from Networks News Transport protocol, SMTP, SSL, PO3, and ident. The false positives also seem to indicate that there is possibly a web

proxy running on port 8000 on MY.NET.179.80.

One interesting link is between 64.213.55.2 port 55850 and MY.NET.130.122 port 2072. The host IP does not resolve. It's possible that it is using a service on port 2072 of MY.NET.130.122, it is also possible that MT.NET.130.122 is exploiting a myserver daemon running on 64.213.55.2.

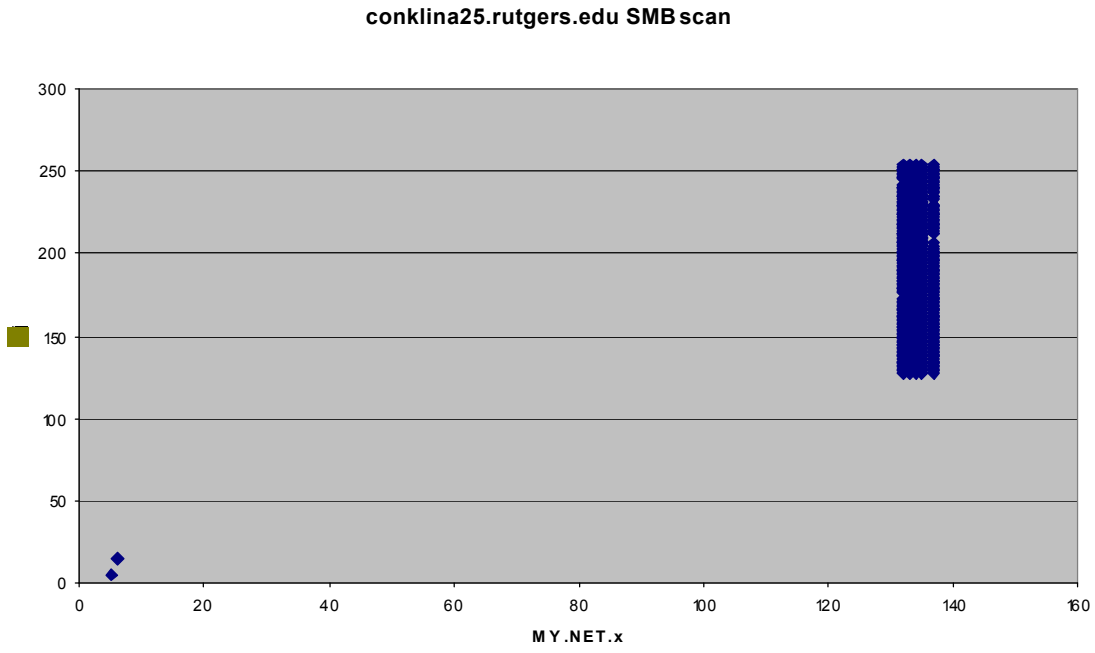
Myserver is part of common rootkits (toolkits used by hackers once they gain access to a system to hide their presence, cover their tracks, and enable them to launch attacks from the compromised machine,) but not an exploit itself. Since port 55850 is a legitimate source port, it is not recommended that it be filtered at the network border.

SMB Name Wildcard

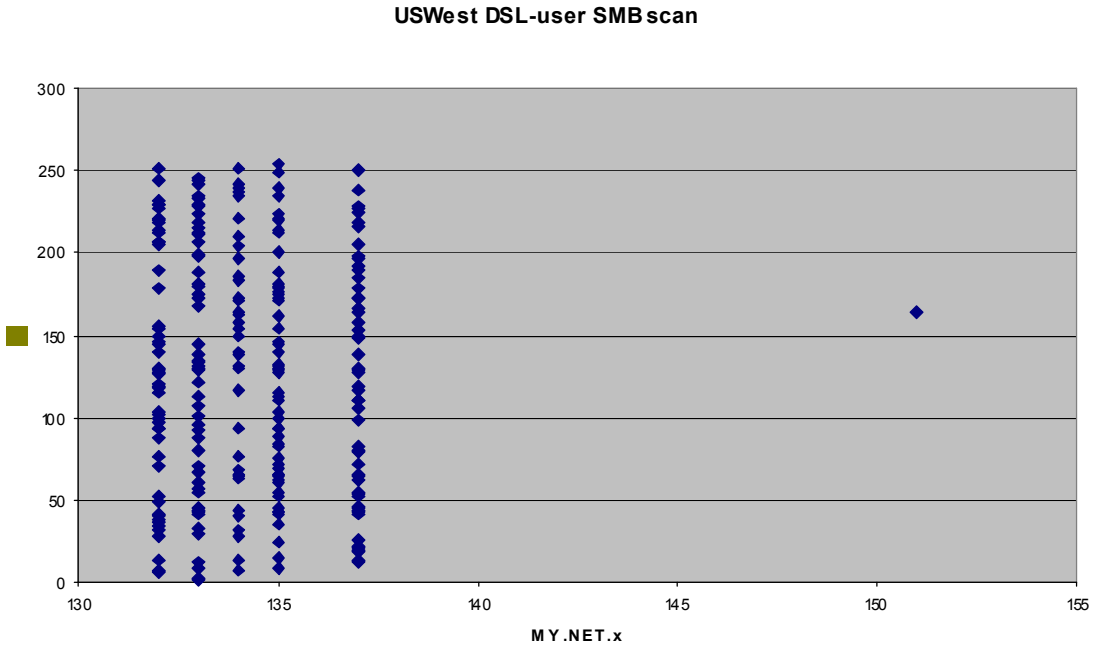
This rule triggers on some sort of netbios name query (possibly IDS177.) It is a reconnaissance technique to gain information about the windows system.

host	hostname	# of scans
165.230.53.35	conklina25.rutgers.edu	1492
216.63.216.27	Unknown	257
216.61.41.249	adsl-216.61.41.249.dsl.austtx.swbell.net	166
216.67.164.34	Unknown	100
MY.NET.162.199		41
130.13.91.62	vdsl-130-13-91-62.phnx.uswest.net	41
130.13.85.245	vdsl-130-13-85-245.phnx.uswest.net	31
130.13.64.30	vdsl-130-13-64-30.phnx.uswest.net	29
130.13.147.94	vdsl-130-13-147-94.phnx.uswest.net	29
130.13.79.197	vdsl-130-13-79-197.phnx.uswest.net	28

Conklina25.rutgers.edu is clearly scanning the network with the following footprint:



It also appears that a Phoenix DSL user is accumulating some intelligence as well:



NetBIOS traffic in general (port 136, 137, and 139,) should be blocked at the border router. It should be restricted to on-campus use, just like RFC1918 network traffic.

High port 65535 tcp - possible Red Worm – traffic

The Red or Adore worm, will open up a shell on port 65535. This rule will trigger on any traffic with a source or destination port of tcp/65535. It runs the risk of many false positives since as a source port, it can be valid traffic. After cleaning out SMTP, FTP, ident, POP3, Kazaa, and SSL traffic we yield the following link-table from June 3rd to July 8th:

Source	Destination	# of Alerts
64.12.168.249	MY.NET.111.139	67
62.242.237.21	MY.NET.201.6	8
MY.NET.182.120	194.64.244.10	6
MY.NET.201.6	62.242.237.21	5
194.64.244.10	MY.NET.182.120	5
MY.NET.218.166	195.55.140.187	3
62.243.123.40	MY.NET.213.218	3
MY.NET.97.195	24.189.99.55	2
24.189.99.55	MY.NET.97.195	2
24.186.247.135	MY.NET.218.70	2
194.230.145.137	MY.NET.153.199	2
65.92.94.198	MY.NET.97.216	1
65.14.165.238	MY.NET.217.166	1
213.45.53.154	MY.NET.217.46	1

The top link is 64.12.168.249:65535 (ftpnscp.newaol.com) to MY.NET.111.139:3805. There are

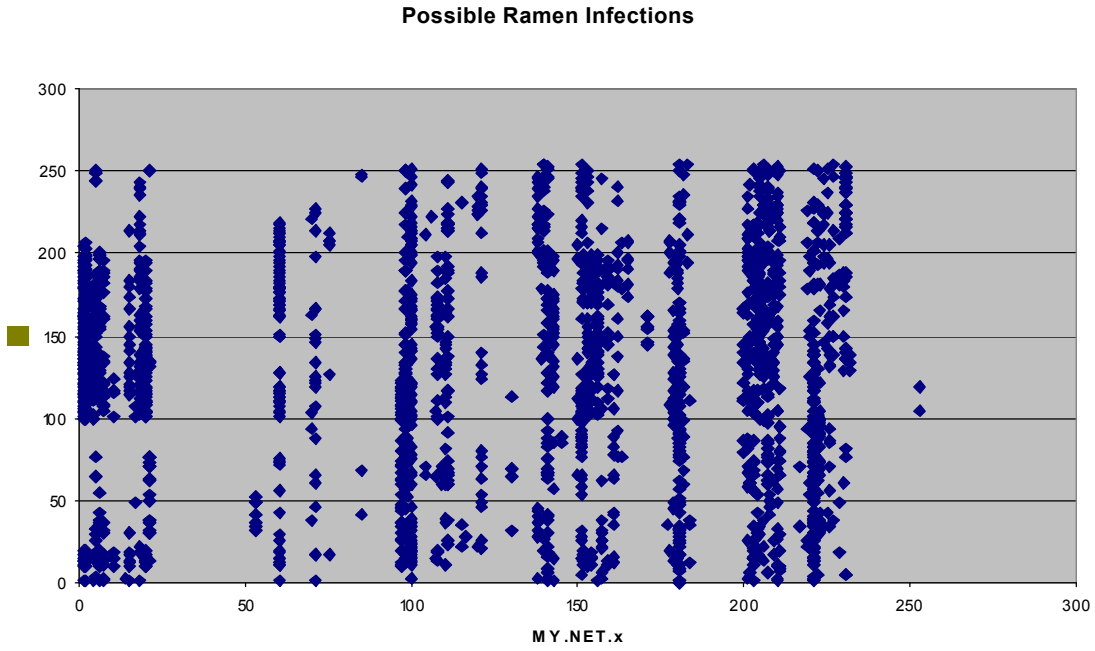
no logs of traffic from MY.NET.111.139 to ftpnscp.newaol.com. It's possible that this is merely a passive ftp session between the two machines.

The link between 62.242.237.21:65535 (cpe.atm2-0-109178.boanxx8.customer.tele.dk) and MY.NET.201.6:6346 is two way. From the logs, it appears that 62.242.237.21 initiated the connection to port 6346 on MY.NET.201.6 on 06/09 0905. This does not appear to be Red Worm traffic, but it is certainly suspicious and would warrant a scan of MY.NET.201.6 at the very least. There was stimulus from a foreign country and there was response.

Further analysis of SMB Name Wildcard alerts can be gleaned from Paul Asadoorian's detects at http://www.sans.org/y2k/practical/Paul_Asadoorian_GIAC.doc, and P.J. Goodwin's at http://www.sans.org/y2k/practical/PJ_Goodwin_GCIA.doc.

Further Worm Analysis: Ramen

The Ramen worm alert will trigger on access to or from port 27374, which is where the Ramen worm will open up a shell for exploitation. This rule can also generate a lot of false positives. Ramen alerts were gathered from the March 1st to July 8th timeframe, well-known port traffic (SMTP, SSL, Napster, etc.) was filtered out and, the source, and destination alerts were correlated to determine 2074 possibly-infected hosts on the protected network:



Since 27374 can be a valid source port for service requests, it is no feasible to block this port at the border routers. A better defense is to block its infection vectors: rcp.statd (block 111/portmapper,) wuftpd (make sure ftp servers are up to date,) and LPRng, (port tcp/515.)

Out-of-Spec Analysis

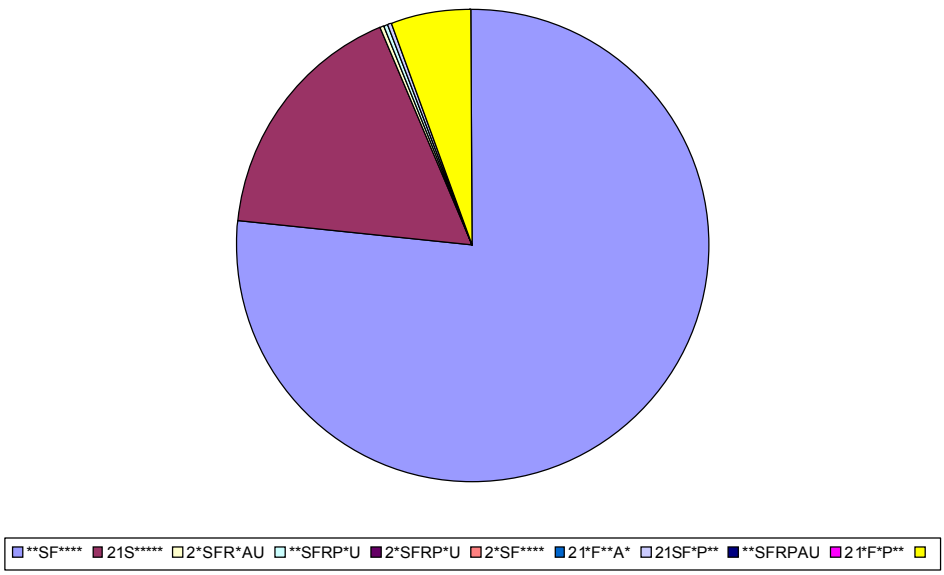
Out-of-Spec logs from March 1st through July 8th were analyzed. The most common OOS packet that enter the network is one with **SF**** flag settings. This is from SYN-FIN scanning activity. The next most common OOS packet has flags set to 21S*****, which could be a false positive, now that the 2 and 1 flags are being used for Quality of Service by newer TCP/IP Stacks. The logged OOS traffic is mostly reconnaissance in nature. The top SYN-FIN scanners are:

211.178.63.4
211.240.28.66
61.13.106.35 (c35.h061013106.is.net.tw)
210.160.190.244 (linux.nangoku.co.jp)
61.11.252.117

Most of the OOS traffic is between the 213.116.0.0 network and MY.NET.100.165. The top 5 internal targets are:

MY.NET.100.165
MY.NET.253.114
MY.NET.253.41
MY.NET.253.43
MY.NET.253.42

Out-of-Spec Packets



%	# of alerts	Flag settings
0.7655	81886	**SF****
0.1702	1820121	S*****
0.0025	2652	*SFR*AU
0.0022	232	**SFRP*U
0.0009	1002	*SFRP*U
0.0007	752	*SF****
0.0007	7521	*F**A*
0.0007	7521	SF*P**
0.0007	74	**SFRPAU
0.0007	7321	*F*P**
0.0552	5908	OTHER

Additionally, the OOS packets were fed through a passive OS fingerprinting routine. Only 438

systems were able to be identified, 338 of which identified themselves as JetDirect cards. The JetDirect classification is made when packets less that 64 TTL, Window size between 5804 and 5840, the DontFragment flag is not set, and Type of service is set to 0. Since most OOS packets are crafted, the reliability of this OS fingerprinting technique is impaired.

Host	Inferred OS
12.78.16.253	Windows 9x/NT [Intel]
142.154.131.118	Windows 2000 [Intel]
152.19.254.81	Solaris 2.x [Intel/Sparc]
152.2.111.5	Windows 9x/NT [Intel]
152.7.17.97	Windows 9x/NT [Intel]
152.7.25.14	Windows 9x/NT [Intel]
152.7.31.253	Windows 9x/NT [Intel]
193.253.187.82	Windows 9x/NT [Intel]
195.204.46.106	Windows 9x/NT [Intel]
200.204.140.180	Windows 9x/NT [Intel]
200.204.176.224	Windows 9x/NT [Intel]
202.104.128.94	Windows 2000 [Intel]
202.174.230.60	Netware 4.11 [Intel]
202.188.229.196	Windows 2000 [Intel]
202.94.64.50	Windows 9x/NT [Intel]
203.106.117.14	Windows 2000 [Intel]
203.173.179.44	Windows 9x/NT [Intel]
203.173.185.113	Windows 9x/NT [Intel]
205.164.224.38	Windows 2000 [Intel]
205.164.227.67	Windows 9x/NT [Intel]
205.164.229.158	Windows 9x/NT [Intel]
206.48.60.142	Windows 2000 [Intel]
208.33.170.121	Windows 9x/NT [Intel]
209.14.216.137	Windows 2000 [Intel]
209.239.199.49	Windows 9x/NT [Intel]
209.252.195.45	Windows 9x/NT [Intel]
209.53.10.100	Windows 2000 [Intel]
209.88.70.151	Windows 9x/NT [Intel]
210.20.9.211	Windows 9x/NT [Intel]
212.123.168.33	Windows 9x/NT [Intel]
212.123.168.71	Windows 9x/NT [Intel]
212.139.182.57	Windows 9x/NT [Intel]
212.139.184.123	Windows 2000 [Intel]
212.139.185.31	Windows 9x/NT [Intel]
212.139.32.41	Netware 4.11 [Intel]
212.139.33.237	Windows 9x/NT [Intel]
212.242.103.35	Windows 2000 [Intel]
213.241.39.250	Windows 9x/NT [Intel]
213.40.9.147	Windows 9x/NT [Intel]
213.48.208.204	Windows 9x/NT [Intel]
213.76.54.119	Windows 2000 [Intel]
213.76.96.55	Windows 2000 [Intel]
216.187.158.187	Windows 9x/NT [Intel]
216.203.235.7	Netware 4.11 [Intel]

216.232.191.87	Netware 4.11 [Intel]
216.232.193.192	Windows 9x/NT [Intel]
216.232.193.196	Windows 9x/NT [Intel]
217.134.68.209	Windows 2000 [Intel]
217.157.68.193	Windows 2000 [Intel]
217.157.84.2	Netware 4.11 [Intel]
217.99.192.129	Windows 2000 [Intel]
24.160.46.26	Windows 9x/NT [Intel]
24.165.130.149	Windows 2000 [Intel]
24.165.71.33	Netware 4.11 [Intel]
24.167.136.210	Windows 2000 [Intel]
4.33.108.187	Windows 9x/NT [Intel]
4.35.220.75	Windows 2000 [Intel]
4.40.188.52	Netware 4.11 [Intel]
4.43.169.68	Windows 9x/NT [Intel]
62.106.24.212	Netware 4.11 [Intel]
62.149.138.224	Windows 9x/NT [Intel]
62.180.194.248	Windows 9x/NT [Intel]
62.180.195.72	Windows 2000 [Intel]
62.180.198.19	Windows 9x/NT [Intel]
62.180.201.29	Windows 9x/NT [Intel]
62.180.203.77	Windows 2000 [Intel]
62.180.204.167	Netware 4.11 [Intel]
62.180.211.103	Windows 9x/NT [Intel]
62.180.215.153	Windows 2000 [Intel]
62.180.218.166	Windows 9x/NT [Intel]
62.180.218.251	Windows 2000 [Intel]
62.29.77.193	Windows 9x/NT [Intel]
62.31.25.197	Windows 9x/NT [Intel]
62.59.136.163	Windows 9x/NT [Intel]
62.59.140.71	Windows 2000 [Intel]
62.59.145.197	Windows 2000 [Intel]
62.59.148.178	Windows 9x/NT [Intel]
62.59.153.158	Windows 9x/NT [Intel]
62.59.4.88	Netware 4.11 [Intel]
62.59.9.78	Windows 2000 [Intel]
63.196.116.175	Netware 4.11 [Intel]
63.197.154.2	Windows 2000 [Intel]
63.253.114.9	Windows 9x/NT [Intel]
63.28.143.235	Windows 9x/NT [Intel]
64.161.19.182	Windows 2000 [Intel]
64.167.150.209	Netware 4.11 [Intel]
64.180.99.102	Windows 9x/NT [Intel]
64.198.133.235	Windows 9x/NT [Intel]
64.48.221.60	Windows 9x/NT [Intel]
64.48.221.66	Windows 9x/NT [Intel]
66.27.128.66	Netware 4.11 [Intel]
66.50.115.21	Windows 9x/NT [Intel]
66.50.120.151	Windows 9x/NT [Intel]
66.50.28.80	Windows 9x/NT [Intel]

66.50.59.232	Windows 9x/NT [Intel]
66.50.77.214	Windows 2000 [Intel]
66.50.84.164	Windows 2000 [Intel]

Scans of Note

Since scans precede attacks, scan traffic from July 1st through July 8th was analyzed to provide a heads-up warning of what is to come.

The top scanning machines:

Host	Hostname	# of Scans
MY.NET.160.114		157996
211.207.15.190		30156
128.143.75.164	bootp-75-164.bootp.Virginia.EDU	28122
66.68.62.229	cs666862-229.austin.rr.com	23501
208.188.3.10	ns1.dallasisd.org	21494
MY.NET.75.196		18072
211.23.9.162		16962
212.17.127.36	TK212017127036.teleweb.at	15636
205.188.233.121	g2lb4.spinner.com	15059
205.188.233.153	g2lb5.spinner.com	14921

The MY.NET.160.114 "scan" appears to be a service running on port 777. All alerts involving MY.NET.160.114 involve this source port except for 1 UDP connection from the internal server out to port 1084. According IANA, port 777 is a multilingual HTTP server.

211.207.15.190 is performing a SYN scan of port 21 on MY.NET.0.0/16. This was a high-speed scan.

128.143.75.164 is performing the same SYN scan of port 21 on MY.NET. This too was a high-speed scan.

66.68.62.229 appears to be retaliating to a port scan from MY.NET.219.42. MY.NET.219.42 performed a random-order SYN scan on 66.68.62.229 on July 1st 12:39. 66.68.62.229 performed a high-speed sequential port scan on MY.NET.219.42 at 21:20 on July 1st. This could also be a faculty/staff/student interacting with their home machine toying around with NMAP.

From the ns1.dallasisd.org name, I expected 208.188.3.10 to be a false positive, but from the logs it is apparent that ns1.dallasisd.org is performing a fast, sequential port-scan of MY.NET.217.142. This could be a precursor to an attack on that machine. It is recommended that an audit be performed on MY.NET.217.142 to make sure that it is secure.

MY.NET.75.196 is performing a SYN scan of 170.140.0.0/16 for port 21. This machine is exhibiting malicious behavior and should be examined.

211.23.9.162 is performing a high-speed SYN scan of port 53 on MY.NET.0.0/16. This is certainly malicious reconnaissance.

212.17.127.36 is performing a high-speed SYN scan of port 21 on MY.NET.0.0/16 on July 4th.

The scans from 205.188.233.0/24 network are looking for UDP ports 6970 and 6972. These ports lie within the range used by RealAudio. The destination IP footprint does not look like a scan, but actual server usage:

Destination	# of alerts
MY.NET.110.33	3383
MY.NET.71.248	3153
MY.NET.180.76	3143
MY.NET.178.222	2874
MY.NET.108.13	2711

MY.NET.145.166	2565
MY.NET.145.197	2377
MY.NET.107.4	2310
MY.NET.15.223	2057
MY.NET.106.184	1828
MY.NET.106.178	1813
MY.NET.70.92	1595
MY.NET.109.62	1558
MY.NET.146.17	1048
MY.NET.111.30	1018
MY.NET.178.219	730
MY.NET.110.169	690
MY.NET.108.16	611
MY.NET.178.154	518
MY.NET.75.103	393
MY.NET.15.22	351
MY.NET.158.25	305
MY.NET.104.216	66
MY.NET.121.23	10

Traffic is both voluminous and strange enough to warrant a quick look a the servers to determine if this is RealAudio traffic sourcing from the protected network.

Rogues Gallery

Host: 212.179.58.200
Reason for selection: Appears to be infected with or exploiting Ultors Trojan
Hostname: Unresolved
Registrant:

ARIN points to RIPE

```
inetnum:      212.179.58.0 - 212.179.58.255
netname:      NV-PICTUREVISION
descr:        network
country:      IL
admin-c:      NP469-RIPE
tech-c:       NP469-RIPE
status:       ASSIGNED PA
notify:       hostmaster@isdn.net.il
mnt-by:       RIPE-NCC-NONE-MNT
changed:      hostmaster@isdn.net.il 20000229
source:       RIPE

route:        212.179.0.0/17
descr:        ISDN Net Ltd.
origin:       AS8551
notify:       hostmaster@isdn.net.il
mnt-by:       AS8551-MNT
changed:      hostmaster@isdn.net.il 19990610
source:       RIPE

person:       Nati Pinko
address:      Bezeq International
address:      40 Hashacham St.
address:      Petach Tikvah  Israel
phone:        +972 3 9257761
e-mail:       hostmaster@isdn.net.il
nic-hdl:      NP469-RIPE
```

```
changed:      registrar@ns.il 19990902
source:       RIPE
```

Host: 211.240.28.66
Reason for selection: Top SIN-FIN scanner
Hostname: Unresolved
Registrant:

ARIN points to APNIC

```
inetnum:      211.232.0.0 - 211.255.255.255
netname:      KRNIC-KR
descr:        KRNIC
descr:        Korea Network Information Center
country:      KR
admin-c:      HM127-AP
tech-c:       HM127-AP
remarks:      ****************************************
remarks:      KRNIC is the National Internet Registry
remarks:      in Korea under APNIC. If you would like to
remarks:      find assignment information in detail
remarks:      please refer to the KRNIC Whois DB
remarks:      http://whois.nic.or.kr/english/index.html
remarks:      ****************************************
mnt-by:       APNIC-HM
mnt-lower:    MNT-KRNIC-AP
changed:      hostmaster@apnic.net 20000908
changed:      hostmaster@apnic.net 20010627
source:       APNIC

person:       Host Master
address:      Korea Network Information Center
address:      Narajongkeum B/D 14F, 1328-3, Seocho-dong, Seocho-ku, Seoul, 137-070,
Republic of Korea
country:      KR
phone:        +82-2-2186-4500
fax-no:       +82-2-2186-4496
e-mail:       hostmaster@nic.or.kr
nic-hdl:      HM127-AP
mnt-by:       MNT-KRNIC-AP
changed:      hostmaster@nic.or.kr 20010514
source:       APNIC
```

Host: 213.255.24.48
Reason for selection: Top SIN-FIN scanner
Hostname: h255-24-48.PD1.albacom.net
Registrant:

ARIN points to RIPE

```
inetnum:      213.255.16.0 - 213.255.31.255
netname:      IT-ALBACOM-19990524
descr:        Albacom Dial Services
country:      IT
admin-c:      AIS16-RIPE
tech-c:       AIS16-RIPE
status:       ASSIGNED PA
mnt-by:       RIPE-NCC-NONE-MNT
changed:      staff@albacom.net 20000302
source:       RIPE

route:        213.255.0.0/19
descr:        Albacom
origin:       AS8968
mnt-by:       ALBACOM-MNT
changed:      staff@albacom.net 20000302
```

```
source:       RIPE

person:       Albacom Internet Staff
address:      Albacom SpA
address:      Via V. Bianchini, 15
address:      I-00141 Roma
address:      Italy
phone:        +39-06-8741111
e-mail:       staff@albacom.net
nic-hdl:      AIS16-RIPE
notify:       staff@albacom.net
changed:      staff@albacom.net 20000301
source:       RIPE
```

Host: 211.178.63.4
Reason for selection: Top SIN-FIN scanner
Hostname: Unresolved
Registrant:

ARIN points to APNIC

```
inetnum:      211.172.0.0 - 211.199.255.255
netname:      KRNIC-KR
descr:        KRNIC
descr:        Korea Network Information Center
country:      KR
admin-c:      HM127-AP
tech-c:       HM127-AP
remarks:      ****************************************
remarks:      KRNIC is the National Internet Registry
remarks:      in Korea under APNIC. If you would like to
remarks:      find assignment information in detail
remarks:      please refer to the KRNIC Whois DB
remarks:      http://whois.nic.or.kr/english/index.html
remarks:      ****************************************
mnt-by:       APNIC-HM
mnt-lower:    MNT-KRNIC-AP
changed:      hostmaster@apnic.net 20000607
changed:      hostmaster@apnic.net 20010606
source:       APNIC

person:       Host Master
address:      Korea Network Information Center
address:      Narajongkeum B/D 14F, 1328-3, Seocho-dong, Seocho-ku, Seoul, 137-070,
Republic of Korea
country:      KR
phone:        +82-2-2186-4500
fax-no:       +82-2-2186-4496
e-mail:       hostmaster@nic.or.kr
nic-hdl:      HM127-AP
mnt-by:       MNT-KRNIC-AP
changed:      hostmaster@nic.or.kr 20010514
source:       APNIC
```

Host: 61.13.106.35
Reason for selection: Top SIN-FIN scanner
Hostname: c35.h061013106.is.net.tw
Registrant:

ARIN points to APNIC

```
inetnum:      61.13.106.0 - 61.13.106.63
netname:      NOTOINTNET
descr:        We are a foreign company
country:      TW
admin-c:      SW124-AP
```

```
tech-c:      JYB1-AP
mnt-by:      IS-NCD
changed:     billjean@mail.infoserve.com.tw 20000526
source:      APNIC

person:      Steve Wu
address:     NATO International Corp.
address:     9F-1, No.79, Sec. 1, Hsin Tai Wu Rd., Taipei
address:     Taiwan, R.O.C
country:     TW
phone:       +886-2-26989596
fax-no:      +886-2-26989731
e-mail:      billjean@mail.infoserve.com.tw
nic-hdl:     SW124-AP
mnt-by:      IS-NCD
changed:     billjean@mail.infoserve.com.tw 20000526
source:      APNIC

person:      Jean YY Bill
address:     12th Fl.-2, No. 33, Sec. 1, Min-Shen Rd., Pan-Chiao, Taipei County
address:     Taiwan, R.O.C
country:     TW
phone:       +886-2-29579972 ext. 101
fax-no:      +886-2-29572515
e-mail:      billjean@mail.infoserve.com.tw
nic-hdl:     JYB1-AP
mnt-by:      IS-NCD
changed:     billjean@mail.infoserve.com.tw 19980623
source:      APNIC
```

Host: 210.160.190.244
Reason for selection: Top SIN-FIN scanner
Hostname: linux.nangoku.co.jp
Registrant:

ARIN points to APNIC

```
inetnum:     210.160.0.0 - 210.175.255.255
netname:     JPNIC-NET-JP
descr:       Japan Network Information Center
country:     JP
admin-c:     JNIC1-AP
tech-c:      JNIC1-AP
remarks:     JPNIC Allocation Block
remarks:     Authoritative information regarding assignments and
remarks:     allocations made from within this block can also be
remarks:     queried at whois.nic.ad.jp. To obtain an English
remarks:     output query whois -h whois.nic.ad.jp x.x.x.x/e
mnt-by:      MAINT-JPNIC
changed:     apnic-ftp@nic.ad.jp 19991208
source:      APNIC

role:        Japan Network Information Center
address:     Fuundo Bldg. 3F, 1-2 Kanda-Ogawamachi
address:     Chiyoda-ku, Tokyo 101-0052, Japan
country:     JP
phone:       +81-3-5297-2311
fax-no:      +81-3-5297-2312
e-mail:      hostmaster@nic.ad.jp
admin-c:     NM6-AP
tech-c:      YM15-AP
tech-c:      IK6-AP
tech-c:      KM19-AP
nic-hdl:     JNIC1-AP
mnt-by:      MAINT-JPNIC
changed:     apnic-ftp@nic.ad.jp 19990629
source:      APNIC
```

```
inetnum:     210.160.190.240 - 210.160.190.255
netname:     NANGOKU
descr:       Nangoku Corporation
country:     JP
admin-c:     SO242JP
tech-c:      TI518JP
remarks:     This information has been partially mirrored by APNIC from
remarks:     JPNIC. To obtain more specific information, please use the
remarks:     JPNIC whois server at whois.nic.ad.jp. (This defaults to
remarks:     Japanese output, use the /e switch for English output)
remarks:     This information has been partially mirrored by APNIC from
remarks:     JPNIC. To obtain more specific information, please use the
remarks:     JPNIC whois server at whois.nic.ad.jp. (This defaults to
remarks:     Japanese output, use the /e switch for English output)
changed:     apnic-ftp@nic.ad.jp 19971030
changed:     apnic-ftp@nic.ad.jp 20010705
source:      JPNIC
```

Host: 202.98.10.70
Reason for selection: Top Port 111 scanner
Hostname: Unresolved
Registrant:

ARIN points to APNIC

```
inetnum:     202.98.0.0 - 202.98.31.255
netname:     CHINANET-JL
descr:       CHINANET Jilin province network
descr:       Data Communication Division
descr:       China Telecom
country:     CN
admin-c:     CH93-AP
tech-c:      XY1-AP
mnt-by:      MAINT-CHINANET
mnt-lower:   MAINT-CHINANET-JL
changed:     hostmaster@ns.chinanet.cn.net 20000101
source:      APNIC

person:      Chinanet Hostmaster
address:     A12,Xin-Jie-Kou-Wai Street
country:     CN
phone:       +86-10-62370437
fax-no:      +86-10-62053995
e-mail:      hostmaster@ns.chinanet.cn.net
nic-hdl:     CH93-AP
mnt-by:      MAINT-CHINANET
changed:     hostmaster@ns.chinanet.cn.net 20000101
source:      APNIC

person:      Xu Yongzhong
address:     Data Communication Bireau
address:     Ministry of Posts and Telecommunications
address:     A12 Xin-jie-kou-wai Street
address:     Beijing   100088
country:     CN
phone:       +86-10-62053991
fax-no:      +86-10-62053995
e-mail:      yzxu@publicf.bta.net.cn
nic-hdl:     XY1-AP
mnt-by:      MAINT-NULL
changed:     hostmaster@apnic.net 19960319
source:      APNIC
```

Host: 61.143.127.86

Reason for selection: Top Port 111 scanner
Hostname: Unresolved
Registrant:

ARIN points to APNIC

```
inetnum:     61.140.0.0 - 61.143.255.255
netname:     CHINANET-GD
descr:       CHINANET Guangdong province network
descr:       Data Communication Division
descr:       China Telecom
country:     CN
admin-c:     CH93-AP
tech-c:      WM12-AP
mnt-by:      MAINT-CHINANET
mnt-lower:   MAINT-CHINANET-GD
changed:     hostmaster@ns.chinanet.cn.net 20000601
source:      APNIC

person:      Chinanet Hostmaster
address:     A12,Xin-Jie-Kou-Wai Street
country:     CN
phone:       +86-10-62370437
fax-no:      +86-10-62053995
e-mail:      hostmaster@ns.chinanet.cn.net
nic-hdl:     CH93-AP
mnt-by:      MAINT-CHINANET
changed:     hostmaster@ns.chinanet.cn.net 20000101
source:      APNIC

person:      WU MIAN
address:     NO.1,RO.DONGYUANHENG,YUEXIUNAN,GUANGZHOU
country:     CN
phone:       +086-20-83788832
fax-no:      +86-20-83788825
e-mail:      wumian@gdnmc.guangzhou.gd.cn
nic-hdl:     WM12-AP
mnt-by:      MAINT-CHINANET-GD
changed:     wumian@gdnmc.guangzhou.gd.cn 20001109
source:      APNIC
```

Host: 65.27.27.1
Reason for selection: Top Port 515 scanner
Hostname: wks-65-27-27-1.kscable.com
Registrant:

```
Road Runner-Central (NETBLK-ROADRUNNER-CENTRAL)
   13241 Woodland Park Road
   Herndon, VA 20171
   US

   Netname: ROADRUNNER-CENTRAL
   Netblock: 65.24.0.0 - 65.27.255.255
   Maintainer: RRCT

   Coordinator:
      ServiceCo LLC  (ZS30-ARIN)  abuse@rr.com
      1-703-345-3416

   Domain System inverse mapping provided by:

   DNS1.RR.COM                  24.30.200.3
   DNS2.RR.COM                  24.30.201.3
   DNS3.RR.COM                  24.30.199.7
   DNS4.RR.COM                  65.24.0.172
```

```
ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

Record last updated on 14-Jun-2001.
Database last updated on 25-Jul-2001 23:06:28 EDT.
```

Host: 165.132.31.137
Reason for selection: Top Port 515 scanner
Hostname: Unresolved
Registrant:

```
Yonsei University (NET-YONSEI-NET)
   134, Shinchon-dong, Seodaemnu-gu
   Seoul, 120-749
   KR

   Netname: YONSEI-NET
   Netblock: 165.132.0.0 - 165.132.255.255

   Coordinator:
      Information systems, Yonsei university  (YI13-ARIN)  yisnet@yonsei.ac.kr
      +82-2-2123-3389

   Domain System inverse mapping provided by:

   NS.YONSEI.AC.KR             165.132.10.21
   NS2.YONSEI.AC.KR            165.132.10.41

   Record last updated on 18-Jul-2000.
   Database last updated on 25-Jul-2001 23:06:28 EDT.
```

Host: 62.242.237.21
Reason for selection: Two-way traffic on Red Worm port
Hostname: cpe.atm2-0-109178.boanxx8.customer.tele.dk
Registrant:

ARIN points to RIPE.

```
inetnum:      62.242.236.0 - 62.242.239.255
netname:      TDC-ADSL
descr:        TDC ProAccess ADSL users
country:      DK
admin-c:      PH385-RIPE
tech-c:       PH385-RIPE
status:       ASSIGNED PA
remarks:      For abuse and security issues contact
remarks:      csirt@csirt.dk, http://www.csirt.dk
mnt-by:       TDK-MNT
changed:      phns@tdk.dk 20010517
source:       RIPE


route:        62.242.0.0/15
descr:        TDC Tele Danmark
origin:       AS3292
remarks:      For abuse and security issues contact
remarks:      csirt@csirt.dk, http://www.csirt.dk
mnt-by:       AS3292-MNT
changed:      staff@ip.tele.dk 20001103
changed:      staff@ip.tele.dk 20010311
changed:      staff@ip.tele.dk 20010312
source:       RIPE


person:       Per Hansen
address:      Tele Danmark NDAI
address:      IPdrift
address:      Sletvej 30
address:      DK-8310 Tranbjerg J.
address:      Denmark
```

```
phone:        +45 8945 5889
fax-no:       +45 8945 5589
e-mail:       phns@tdk.dk
nic-hdl:      PH385-RIPE
notify:       ripe-notify@uninett.no
mnt-by:       AS3292-MNT
changed:      pha@datacon.dk400.dk 19981102
changed:      pha@datacom.dk400.dk 19990427
source:       RIPE
```

Host: 211.178.63.4
Reason for selection: Top source of Out-of-Spec packets
Hostname: Unresolved
Registrant:

ARIN points to APNIC

```
inetnum:      211.172.0.0 - 211.199.255.255
netname:      KRNIC-KR
descr:        KRNIC
descr:        Korea Network Information Center
country:      KR
admin-c:      HM127-AP
tech-c:       HM127-AP
remarks:      ****************************************
remarks:      KRNIC is the National Internet Registry
remarks:      in Korea under APNIC. If you would like to
remarks:      find assignment information in detail
remarks:      please refer to the KRNIC Whois DB
remarks:      http://whois.nic.or.kr/english/index.html
remarks:      ****************************************
mnt-by:       APNIC-HM
mnt-lower:    MNT-KRNIC-AP
changed:      hostmaster@apnic.net 20000607
changed:      hostmaster@apnic.net 20010606
source:       APNIC

person:       Host Master
address:      Korea Network Information Center
address:      Narajongkeum B/D 14F, 1328-3, Seocho-dong, Seocho-ku, Seoul, 137-070,
Republic of Korea
country:      KR
phone:        +82-2-2186-4500
fax-no:       +82-2-2186-4496
e-mail:       hostmaster@nic.or.kr
nic-hdl:      HM127-AP
mnt-by:       MNT-KRNIC-AP
changed:      hostmaster@nic.or.kr 20010514
source:       APNIC
```

Host: 211.207.15.190
Reason for selection: SYN scan of port 21
Hostname: Unresolved
Registrant:

ARIN points to APNIC

```
inetnum:      211.206.0.0 - 211.211.255.255
netname:      HANANET
descr:        Hanaro Telecom, Inc.
country:      KR
admin-c:      IS37-AP
tech-c:       SH243-AP
remarks:      ****************************************
remarks:      Allocated to KRNIC Member.
remarks:      If you would like to find assignment
```

```
remarks:     information in detail please refer to
remarks:     the KRNIC Whois Database at:
remarks:     http://whois.nic.or.kr/english/index.html
remarks:     ****************************************
mnt-by:      MNT-KRNIC-AP
mnt-lower:   MNT-KRNIC-AP
changed:     hostmaster@apnic.net 20001228
changed:     hostmaster@apnic.net 20010627
source:      APNIC

person:      Inyup Sung
address:     Hanaro Telecom Co.
address:     Kukje Electornics Cneter Bldg. 1445-3 Seocho-Dong Seocho-Ku
address:     SEOUL
address:     137-070
country:     KR
phone:       +82-2-106
fax-no:      +82-2-6266-6483
e-mail:      info@hananet.net
nic-hdl:     IS37-AP
mnt-by:      MNT-KRNIC-AP
changed:     hostmaster@nic.or.kr 20010523
source:      APNIC

person:      Seungchul Hwang
address:     Hanaro Telecom Co.
address:     Kukje Electornics Cneter Bldg., 1445-3 Seocho-Dong Seocho-Ku
address:     SEOUL
address:     137-070
country:     KR
phone:       +82-2-106
fax-no:      +82-2-6266-6483
e-mail:      info@hananet.net
nic-hdl:     SH243-AP
mnt-by:      MNT-KRNIC-AP
changed:     hostmaster@nic.or.kr 20010523
source:      APNIC
```

Host: 128.143.75.164
Reason for selection: SYN scan of port 21
Hostname: bootp-75-164.bootp.Virginia.EDU
Registrant:

```
University of Virginia (NET-VIRGINIA)
   Charlottesville, VA 22903
   US

   Netname: VIRGINIA
   Netblock: 128.143.0.0 - 128.143.255.255

   Coordinator:
      Jokl, James A.  (JAJ17-ARIN)  jaj@VIRGINIA.EDU
      (804) 924-0616

   Domain System inverse mapping provided by:

   UVAARPA.VIRGINIA.EDU         128.143.2.7
   JUNO.ACC.VIRGINIA.EDU        128.143.22.119
   NOM.VIRGINIA.EDU             128.143.3.7

   Record last updated on 21-Mar-1996.
   Database last updated on 25-Jul-2001 23:06:28 EDT.
```

Host: 208.188.3.10

Reason for selection: Portscan of MY.NET.217.142
Hostname: ns1.dallasisd.org
Registrant:

```
Dallas Independent School District (NETBLK-SBCIS40790)
   2701 W. 15th St.
   PMB 236
   Plano, TX 75075
   US

   Netname: SBCIS40790
   Netblock: 208.188.3.0 - 208.188.3.255

   Coordinator:
      Southwestern Bell Internet Services  (ZS44-ARIN)  ipadmin@swbell.net
      888-212-5411

   Record last updated on 11-Sep-1999.
   Database last updated on 25-Jul-2001 23:06:28 EDT.
```

Host: 211.23.9.162
Reason for selection: SYN scan of port 53
Hostname: Unresolved
Registrant:

ARIN points to APNIC

```
inetnum:     211.23.0.0 - 211.23.255.255
netname:     HINET-TW
descr:       CHTD, Chunghwa Telecom Co.,Ltd.
descr:       Data-Bldg.6F, No.21, Sec.21, Hsin-Yi Rd.
descr:       Taipei Taiwan 100
country:     TW
admin-c:     HN27-AP
tech-c:      HN28-AP
remarks:     This information has been partially mirrored by APNIC from
remarks:     TWNIC. To obtain more specific information, please use the
remarks:     TWNIC whois server at whois.twnic.net.
mnt-by:      TWNIC-AP
changed:     hostmaster@twnic.net 20001106
source:      APNIC

person:      HINET Network-Adm
address:     CHTD, Chunghwa Telecom Co., Ltd.
address:     Data-Bldg. 6F,  No. 21, Sec. 21, Hsin-Yi Rd.,
address:     Taipei Taiwan 100
country:     TW
phone:       +886 2 2322 3495
phone:       +886 2 2322 3442
phone:       +886 2 2344 3007
fax-no:      +886 2 2344 2513
fax-no:      +886 2 2395 5671
e-mail:      network-adm@hinet.net
nic-hdl:     HN27-AP
remarks:     same as TWNIC nic-handle HN184-TW
mnt-by:      TWNIC-AP
changed:     hostmaster@twnic.net 20000721
source:      APNIC

person:      HINET Network-Center
address:     CHTD, Chunghwa Telecom Co., Ltd.
address:     Data-Bldg. 6F,  No. 21, Sec. 21, Hsin-Yi Rd.,
address:     Taipei Taiwan 100
country:     TW
phone:       +886 2 2322 3495
phone:       +886 2 2322 3442
phone:       +886 2 2344 3007
```

```
fax-no:      +886 2 2344 2513
fax-no:      +886 2 2395 5671
e-mail:      network-center@hinet.net
nic-hdl:     HN28-AP
remarks:     same as TWNIC nic-handle HN185-TW
mnt-by:      TWNIC-AP
changed:     hostmaster@twnic.net 20000721
source:      APNIC

inetnum:     211.23.0.0 - 211.23.127.255
netname:     HINET-NET
descr:       Chunghwa Telecom Data communication Business Group
descr:       No.21, Hsin-Yi Rd., sec. 1
descr:       Taipei Taiwan
country:     TW
admin-c:     CYK-TW
tech-c:      CYK-TW
remarks:     This information has been partially mirrored by APNIC from
remarks:     TWNIC. To obtain more specific information, please use the
remarks:     TWNIC whois server at whois.twnic.net.
mnt-by:      TWNIC-AP
changed:     twnic-update@hinet.net 20001213
source:      TWNIC
```

Host: 212.17.127.36
Reason for selection: SYN scan of port 21
Hostname: TK212017127036.teleweb.at
Registrant:

ARIN points to RIPE

```
inetnum:      212.17.124.0 - 212.17.127.255
netname:      SK-15-CUSTOMER-2
descr:        Telekabel Wien GmbH
country:      AT
admin-c:      HTK1-RIPE
tech-c:       HTK1-RIPE
rev-srv:      ns1.telekabel.at
rev-srv:      ns2.telekabel.at
status:       ASSIGNED PA
notify:       hostmaster@telekabel.at
mnt-by:       TK-MNT
changed:      sbaumann@chello.at 19991125
source:       RIPE

route:        212.17.64.0/18
descr:        AT-TELEKABEL-980716
origin:       AS6830
mnt-by:       CHELLO-MNT
changed:      hostmaster@chello.at 20001207
source:       RIPE

role:         Hostmaster Telekabel Wien
address:      chello Broadband GmbH
address:      Internet Services
address:      Reumannplatz 7
address:      A-1100 Vienna
address:      Austria
phone:        +43 1 96062
fax-no:       +43 1 96062 5666
e-mail:       hostmaster@chello.at
trouble:      help@chello.at
admin-c:      AK991-RIPE
tech-c:       SB9000-RIPE
tech-c:       MH392-RIPE
tech-c:       MG872-RIPE
tech-c:       AK991-RIPE
```

```
nic-hdl:      HTK1-RIPE
notify:       hostmaster@chello.at
notify:       hm-dbm-msgs@ripe.net
mnt-by:       CHELLO-MNT
changed:      hostmaster@chello.at 20000904
source:       RIPE
```

Tools Used/Analysis Process

The log files supplied fell into three major categories: alerts, scans, and out-of-spec. The off-the-shelf tools initially selected were:

Microsoft Excel, for visualization and charting
PassiveOS.pl by Craig Smith, used to analyze the out-of-spec packets
SnortSnarf, by SiliconDefense was originally considered
Snort_stat.pl by Yen-Ming Chen
Snort_sort.pl by Andew R. Baker
The correlation scripts by Lennyy Zeltser (231) available at http://www.sans.org/y2k/practical/Lenny_Zeltser.htm.

The volume of the logs overloaded SnortSnarf. Even midrange servers were unable to analyze more than one day's worth of logs. The original version of Yen-Ming Chen's snort_stat.pl did not handle the fast alert format. The script was quickly hacked to generate useable output.

Overall, Lenny Zeltser's scripts proved to be the most useful. Additional scripts were written to generate alert and scan footprints from the CSV files. These footprints were used to create the tables in this analysis. The scripts simply reduced the CSV files down to source IP, source port, destination IP, destination port and performed quick histograms using selected key formats such as:

Source IP to detect top-talkers
Destination IP to detect top-targets
Source IP/Destination IP to detect links
Source Port/Destination Port to detect false positives for the Worm alerts.
Destination Port, used when the alert's source port is already fixed

Executive Summary

- Log data from June 3rd through July 8th were used to determine the most common alerts noted by the SNORT sensors. A large number of false-positives were caused by traffic from Multicast, RFC1918, and Windows DHCP-orphans.
- In exploring traffic from ISDNNET 212.179.0.0/16 MY.NET.150.220 appeared to be running a service on port 1234. This should be examined.
- The University should review its policy regarding Peer-to-Peer programs such as Kazaa, Napster, and Gnutella. These programs increase the risk of exposure to malicious code.
- MY.NET.97.165 should be examined for a service running on port 1372.
- MY.NET.156.55 should be examined for a service running on port 4734.
- The following servers may be infected with SubSeven:

  MY.NET.15.214
  MY.NET.146.95
  MY.NET.202.26
  MY.NET.228.50
  MY.NET.202.34
  MY.NET.208.142
  MY.NET.97.200
  MY.NET.98.123
  MY.NET.60.16
  MY.NET.98.110
  MY.NET.105.120
  MY.NET.98.228

- SYN-FIN packets should be blocked at the network border to limit the usefulness of stealth scans against the University.
- The University's policy regarding Remote Procedure Calls and Remote Printing need to be established, blocking ports 111 and port 515 respectively at the border.
- The following servers should be checked for web proxies, if found, their access control lists should be tightened to block offsite users:

  MY.NET.157.5
  MY.NET.100.203
  MY.NET.60.16
  MY.NET.60.39
  MY.NET.60.38
  MY.NET.60.8
  MY.NET.217.142
  MY.NET.218.162
  MY.NET.97.223
  MY.NET.217.58

- MY.NET.130.122 was possibly exploiting 64.213.55.2 on port 55850.
- Ports 136, 137, and 139 should be blocked at the border. NetBIOS traffic should be restricted at the very least to on-campus use only.
- There was two-way communication between 62.242.237.21 and MY.NET.201.6 that warrants further investigation.
- Analysis of Ramen traffic generated a list of 2074 potentially-infected machines. A scan of port 27374 on MY.NET.0.0/16 to generate a more accurate infection-list is recommended.
- MY.NET.160.114 appears to be running a service on port 777. This may be a multilingual webserver. This should be verified.

- UPD ports 6070 and 6072 should be examined on the following hosts:

MY.NET.110.33
MY.NET.71.248
MY.NET.180.76
MY.NET.178.222
MY.NET.108.13
MY.NET.145.166
MY.NET.145.197
MY.NET.107.4
MY.NET.15.223
MY.NET.106.184
MY.NET.106.178
MY.NET.70.92
MY.NET.109.62
MY.NET.146.17
MY.NET.111.30
MY.NET.178.219
MY.NET.110.169
MY.NET.108.16
MY.NET.178.154
MY.NET.75.103
MY.NET.15.22
MY.NET.158.25
MY.NET.104.216
MY.NET.121.23