Intrusion Detects and Analysis
GCIA Practical Assignment

Matthew Waddell

Version 2.7   New Orleans 2001

Submitted on April 4, 2001

Assignment 1

Detect # 1

Incoming Traffic:

```
15:18:38.183120 213.39.28.53.6667 > home.addr.73.30.50796: S 514319330:514319330(0) ack 1 win 8040 <mss 536> (DF)
15:19:11.902461 213.39.28.53.6667 > home.addr.40.107.50605: S 1026500771:1026500771(0) ack 1 win 8040 <mss 536> (DF)
15:19:58.617718 213.39.28.53.6667 > home.addr.65.123.23360: S 3916551146:3916551146(0) ack 1 win 8040 <mss 536> (DF)
15:20:36.426330 213.39.28.53.6667 > home.addr.111.15.8690: S 2602548537:2602548537(0) ack 1 win 8040 <mss 536> (DF)
15:21:28.192992 213.39.28.53.6667 > home.addr.112.124.26026: S 3146359024:3146359024(0) ack 1 win 8040 <mss 536> (DF)
15:22:11.628526 213.39.28.53.6667 > home.addr.14.11.62604: S 1761663679:1761663679(0) ack 1 win 8040 <mss 536> (DF)
15:22:55.150658 213.39.28.53.6667 > home.addr.79.73.25835: S 1153917478:1153917478(0) ack 1 win 8040 <mss 536> (DF)
15:23:38.032835 213.39.28.53.6667 > home.addr.159.125.28692: S 688789936:688789936(0) ack 1 win 8040 <mss 536> (DF)
15:24:09.713497 213.39.28.53.6667 > home.addr.122.22.24691: S 670236619:670236619(0) ack 1 win 8040 <mss 536> (DF)
15:24:50.314089 213.39.28.53.6667 > home.addr.4.92.3719: S 1308119622:1308119622(0) ack 1 win 8040 <mss 536> (DF)
15:25:32.173757 213.39.28.53.6667 > home.addr.124.112.59364: S 31786951:31786951(0) ack 1 win 8040 <mss 536> (DF)
…
```

Outbound Traffic:

```
15:18:38.187144 home.addr.73.30.50796 > 213.39.28.53.6667: R 1:1(0) win 0 (DF)
15:19:11.903162 home.addr.40.107.50605 > 213.39.28.53.6667: R 1:1(0) win 0 (DF)
15:19:58.618945 home.addr.65.123.23360 > 213.39.28.53.6667: R 1:1(0) win 0 (DF)
15:20:36.426784 home.addr.111.15.8690 > 213.39.28.53.6667: R 1:1(0) win 0 (DF)
15:21:28.193442 home.addr.112.124.26026 > 213.39.28.53.6667: R 1:1(0) win 0 (DF)
15:22:11.632090 home.addr.14.11.62604 > 213.39.28.53.6667: R 1:1(0) win 0 (DF)
15:22:55.151198 home.addr.79.73.25835 > 213.39.28.53.6667: R 1:1(0) win 0 (DF)
15:23:38.034713 home.addr.159.125.28692 > 213.39.28.53.6667: R 1:1(0) win 0 (DF)
15:24:09.714007 home.addr.122.22.24691 > 213.39.28.53.6667: R 1:1(0) win 0 (DF)
15:24:50.314584 home.addr.4.92.3719 > 213.39.28.53.6667: R 1:1(0) win 0 (DF)
15:25:32.174264 home.addr.124.112.59364 > 213.39.28.53.6667: R 1:1(0) win 0 (DF)
…
```

Source of trace:

This trace originated from an ISP in Paris France (http://www.planet-work.com).

Detect generated by:

This detect was generated by Shadow and TCPDump.

Was source spoofed?:

The source address was most likely spoofed, explanation below.

Description of attack:

The TCPDump console initially showed this attack with outgoing RST's to port 6667 from several of our internal IP Addresses.
All incoming packets were SYN ACK's from a single IP Address.
Our machines responded to the SYN ACK's with RST's, not the normal end of handshake ACK's.
Since there were no initial SYN packets coming from our network, the SYN ACK's are most likely the "responses" from another machine.

The time spacing between inbound packets leads me to believe that other hosts are being spoofed as well.

After finishing my analysis I put the spoofed IP into a browser window.
The page that displayed (http://www.planet-work.com) was loading very slowly; leading me to believe that the DOS attack was effectively slowing down their connection.

Further information on Port 6667 shows ScheduleAgent, Trinity, and WinSatan to be active Trojans that use this port.

IRC can also be seen most often running on this port.

Attack mechanism:

Our network is not being directly attacked. This is probably a Denial of Service attack against the remote host.

Correlations:

I was unable to find any matching correlations with this IP address in our database.
This may have been the only time that we were spoofed in correlation with this address.

Active targeting:

This was not active targeting towards are own network, but perhaps towards the ISP located at 213.39.28.53.

Severity:

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

Criticality:	1	Home network is not the target of the attack.
Lethality:	1	Responsive traffic. Not causing enough traffic to worry about network congestion.
Sys Counters:	5	Nothing to defend against in this case.
Net Counters:	5	Most unsolicited packets are being dropped by our network firewall.

Severity:	-8	

Defensive Recommendations:

No immediate recommendations. If traffic were to increase so as to become a problem, I would coordinate with the gateway in order to drop packets coming from IP address 213.39.28.53 at the Internet border gateway router.

Multiple Choice:

What non-business application can most often be found on port 6667?
a. SSH
b. SNMP
c. IRC
d. ICQ

Answer:
c. IRC frequently runs on port 6667, along with a few Trojans such as ScheduleAgent, Trinity, and WinSatan.

Detect # 2

Incoming traffic:

```
03:37:45.408275 62.13.12.6.54321 > home.addr.192.241.54321: S 159034077:159034077(0) win 40
03:37:45.408280 62.13.12.6.54321 > home.addr.192.240.54321: S 159034077:159034077(0) win 40
03:37:45.408283 62.13.12.6.54321 > home.addr.192.242.54321: S 159034077:159034077(0) win 40
03:37:45.408287 62.13.12.6.54321 > home.addr.192.243.54321: S 159034077:159034077(0) win 40
03:37:45.409953 62.13.12.6.54321 > home.addr.192.244.54321: S 159034077:159034077(0) win 40
03:37:45.409958 62.13.12.6.54321 > home.addr.192.245.54321: S 159034077:159034077(0) win 40
03:37:45.411103 62.13.12.6.54321 > home.addr.192.246.54321: S 159034077:159034077(0) win 40
03:37:45.411106 62.13.12.6.54321 > home.addr.192.247.54321: S 159034077:159034077(0) win 40
03:37:45.413193 62.13.12.6.54321 > home.addr.192.248.54321: S 159034077:159034077(0) win 40
03:37:45.414870 62.13.12.6.54321 > home.addr.192.249.54321: S 159034077:159034077(0) win 40
03:37:45.414875 62.13.12.6.54321 > home.addr.192.250.54321: S 159034077:159034077(0) win 40
03:37:45.415321 62.13.12.6.54321 > home.addr.192.251.54321: S 159034077:159034077(0) win 40
03:37:45.438627 62.13.12.6.54321 > home.addr.192.252.54321: S 159034077:159034077(0) win 40
03:37:45.441039 62.13.12.6.54321 > home.addr.192.253.54321: S 159034077:159034077(0) win 40
03:37:45.441043 62.13.12.6.54321 > home.addr.192.254.54321: S 159034077:159034077(0) win 40
03:37:45.442229 62.13.12.6.54321 > home.addr.192.255.54321: S 159034077:159034077(0) win 40
03:37:45.447592 62.13.12.6.54321 > home.addr.193.1.54321: S 159034077:159034077(0) win 40
03:37:45.447597 62.13.12.6.54321 > home.addr.193.2.54321: S 159034077:159034077(0) win 40
03:37:45.448783 62.13.12.6.54321 > home.addr.193.3.54321: S 159034077:159034077(0) win 40
03:37:45.449395 62.13.12.6.54321 > home.addr.193.4.54321: S 159034077:159034077(0) win 40
03:37:45.451603 62.13.12.6.54321 > home.addr.193.5.54321: S 159034077:159034077(0) win 40
03:37:45.453164 62.13.12.6.54321 > home.addr.193.6.54321: S 159034077:159034077(0) win 40
03:37:45.453172 62.13.12.6.54321 > home.addr.193.10.54321: S 159034077:159034077(0) win 40
03:37:45.453658 62.13.12.6.54321 > home.addr.193.13.54321: S 159034077:159034077(0) win 40
03:37:45.454677 62.13.12.6.54321 > home.addr.193.17.54321: S 159034077:159034077(0) win 40
03:37:45.455227 62.13.12.6.54321 > home.addr.193.22.54321: S 159034077:159034077(0) win 40
03:37:45.458898 62.13.12.6.54321 > home.addr.193.26.54321: S 159034077:159034077(0) win 40
03:37:45.460046 62.13.12.6.54321 > home.addr.193.27.54321: S 159034077:159034077(0) win 40
03:37:45.461396 62.13.12.6.54321 > home.addr.193.28.54321: S 159034077:159034077(0) win 40
03:37:45.462667 62.13.12.6.54321 > home.addr.193.29.54321: S 159034077:159034077(0) win 40
03:37:45.462670 62.13.12.6.54321 > home.addr.193.30.54321: S 159034077:159034077(0) win 40
03:37:45.462674 62.13.12.6.54321 > home.addr.193.31.54321: S 159034077:159034077(0) win 40
03:37:45.463365 62.13.12.6.54321 > home.addr.193.32.54321: S 159034077:159034077(0) win 40
```

```
03:37:45.464795 62.13.12.6.54321 > home.addr.193.33.54321: S 159034077:159034077(0) win 40
03:37:45.464799 62.13.12.6.54321 > home.addr.193.34.54321: S 159034077:159034077(0) win 40
03:37:45.464801 62.13.12.6.54321 > home.addr.193.35.54321: S 159034077:159034077(0) win 40
03:37:45.464805 62.13.12.6.54321 > home.addr.193.36.54321: S 159034077:159034077(0) win 40
03:37:45.464808 62.13.12.6.54321 > home.addr.193.37.54321: S 159034077:159034077(0) win 40
…
```

Source of trace:

This scan came from an Internet Service Provider in Sweden.

Detect generated by:

This detect was generated by Shadow and TCP Dump.

Was source spoofed?:

This source was probably not spoofed; it is most likely reconnaissance so the scanner would want the information to come back to his computer.

Description of attack:

The scanner is looking for responses from port 54321.
The scan sends initial SYN packets with identical sequence numbers to numerous IP Addresses. RST's that are sent back indicate responsive machines with a service running on port 54321.

A search for information on port 54321 show this as being either "Back Orifice 2000", a Trojan called "School Bus", or an exploit known as "loadavg".

- Back Orifice is a Trojan that only affects computers running Microsoft Windows.
  Since all of the machines on this network are running Unix and Linux, the scan does not really concern us.

- School Bus is a "Remote DoS exploit for NetBus 1.7 client."
  Since our network is not running NetBus, and the traffic is nowhere near Denial of Service proportions, this does not concern us.

- Since the exploit "loadavg" travels over UDP, and this is a TCP scan, it is most likely not an exploit being initiated against our network.

Attack mechanism:

This scan came from port 54321 to port 54321; also all of the sequence numbers are the same so the packets are most likely crafted.
It is possible that this is a script kiddie running the server side of Back Orifice on their machine looking for clients to connect to.

Correlations:

Robert V. McMillen Jr. has a very good Overview of Back Orifice 2000 that can be found at: http://www.sans.org/infosecFAQ/malicious/back_orifice.htm

Active targeting:

This scan is probably not targeting us; it seems like just a scan passing through our network. Since no machines responded to this scan the attacker did not get any information back either.

Severity:

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

Criticality:	1	This is the lowest score that I could give this scan, there are no windows machines on this particular network, only UNIX servers.
Lethality:	1	Again no Windows based machines on this network.
Sys Counters:	5	No Countermeasures necessary at this time.
Net Counters:	5	The firewall is set up for UNIX machines, however blocking 54321 might not be a bad idea.

Severity:	-8	

Defensive Recommendations:

None at this time, perhaps block port 54321 at the firewall to prevent incoming packets.

Multiple Choice:

What Windows Trojan uses port 54321 by default?

a) Back Orifice 2000.
b) Sub Seven
c) RemoConChubo
d) Portmapper

The Correct Answer is a.

Detect # 3

<u>NID alert via Email:</u>

Date: 3/9/2001 9:04 AM
Subject: NID PortScan Alert

Port Scan detected.
Origination MAC address: x:x:x:8d:f9:c4
Destination MAC address: x:x:x:12:57:dd
Origination IP address : bad.home.addr.64.63
Destination IP address : scanned.home.addr.153

<u>TCPDump Data:</u>

```
09:04:14.415418 bad.home.addr.64.63.20 > scanned.home.addr.153.1: S 1234:1234(0) win 4096
09:04:14.415542 bad.home.addr.64.63.20 > scanned.home.addr.153.1: S 1234:1234(0) win 4096
09:04:14.472771 bad.home.addr.64.63.20 > scanned.home.addr.153.7: S 1234:1234(0) win 4096
09:04:14.472897 bad.home.addr.64.63.20 > scanned.home.addr.153.7: S 1234:1234(0) win 4096
09:04:14.535312 bad.home.addr.64.63.20 > scanned.home.addr.153.9: S 1234:1234(0) win 4096
09:04:14.535488 bad.home.addr.64.63.20 > scanned.home.addr.153.9: S 1234:1234(0) win 4096
09:04:14.603330 bad.home.addr.64.63.20 > scanned.home.addr.153.11: S 1234:1234(0) win 4096
09:04:14.603488 bad.home.addr.64.63.20 > scanned.home.addr.153.11: S 1234:1234(0) win 4096
09:04:14.663929 bad.home.addr.64.63.20 > scanned.home.addr.153.13: S 1234:1234(0) win 4096
09:04:14.664117 bad.home.addr.64.63.20 > scanned.home.addr.153.13: S 1234:1234(0) win 4096
09:04:14.726225 bad.home.addr.64.63.20 > scanned.home.addr.153.15: S 1234:1234(0) win 4096
09:04:14.726386 bad.home.addr.64.63.20 > scanned.home.addr.153.15: S 1234:1234(0) win 4096
09:04:14.800974 bad.home.addr.64.63.20 > scanned.home.addr.153.19: S 1234:1234(0) win 4096
09:04:14.801117 bad.home.addr.64.63.20 > scanned.home.addr.153.19: S 1234:1234(0) win 4096
09:04:14.864544 bad.home.addr.64.63.20 > scanned.home.addr.153.20: S 1234:1234(0) win 4096
09:04:14.864687 bad.home.addr.64.63.20 > scanned.home.addr.153.20: S 1234:1234(0) win 4096
09:04:14.924490 bad.home.addr.64.63.20 > scanned.home.addr.153.21: S 1234:1234(0) win 4096
09:04:14.924662 bad.home.addr.64.63.20 > scanned.home.addr.153.21: S 1234:1234(0) win 4096
```

```
09:04:14.987991 bad.home.addr.64.63.20 > scanned.home.addr.153.22: S 1234:1234(0) win 4096
09:04:14.988100 bad.home.addr.64.63.20 > scanned.home.addr.153.22: S 1234:1234(0) win 4096
09:04:15.054482 bad.home.addr.64.63.20 > scanned.home.addr.153.23: S 1234:1234(0) win 4096
09:04:15.054600 bad.home.addr.64.63.20 > scanned.home.addr.153.23: S 1234:1234(0) win 4096
09:04:15.113005 bad.home.addr.64.63.20 > scanned.home.addr.153.25: S 1234:1234(0) win 4096
09:04:15.113103 bad.home.addr.64.63.20 > scanned.home.addr.153.25: S 1234:1234(0) win 4096
09:04:15.198699 bad.home.addr.64.63.20 > scanned.home.addr.153.37: S 1234:1234(0) win 4096
09:04:15.198801 bad.home.addr.64.63.20 > scanned.home.addr.153.37: S 1234:1234(0) win 4096
09:04:15.254265 bad.home.addr.64.63.20 > scanned.home.addr.153.43: S 1234:1234(0) win 4096
09:04:15.254353 bad.home.addr.64.63.20 > scanned.home.addr.153.43: S 1234:1234(0) win 4096
09:04:15.319680 bad.home.addr.64.63.20 > scanned.home.addr.153.53: S 1234:1234(0) win 4096
09:04:15.319775 bad.home.addr.64.63.20 > scanned.home.addr.153.53: S 1234:1234(0) win 4096
09:04:15.394302 bad.home.addr.64.63.20 > scanned.home.addr.153.57: S 1234:1234(0) win 4096
09:04:15.394454 bad.home.addr.64.63.20 > scanned.home.addr.153.57: S 1234:1234(0) win 4096
09:04:15.459143 bad.home.addr.64.63.20 > scanned.home.addr.153.70: S 1234:1234(0) win 4096
09:04:15.459343 bad.home.addr.64.63.20 > scanned.home.addr.153.70: S 1234:1234(0) win 4096
09:04:15.529355 bad.home.addr.64.63.20 > scanned.home.addr.153.77: S 1234:1234(0) win 4096
09:04:15.529476 bad.home.addr.64.63.20 > scanned.home.addr.153.77: S 1234:1234(0) win 4096
09:04:15.644876 bad.home.addr.64.63.20 > scanned.home.addr.153.80: S 1234:1234(0) win 4096
09:04:15.644990 bad.home.addr.64.63.20 > scanned.home.addr.153.80: S 1234:1234(0) win 4096
09:04:15.716672 bad.home.addr.64.63.20 > scanned.home.addr.153.87: S 1234:1234(0) win 4096
09:04:15.716770 bad.home.addr.64.63.20 > scanned.home.addr.153.87: S 1234:1234(0) win 4096
09:04:15.769629 bad.home.addr.64.63.20 > scanned.home.addr.153.88: S 1234:1234(0) win 4096
09:04:15.769735 bad.home.addr.64.63.20 > scanned.home.addr.153.88: S 1234:1234(0) win 4096
09:04:15.863185 bad.home.addr.64.63.20 > scanned.home.addr.153.95: S 1234:1234(0) win 4096
09:04:15.863302 bad.home.addr.64.63.20 > scanned.home.addr.153.95: S 1234:1234(0) win 4096
09:04:15.945075 bad.home.addr.64.63.20 > scanned.home.addr.153.101: S 1234:1234(0) win 4096
09:04:15.945178 bad.home.addr.64.63.20 > scanned.home.addr.153.101: S 1234:1234(0) win 4096
09:04:16.008283 bad.home.addr.64.63.20 > scanned.home.addr.153.102: S 1234:1234(0) win 4096
09:04:16.008444 bad.home.addr.64.63.20 > scanned.home.addr.153.102: S 1234:1234(0) win 4096
09:04:16.069456 bad.home.addr.64.63.20 > scanned.home.addr.153.103: S 1234:1234(0) win 4096
…
```

Source of trace:

This trace came from a machine within our own network.

Detect generated by:

NID generated initial email alert; data was further researched using TCP Dump.

Was source spoofed:

This scan was not a spoofed address (see explanation below.)

Description of attack:

A computer inside our network started port scanning other computers on our network (all computers are within the firewall.)
The initial computer was scanning one IP address for it's full range of open ports. The scan was coming from port 20.
All sequence numbers were identical (1234), when the scan had reached port 65535 it stopped and moved on to another IP address covering a total of five IP addresses in all.

Correlations:

After calling our Incident Response team and making a fuss over an internal scan, compromised machine, etc… We discovered that the scanner was our own penetration team doing a routine scan of different systems.
The negligence here really falls on two people:
The scanner should not have been scanning systems from a different IP Address (his normal system has exception rules in the NID and TCP Dump).
And myself, for not cross checking my scan alert email that I am sent before penetration testing is conducted, the time of the scan and the IP Addresses that he was scanning were plainly listed so that I would not cause alerts over them.

Severity:

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

Criticality:	2	Unix and Windows machines were involved in this scan.
Lethality:	1	Not an attack, just a vulnerability scan (very trusted scanner doing his job to patch systems.)
Sys Counters:	4	Two systems showed open ports that would need to be patched.
Net Counters:	1	Attack came from within the firewall.

Severity:	-2	

Defensive Recommendations:

This was a man that I know very well doing his job. This job involves scanning systems for vulnerabilities and then patching these systems. He is one of three people allowed to run internal scans on our network.
If this had been Joe User that had downloaded a scanning tool, or worse a machine under remote control (such as Loki), I would have been really concerned and the machine would have been taken off of the network immediately. Forensics would have been performed and life would go on.

Multiple Choice:

What Intrusion Detection tool will catch a port scan that is contained inside your firewall?

a) Tripwire
b) A Shadow sensor in the DMZ
c) A NID inside your firewall
d) Firewall

The correct answer is C.

Detect # 4

Here is the original email from our networks "South Side" System Administrator:

Hi there,

There were some FTP attempts on the "south.side.addr.edu" side.
Here is the information that we have gathered so far:

Server: south.side.addr.edu
Address: 123.123.123.193

Name: cc8250-b.zwoll1.ov.nl.home.com
Address: 213.51.101.142

```
Mar 14 21:12:32 6D: jupiter.south.side.addr.58 ftpd[128621]: connection from 213.51.101.142
Mar 14 21:12:32 5E: jupiter.south.side.addr.58 ftpd[128621]: ANONYMOUS FTP LOGIN REFUSED FROM 213.51.101.142
Mar 14 21:12:37 6C: dispersion.south.side.addr ftpd[91890]: connect from 213.51.101.142
Mar 14 21:12:41 5D: dispersion.south.side.addr ftpd[91890]: FTP LOGIN REFUSED
(ftp not in /etc/passwd) FROM 213.51.101.142 [213.51.101.142], anonymous
Mar 14 21:12:41 6C: kueijung.south.side.addr ftpd[40418]: connect from 213.51.101.142
Mar 14 21:12:41 6D: kueijung.south.side.addr ftpd[40418]: connection from 213.51.101.142
Mar 14 21:12:42 5E: kueijung.south.side.addr ftpd[40418]: ANONYMOUS FTP LOGIN REFUSED FROM 213.51.101.142
Mar 14 21:12:42 6C: ceres.south.side.addr ftpd[87043]: connect from 213.51.101.142
Mar 14 21:12:42 6D: ceres.south.side.addr ftpd[87043]: connection from 213.51.101.142
Mar 14 21:12:42 6C: earth.south.side.addr ftpd[100576]: connect from 213.51.101.142
Mar 14 21:12:42 6D: earth.south.side.addr ftpd[100576]: connection from 213.51.101.142
Mar 14 21:12:42 5L: aster.south.side.addr ftpd[3339]: FTP LOGIN REFUSED (ftp not in /etc/passwd) FROM cc8250-b.zwoll1.ov.nl.home.com
[213.51.101.142], anonymous
Mar 14 21:12:42 6C: dust.south.side.addr ftpd[35036]: connect from 213.51.101.142
Mar 14 21:12:42 6K: eos.south.side.addr xinetd[450]: START: ftp pid=25663 from=213.51.101.142
Mar 14 21:12:42 6D: dust.south.side.addr ftpd[35036]: connection from 213.51.101.142
Mar 14 21:12:42 5E: earth.south.side.addr ftpd[100576]: ANONYMOUS FTP LOGIN REFUSED FROM 213.51.101.142
Mar 14 21:12:42 5E: disease.south.side.addr ftpd[5268]: ANONYMOUS FTP LOGIN REFUSED FROM 213.51.101.142
Mar 14 21:12:43 5E: dust.south.side.addr ftpd[35036]: ANONYMOUS FTP LOGIN REFUSED FROM 213.51.101.142
Mar 14 21:12:43 5L: eos.south.side.addr ftpd[25663]: FTP LOGIN REFUSED (ftp not in /etc/passwd) FROM cc8250-b.zwoll1.ov.nl.home.com
[213.51.101.142], anonymous
Mar 14 21:12:43 6C: fog.south.side.addr ftpd[414326]: connect from 213.51.101.142
Mar 14 21:12:43 6D: fog.south.side.addr ftpd[414326]: connection from 213.51.101.142
Mar 14 21:12:43 5L: moon.south.side.addr ftpd[11440]: FTP LOGIN REFUSED (ftp not in /etc/passwd) FROM cc8250-b.zwoll1.ov.nl.home.com
[213.51.101.142], anonymous
Mar 14 21:12:43 5L: glory.south.side.addr ftpd[1298]: FTP LOGIN REFUSED
(ftp not in /etc/passwd) FROM cc8250-b.zwoll1.ov.nl.home.com [213.51.101.142], anonymous
Mar 14 21:12:43 6D: smoke.south.side.addr ftpd[35422]: connection from 213.51.101.142
```

The Initial Scan:

```
21:12:08.492705 213.51.101.142.3144 > south.side.addr.1.21: S 23381297:23381297(0) win 65535 <mss 1460,nop,nop,sackOK>
21:12:08.502494 213.51.101.142.3145 > south.side.addr.2.21: S 23381298:23381298(0) win 65535 <mss 1460,nop,nop,sackOK>
21:12:08.502499 213.51.101.142.3146 > south.side.addr.3.21: S 23381299:23381299(0) win 65535 <mss 1460,nop,nop,sackOK>
21:12:08.503032 213.51.101.142.3147 > south.side.addr.4.21: S 23381300:23381300(0) win 65535 <mss 1460,nop,nop,sackOK>
21:12:08.503643 213.51.101.142.3148 > south.side.addr.5.21: S 23381302:23381302(0) win 65535 <mss 1460,nop,nop,sackOK>
21:12:08.504339 213.51.101.142.3149 > south.side.addr.6.21: S 23381303:23381303(0) win 65535 <mss 1460,nop,nop,sackOK>
21:12:08.504342 213.51.101.142.3150 > south.side.addr.7.21: S 23381304:23381304(0) win 65535 <mss 1460,nop,nop,sackOK>
21:12:08.504790 213.51.101.142.3151 > south.side.addr.8.21: S 23381305:23381305(0) win 65535 <mss 1460,nop,nop,sackOK>
21:12:08.505443 213.51.101.142.3152 > south.side.addr.9.21: S 23381306:23381306(0) win 65535 <mss 1460,nop,nop,sackOK>
21:12:08.506097 213.51.101.142.3153 > south.side.addr.10.21: S 23381308:23381308(0) win 65535 <mss 1460,nop,nop,sackOK>
21:12:08.511830 213.51.101.142.3154 > south.side.addr.11.21: S 23381309:23381309(0) win 65535 <mss 1460,nop,nop,sackOK>
21:12:08.512365 213.51.101.142.3155 > south.side.addr.12.21: S 23381310:23381310(0) win 65535 <mss 1460,nop,nop,sackOK>
21:12:08.512814 213.51.101.142.3156 > south.side.addr.13.21: S 23381312:23381312(0) win 65535 <mss 1460,nop,nop,sackOK>
21:12:08.513405 213.51.101.142.3157 > south.side.addr.14.21: S 23381313:23381313(0) win 65535 <mss 1460,nop,nop,sackOK>
```

```
21:12:08.514658 213.51.101.142.3158 > south.side.addr.15.21: S 23381314:23381314(0) win 65535 <mss 1460,nop,nop,sackOK>
```

Corresponding TCP Dump Information:

```
21:12:32.335527 213.51.101.142.3278 > south.side.addr.60.21: S 23405129:23405129(0) win 65535 <mss 1460,nop,nop,sackOK>
21:12:32.337101 south.side.addr.60.21 > 213.51.101.142.3278: S 3230101893:3230101893(0) ack 23405130 win 33120 <mss 1380> (DF)
21:12:32.553252 213.51.101.142.3278 > south.side.addr.60.21: . ack 1 win 65535
21:12:32.734259 south.side.addr.60.21 > 213.51.101.142.3278: P 1:70(69) ack 1 win 33120 (DF)
21:12:32.863215 213.51.101.142.3278 > south.side.addr.60.21: P 1:17(16) ack 70 win 65466
21:12:32.925614 south.side.addr.60.21 > 213.51.101.142.3278: P 70:99(29) ack 17 win 33120 (DF)
21:12:33.064553 213.51.101.142.3278 > south.side.addr.60.21: F 17:17(0) ack 99 win 65437
21:12:33.064882 213.51.101.142.3278 > south.side.addr.60.21: R 23405147:23405147(0) win 0
21:12:33.065375 213.51.101.142.3280 > south.side.addr.61.21: S 23405828:23405828(0) win 65535 <mss 1460,nop,nop,sackOK>
21:12:33.066347 south.side.addr.61.21 > 213.51.101.142.3280: S 1347575041:1347575041(0) ack 23405829 win 33120 <mss 1380> (DF)
21:12:33.200038 213.51.101.142.3280 > south.side.addr.61.21: . ack 1 win 65535
21:12:33.211073 south.side.addr.61.21 > 213.51.101.142.3280: P 1:71(70) ack 1 win 33120 (DF)
21:12:33.369441 213.51.101.142.3280 > south.side.addr.61.21: P 1:17(16) ack 71 win 65465
21:12:33.394785 south.side.addr.61.21 > 213.51.101.142.3280: P 71:100(29) ack 17 win 33120 (DF)
21:12:33.525069 213.51.101.142.3280 > south.side.addr.61.21: F 17:17(0) ack 100 win 65436
21:12:33.525531 south.side.addr.61.21 > 213.51.101.142.3280: . ack 18 win 33120 (DF)
21:12:33.525571 213.51.101.142.3280 > south.side.addr.61.21: R 23405846:23405846(0) win 0
21:12:33.525638 south.side.addr.61.21 > 213.51.101.142.3280: P 100:137(37) ack 18 win 33120 (DF)
21:12:33.536213 213.51.101.142.3282 > south.side.addr.62.21: S 23406297:23406297(0) win 65535 <mss 1460,nop,nop,sackOK>
21:12:33.538210 south.side.addr.62.21 > 213.51.101.142.3282: S 2060488902:2060488902(0) ack 23406298 win 60720 <mss 1380,nop,nop,sackOK> (DF)
21:12:33.653839 213.51.101.142.3280 > south.side.addr.61.21: R 23405846:23405846(0) win 0
…
```

Source of trace:

The source of this trace is from our sister network shown here as the south side. The FTP Login Attempts came from an @Home type of ISP located in the Netherlands.

Detect generated by:

These detects were picked up by the South Side system logs. These system logs are grepped for ANONYMOUS, REFUSED, and other signs of abuse. The Network Administrator shows here that there were several initial FTP attempts
Further information was gathered using TCP Dump and Shadow to search for attempts outside of the system logs.

Was source spoofed?:

This source address was probably not spoofed. This was an attempt at gathering information, while simultaneously attempting to log into any Anonymous FTP servers. Since the information contained in this scan is valuable to the attacker, and since SYN ACK's were sent back in response to FTP connections the attacker is most likely seeing the traffic come back to their own address.

Description of attack:

An attacker is scanning our network for responsive FTP ports, if a response are sent back, an attempt is made to connect to the FTP server with an anonymous login.
Initially the network was scanned with SYN packets using identical sequence numbers. When an FTP server responds to a SYN with a SYN ACK, the attacker sends a PSH (the anonymous login). FTP servers that have anonymous login turned off respond with a RST. Since no connection was made the attacker continues scanning where the scan left off.

The CVE website lists this as a candidate for inclusion in the CVE list [CAN-1999-0497 (under review)] with a description of: Anonymous FTP is enabled. (Interestingly, Northcut voted against allowing this to be allowed in the CVE list.)

Attack mechanism:

The time stamps indicate that the attacker may be using some sort of program to scan.
It is unknown what would happen if an FTP server with anonymous login turned on responded since all of the FTP servers in our network have that option disabled.

Correlations:

I was unable to find any other correlation with this particular IP Address in our own database of intruders.
However, I did send a nice little email to the offending IP abuse department with a tracking number. They in turn assigned me a tracking number and sent me a canned "were sorry that this happened we are looking into the matter. I followed up on this a few days later, but received the same canned response as I did earlier. I have yet to hear from any live person on their end.

Active targeting:

Possibly, this scan was targeted at our FTP servers. Even though no anonymous logins were allowed the scanner did get a pretty picture of where all of the FTP servers were located on our network.
This could open us up to future attacks from the scanner in the future, since he now has a pretty picture of where our FTP servers are residing.

Severity:

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

Criticality:	4	These are FTP servers holding valuable data, now someone knows where they are located.
Lethality:	2	Attacker is very unlikely to succeed, however I decided to rate this just above a 1 because I do not know the attack mechanism.
Sys Counters:	5	Modern, well patched machines running Redhat 7.
Net Counters:	3	Firewall exists, but is full of holes where it blocks these machines. Not adequate protection in my opinion, but remote access is needed.

Severity:	-2	

Defensive Recommendations:

There wasn't much that I could do here since Anonymous FTP is already turned of on all of the FTP servers.
I did contact the attacking ISP in order to notify them of the attacking system.

Multiple Choice:

Your network has anonymous FTP disabled on all of your machines. You are concerned when you see an unsuccessful anonymous FTP scan come across your entire network. Why are you concerned?

a) Someone has enabled Anonymous FTP Login
b) Your FTP servers are compromised
c) Network countermeasures failed to stop this attacker.
d) Someone has just mapped your FTP Servers

The correct answer is D.

Detect # 5

```
19:15:08.399179    72.94.113.42.6637 > 123.123.17.113.17767: P 441053228:441053368(140) ack 0 win 49657
19:15:08.399179    72.94.113.42.6637 > 123.123.17.113.17767: P 0:140(140) ack 1 win 49657
19:15:08.400450    136.244.103.129.2834 > 123.123.17.113.50075: P 2731223780:2731223920(140) ack 2464485745 win 42804
19:15:08.400450    136.244.103.129.2834 > 123.123.17.113.50075: P 0:140(140) ack 1 win 42804
19:15:08.402269    0.0.0.110.4883 > 123.123.17.113.13797: . 371388060:371388200(140) ack 2464485745 win 2728
19:15:08.402269    0.0.0.110.4883 > 123.123.17.113.13797: . 0:140(140) ack 1 win 2728
19:15:08.402974    96.81.221.31 > 123.123.17.113: icmp: time stamp request
19:15:08.402974    96.81.221.31 > 123.123.17.113: icmp: time stamp request
19:15:08.403672    252.199.134.3.40560 > 123.123.17.113.58649: P 1879842993:1879843133(140) ack 2464485745 win 12134
19:15:08.403672    252.199.134.3.40560 > 123.123.17.113.58649: P 0:140(140) ack 1 win 12134
19:15:08.404394    181.193.235.66.14117 > 123.123.17.113.80: P 3406385502:3406385642(140) ack 2464485745 win 28544 urg 0
19:15:08.404394    181.193.235.66.14117 > 123.123.17.113.80: P 0:140(140) ack 1 win 28544 urg 0
19:15:08.406015    152.16.118.143.34539 > 123.123.17.113.42392: P 1425521400:1425521540(140) ack 2464485745 win 44837
19:15:08.406015    152.16.118.143.34539 > 123.123.17.113.42392: P 0:140(140) ack 1 win 44837
19:15:08.407985    208.171.254.217.1886 > 123.123.17.113.32611: P 2338519159:2338519299(140) ack 2464485745 win 5440
19:15:08.407985    208.171.254.217.1886 > 123.123.17.113.32611: P 0:140(140) ack 1 win 5440
19:15:08.408722    155.0.58.138.60053 > 123.123.17.113.22095: P 652013555:652013695(140) ack 2464485745 win 40381
19:15:08.408722    155.0.58.138.60053 > 123.123.17.113.22095: P 0:140(140) ack 1 win 40381
19:15:08.409358    141.116.133.0.47284 > 123.123.17.113.22782: P 459781386:459781526(140) ack 2464485745 win 39843 urg 0
19:15:08.409358    141.116.133.0.47284 > 123.123.17.113.22782: P 0:140(140) ack 1 win 39843 urg 0
19:15:08.410003    189.204.153.33.6789 > 123.123.17.113.19274: P 727428724:727428864(140) ack 2464485745 win 57587
19:15:08.410003    189.204.153.33.6789 > 123.123.17.113.19274: P 0:140(140) ack 1 win 57587
19:15:08.410555    184.38.129.30.41456 > 123.123.17.113.0: udp 4294967288
19:15:08.410555    184.38.129.30.41456 > 123.123.17.113.0: udp 4294967288
19:15:08.411525    251.246.33.109.23890 > 123.123.17.113.3046: P 243286881:243287021(140) ack 2464485745 win 63303
19:15:08.411525    251.246.33.109.23890 > 123.123.17.113.3046: P 0:140(140) ack 1 win 63303
19:15:08.412257    120.176.109.27.16944 > 123.123.17.113.55014: . 2170455944:2170456084(140) ack 2464485745 win 9881
19:15:08.412257    120.176.109.27.16944 > 123.123.17.113.55014: . 0:140(140) ack 1 win 9881
19:15:08.413767    0.0.19.167.283 > 123.123.17.113.80: P 2283670867:2283670867(0) ack 2464485745 win 57633 urg 0
19:15:08.413767    0.0.19.167.283 > 123.123.17.113.80: P 0:0(0) ack 1 win 57633 urg 0
19:15:08.414343    7.128.175.6.10503 > 123.123.17.113.0: udp 4294967288
19:15:08.414343    7.128.175.6.10503 > 123.123.17.113.0: udp 4294967288
19:15:08.415203    119.9.111.64.20221 > 123.123.17.113.51502: P 1660683398:1660683538(140) ack 2464485745 win 16615 urg 0
19:15:08.415203    119.9.111.64.20221 > 123.123.17.113.51502: P 0:140(140) ack 1 win 16615 urg 0
```

```
19:15:08.415905    101.141.161.1.46238 > 123.123.17.113.58102: P 2922617126:2922617266(140) ack 2464485745 win 20733
19:15:08.415905    101.141.161.1.46238 > 123.123.17.113.58102: P 0:140(140) ack 1 win 20733
19:15:08.416468    0.0.234.96.51335 > 123.123.17.113.0: udp 4294967288
19:15:08.416468    0.0.234.96.51335 > 123.123.17.113.0: udp 4294967288
19:15:08.417300    110.78.201.10.34627 > 123.123.17.113.35528: . 265814290:265814430(140) ack 2464485745 win 2464
19:15:08.417300    110.78.201.10.34627 > 123.123.17.113.35528: . 0:140(140) ack 1 win 2464
19:15:08.418783    0.0.0.110.39810 > 123.123.17.113.59172: . 635646050:635646190(140) ack 2464485745 win 18871 urg 0
…
```

Source of trace:

This trace was generated on a small test network. One computer was used to run stick against another computer; a third computer on that network was running TCP Dump in order to capture the data.

Detect generated by:

This detect was generated by TCP Dump.

Was source spoofed:

All of the source addresses are spoofed in this attack, see description of attack below.

Note: Some of the host names do resolve, for example:
136.244.103.129 is Connecticut College and 141.116.133.0 is the Army Information Systems Command-Pentagon.
Others IP Addresses that were listed are reserved addresses.

Description of attack:

One complete second of stick resulted in over 2000 lines of TCPDump data.

An IDS knows how to process alarms based on a rule set that it has been programmed with. A small number of alarms all occurring at the same time might be forgivable by the IDS, however the stick console sends out over 450 known attack signatures. The IDS's CPU will jump up to 100% and begin logging alarms, sending out alert emails, and possibly even dropping packets. The system will be too busy to generate real alerts, and in many instances the IDS will simply stop responding.

If this were a real attack going on during the 450 spoofed attacks, which one would stand out as real, how would an analyst begin to wade through the data to search for the real offender?

Attack mechanism:

The attack here is a stimulus target at the Intrusion Detection System hoping to get a response of null activity from the system.
This attack is a very real threat at IDS systems since the IDS location does not have to be known, and an attacker can operate in near secrecy while running this attack.

This attack could be listed as a Denial of Service aimed at the Intrusion Detection System.

Correlations:

Information can be found on stick at the following locations:

Stick Home page:
http://packetstorm.securify.com/distributed/stick.htm

Packetstorm:
As Listed in the "100 most recent additions to Packet Storm."
"http://packetstorm.securify.com/whatsnew100.html
Stick is a distributed denial of service attack which targets IDS systems. It takes a snort rule file as input. Binary distribution.
Homepage: By Coretez Giovanni
Stick can be found at: http://packetstorm.securify.com/distributed/stick.tgz"

Securiteam:
http://www.securiteam.com/securitynews/_Stick_-_A_New_Denial_of_Service_Against_IDS_Systems.html

New Order:
http://neworder.box.sk/showme.php3?id=4264

A search of the SANS web page did not show any related links to the "stick" attack.

Active targeting:

This would definitely fall under active targeting; the attacker here is attempting to blind the Intrusion Detection staff.

Severity:

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

Criticality:	5	Home network IDS is most likely the target of the attack.
Lethality:	4	Currently very lethal to IDS systems. However since Root access cannot be gained with this attack, I only rated it as a four.
Sys Counters:	4	Very difficult to defend against in this case (spoofed packets with known "bad" signatures), and the IDS is current, however, there are known vulnerabilities with this attack, so even well patched IDS systems are vulnerable.
Net Counters:	2	Firewall not able to block against such a large number of spoofed addresses; this attack falls into a Denial of Service range.

Severity:	3	

Defensive Recommendations:

Patches are supposedly being made for various Intrusion Detection Systems, however a more immediate fix would be to write a script or cron job that would check your IDS every few minutes and restart it if the daemon was not running.

Multiple Choice:

How does a Denial of Service being aimed at your IDS affect your analysis of ongoing data?

a) Data will be saved and analyzed after the DOS attack has subsided.
b) The DOS attack can be blocked at the firewall.
c) Analyzing data becomes very difficult.
d) A second IDS will pick up dropped packets.

Answer:
The correct answer is C.

[This page intently left blank.]

Assignment 2
Remote OS Detection using TCP/IP and other Protocols

Matthew Waddell

"If your goal is to understand your network from a 40,000-foot view, then Windows port scanning tools will suffice. But if you're serious about your security and looking for the holes that crackers will find, then take the time to install a Linux box and use nmap."
-- Info World - July 6, 1998 --

Hackers and Administrators alike both require tools that can probe a network to look for responsive machines, open ports, and network configurations. Determining what Operating System is running at a certain IP address can be helpful to determine what needs to be patched, and in general what services should and shouldn't be running. Many times hacker's use these same tools to determine what Operating System is running or what version of a service is running in order that they may apply the correct tools or scripts used to infiltrate the system.

Banner Checking

The easiest way to check an OS is to connect to the client and read the banner that is commonly displayed at login. Most Operating Systems ship with this banner active, and many administrators do not change or turn the banners off.
However, some administrators have been known to "Lie" in the banner. This can be effective because a hacker could potentially spend days attempting to hack into an older version of a service when the current service is a well patched and protected one.
However, even after turning a banner off, many applications will give away information if queried correctly. For example with FTP the SYST command will feed you additional information about the computer that you are logged into.

Passive Fingerprinting

Normally OS Fingerprinting has been performed by using pre-made tools. Many of these tools are readily available on the Internet and a very few are not referenced in hacker documentation. Nmap and Queso are two of the best know tools that operate on the principle that an Operating System has it's own subtle idiosyncrasies when communicating on a network. Specifically, each OS will respond to packets that are sent to it differently. All that is left to do is to build a database of the ways that Operating Systems respond to certain transmissions, and then test a system against this known response database. The trouble with this approach is that if the host is monitoring incoming scans and connection attempts with some sort of Intrusion Detection System, they will see and record any reconnaissance attempt.

Passive fingerprinting confronts this problem differently. Instead of actively querying a remote system, all you need to do is to capture data packets that are sent from that remote system. Based on a sniffer trace of these packets you can determine what Operating System the remote host is running.

This system of analyzing is not exactly 100% accurate, and it seems to work better for some operating systems than it does for others. Also, no single signature can reliably determine the remote operating system. However, by looking at several signatures and combining the information, you can increase your chances of identifying the remote host. Also, applications that build their own packets (such as nmap, hunt, teardrop, etc) will not use the same signatures as the operating system that the tool is run on.

Type Of Service Reserved Bit

Every IP Datagram has an 8-bit Type-Of-Service field that helps prioritize the IP Datagram traveling over the Internet. This TOS field has three fields:
The Precedence Field, a Type-Of-Service Field, and the (MBZ) Must-Be-Zero Field.
RFC 1349 states that the MBZ field is unused and must contain a zero. The RFC also states that routers and hosts should ignore this bit.

Using only ICMP Echo requests, we can change this bit and get some idea about the OS that a Box is running. When the TOS bit is set, and an ICMP Echo Request is sent out, most Operating Systems will echo back the unused bit. However, Microsoft Windows 2000 Professional will respond with a zero set in this bit.
When using ICMP Query messages other than the ICMP Echo Request, Microsoft Windows versions 95, 98se, and ME will also zero out this field.
Further identification can be done when querying with an ICMP Address Mask request.
Only Windows 98 and 98se will respond, Windows ME will not reply, thus enabling us to identify it.

TTL

When comparing the information that you have gathered to a database of signatures you can also use the Time To Live that was used by the remote host.

The following table shows the normal default Time to Live for TCP and UDP packets.

OS Version	"safe"	tcp_ttl	udp_ttl
AIX	n	60	30
DEC Pathworks V5	n	30	30
FreeBSD 2.1R	y	64	64
HP/UX 9.0x	n	30	30
HP/UX 10.01	y	64	64
Irix 5.3	y	60	60
Irix 6.x	y	60	60
Linux	y	64	64
MacOS/MacTCP 2.0.x	y	60	60
OS/2 TCP/IP 3.0	y	64	64
OSF/1 V3.2A	n	60	30
Solaris 2.x	y	255	255
SunOS 4.1.3/4.1.4	y	60	60
Ultrix V4.1/V4.2A	n	60	30
VMS/Multinet	y	64	64
VMS/TCPware	y	60	64
VMS/Wollongong 1.1.1.1	n	128	30
VMS/UCX (latest rel.)	y	128	128
MS WfW	n	32	32
MS Windows 95	n	32	32
MS Windows NT 3.51	n	32	32

MS Windows NT 4.0	y	128	128

However TTL can be changed fairly easily on a machine to fool passive fingerprinting, for example:

For a Solaris system type:
ndd -set /dev/ip ip_def_ttl 'number'

On a Linux box type the command:
echo 'number' > /proc/sys/net/ipv4/ip_default_ttl

On a Windows NT machine edit the registry key:
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters

Don’t Fragment bit

Most systems have the DF bit set, so this is of limited value. However, a few systems do not use the DF flag (such as SCO and OpenBSD).
When analyzing data packets, every little bit counts... (I couldn’t resist.)

Window Size

Window Size’s are another effective tool, specifically what Window Size is used and how often the size changes. Hexadecimal 0x7D78 (32120) is a default Window Size commonly used by Linux. Linux, FreeBSD, and Solaris all tend to maintain the same Window Size throughout a session. However, Cisco routers and Microsoft Windows/NT Window Sizes are constantly changing as data fills and is released out of the memory window. A test of the Window Size is more accurate if it is measured after the initial three-way handshake (due to a slow start with TCP).

Overall we are not limited to just the four signatures listed here. There are other values that can be tracked and used to categorize OS types such as Initial sequence numbers, IP Identification numbers, and IP options. Databases also exist, which list ways to passively fingerprint systems with only a sample of the traffic.

Tools

Here is a collection of a few tools that have the ability to test for Operating Systems while querying a network.

Nmap

Nmap is perhaps one of the most well documented remote host discovery tool in existence. Nmap users have translated the nmap man page into Spanish, French, Russian, Italian, Portuguese, and Lithuanian. This shows that nmap has considerable support and is used worldwide.

A few scripts exist that can be used to harden a Microsoft Window's box so that the current version of nmap can not easily discover the OS that you are running. These scripts can be found at: http://neworder.box.sk/showme.php3?id=1898
Nmap can be found at: http://www.insecure.org/nmap/

Hping2

"Hping is a command-line oriented TCP/IP packet assembler/analyzer. The interface is very similar to the ping Unix command, with many extensions. It supports TCP, UDP, ICMP and RAW-IP protocols." http://www.kyuzz.org/antirez/hping.html

Cheops

Cheops is quoted as being a "network swiss army knife."
It has a combination of tools that allows system administrators (or hackers) to manage and access the network. One of Cheops features is that it includes OS detection by using invalid flags on TCP packets. http://www.marko.net/cheops/

Queso

Queso is one of the newest and most organized of the scanners. One of the most advanced features that queso has is that the builders moved the OS fingerprints out of the code. Instead of issuing a new version every time a new OS fingerprint is discovered, you can simply update the configuration file to keep it up to date and current. This allows queso to scale better and also makes it easy for a user to append a few lines as OS fingerprint discovery increases.
http://www.apostols.org

References:

http://www.insecure.org/nmap/press/infoworld-windows_scanners.txt

http://neworder.box.sk/showme.php3?id=1898

http://neworder.box.sk/showme.php3?id=3448

http://www.insecure.org/nmap/nmap-fingerprinting-article.html

http://newdata.box.sk/2000a/papers/finger.html

http://www.switch.ch/docs/ttl_default.html

http://www.kyuzz.org/antirez/hping.html

http://www.marko.net/cheops/

[This page intently left blank.]

Assignment 3

Matthew Waddell Security Services

This is a bid to provide security services to GIAC Enterprises an e-business startup that sells electronic fortune cookie sayings. Your organization has provided us with one months worth of snort data to analyze so as to provide a detailed analysis of what security services GIAC Enterprises may need.

The month's data was provided in four separate files: alert.zip, scan1.zip, scan2.zip, and OOS.zip.
These zip files were a collection of a months worth of snort data and snort alerts.
Here is a file breakdown:

- The Alert files contained the detailed Snort alert files.
- The Scans files contained the Snort scan detection preprocessor output files.
- The OOS files contained what would normally be in Snort's Log files.
  (This contained the same information as the alerts, but included more header information as well as the data payload.)

After downloading these files and unzipping them on my machine, I searched through the files for similar days of data, which I deleted one of each instance of in order to reduce the data load.

Next I concatenated all of the similar data files using the following commands:

```
cat SnortA* >  alert.txt.1
cat SnortS*  > scan.txt.1
cat OOS*    >  OOS.txt.1
```

These files were originally on a DOS (Windows) machine so they had carriage returns (^M) at the end of every line.
To remove the carriage returns I ran this command for each file:

```
dos2unix   oldfile.txt.1   newfile.txt
```

I then removed the alert.txt.1, scan.txt.1, and OOS.txt.1 files.

After downloading and playing with the program SnortSnarf (http://www.silicondefense.com/software/snortsnarf/) I found that the program does not like working with files that have IP addresses that resolve into letters (ex. MY.NET.123.456)
So I had to remove the "MY.NET." and change it into something numeric.
(This was later changed back to MY.NET for the paper.)
I chose to change MY.NET into 192.168.

To change this throughout the three files, I generated the following commands using perl:

```
perl -pi -e "s/MY.NET/192.168/g" alert.txt
perl -pi -e "s/MY.NET/192.168/g" scan.txt
perl -pi -e "s/MY.NET/192.168/g" OOS.txt
```

(The program "snortsnarf.pl" had a problem running due to not being able to find the perl script.
On this particular Sparc 10 perl is located at:
/usr/local/bin/perl
The first line in "snortsnarf.pl" had to be changed to reflect this.)

The SnortSnarf generated page led me to the following analysis:

The actual data that is analyzed falls in this time frame:

11/24-00:09:51.110204 - 11/29-23:17:50.134801
12/01-00:43:32.011356 - 12/31-23:45:47.026613
01/01-00:00:46.876474 - 01/18-23:50:31.755872

SnortSnarf recorded the First Detect as being recorded on: 01/01-00:00:46.876474
with the Last Detect being recorded on: 12/31-23:45:47.026613.
[The period of data that falls from 11/24-00:09:51.110204 through 11/29-23:17:50.134801 was allowed to remain so that a larger sampling of data could be recorded. This would also take anomalies into account such as end of month backups and once a month cron jobs. This would also complete the full months data. Making up for the gap between 01/18 and 12/01.]

Here is a summary of all of the possible attacks directed against your network.

1	instance of SITE EXEC - Possible wu-ftpd exploit - GIAC000623
1	instance of STATDX UDP attack
1	instance of Happy 99 Virus
2	instances of site exec - Possible wu-ftpd exploit - GIAC000623
8	instances of Probable NMAP fingerprint attempt
59	instances of External RPC call
77	instances of Back Orifice
100	instances of TCP SMTP Source Port traffic
154	instances of Broadcast Ping to subnet 70
159	instances of connect to 515 from inside
204	instances of SUNRPC highport access!
515	instances of SMB Name Wildcard
546	instances of Russia Dynamo - SANS Flash 28-jul-00
558	instances of NMAP TCP ping!
591	instances of SNMP public access
710	instances of Queso fingerprint
826	instances of Null scan!
2053	instances of Attempted Sun RPC high port access
2239	instances of WinGate 1080 Attempt
2401	instances of Watchlist 000222 NET-NCFC
4238	instances of connect to 515 from outside
5340	instances of Tiny Fragments - Possible Hostile Activity
16146	instances of DNS udp DoS attack described on unisog
51192	instances of SYN-FIN scan!
105918	instances of Watchlist 000220 IL-ISDNNET-990517

(Also see: Appendix 1 Monthly Overview)

1	instance of	SITE EXEC - Possible wu-ftpd exploit - GIAC000623
1	instance of	STATDX UDP attack
1	instance of	Happy 99 Virus
2	instances of	site exec - Possible wu-ftpd exploit - GIAC000623
8	instances of	Probable NMAP fingerprint attempt
59	instances of	External RPC call
77	instances of	Back Orifice
100	instances of	TCP SMTP Source Port traffic
154	instances of	Broadcast Ping to subnet 70
159	instances of	connect to 515 from inside
204	instances of	SUNRPC highport access!
515	instances of	SMB Name Wildcard
546	instances of	Russia Dynamo - SANS Flash 28-jul-00
558	instances of	NMAP TCP ping!
591	instances of	SNMP public access
710	instances of	Queso fingerprint
826	instances of	Null scan!
2053	instances of	Attempted Sun RPC high port access
2239	instances of	WinGate 1080 Attempt
2401	instances of	Watchlist 000222 NET-NCFC
4238	instances of	connect to 515 from outside
5340	instances of	Tiny Fragments - Possible Hostile Activity
16146	instances of	DNS udp DoS attack described on unisog
51192	instances of	SYN-FIN scan!
105918	instances of	Watchlist 000220 IL-ISDNNET-990517

Here is a brief summary of the Data including some of your Heaviest Hitters.

One of your networks Heaviest Hitters was the “Watchlist 000220 IL-ISDNNET-990517” with 105918 instances of possible attacks.

These attacks came from Israel and went to several different machines. The activity accessed different ports, including ports:

Port 443 (http protocol over TLS/SSL)
Port 1779 (pharmasoft)
Port 143 (imap)
Port 25 (smtp)
And Port 23 (Telnet.)

If you are conducting legitimate activity with a network in Israel, then you may want to set up some inclusions in your Snort filters to allow these IP Addresses through without generating alerts.

An intensive SYN-FIN scan was run against your network on several occasions.
These scans were directed at the following ports:

53	- DNS
21	- FTP
259	- esro-gen
146	- iso-tp0
109	- pop2
1995	- cisco perf port

Several of these scans lasted almost exactly 22 minutes:

```
01/07-03:47:11.227520 [**] SYN-FIN scan! [**] 211.34.40.1:53 -> MY.NET.1.1:53
-Through-
01/07-04:08:46.567612 [**] SYN-FIN scan! [**] 211.34.40.1:53 -> MY.NET.254.254:53
```

211.34.40.1 is Korea Network Information Center scanning for DNS.

```
01/10-12:00:54.263825 [**] SYN-FIN scan! [**]  195.56.182.206:21 -> MY.NET.1.2:21
-Through-
01/10-12:22:29.618261 [**] SYN-FIN scan! [**] 195.56.182.206:21 -> MY.NET.254.248:21
```

195.56.182.206 is DataNet Telecommunication Ltd. In Hungary scanning for FTP.

```
11/28-20:02:46.608820 [**] SYN-FIN scan! [**] 139.130.61.206:109 -> MY.NET.1.4:109
-Through-
11/28-20:24:20.985397 [**] SYN-FIN scan! [**] 139.130.61.206:109 -> MY.NET.254.231:109
```

139.130.61.206 is Telstra Corporation Limited in Australia scanning for pop2.

It is very likely that these scanners have a detailed map of your network.

On January 6th between 18:30 and 20:00 your DNS was under a “udp DoS attack” which appeared to originate from Exodus Communications at IP Address 209.67.50.203.

Here is the first alert:

```
01/06-18:30:02.600073 [**] DNS udp DoS attack described on unisog [**] 209.67.50.203:9247 -> MY.NET.1.3:53
```

And the last alert:

```
01/06-20:00:01.567114 [**] DNS udp DoS attack described on unisog [**] 209.67.50.203:23516 -> MY.NET.1.3:53
```

Your machines MY.NET.1.10 and MY.NET.1.8 continually communicate with Tiny Fragments to a variety of different hosts.

Some of these hosts names resolve to such addresses as:

61.155.13.3	CHINANET Jiangsu province network (China)
210.12.160.130	Jitong Communications Co.,Ltd (also in China)
202.101.43.220	Shanghai Long Distance Telecom (again, China)
65.4.87.43	@Home Network (Cable Modems)

It is possible that these machines have been infected with some sort of Trojan or backdoor type of program, further analysis of the machines would be necessary in order to make a proper analysis.

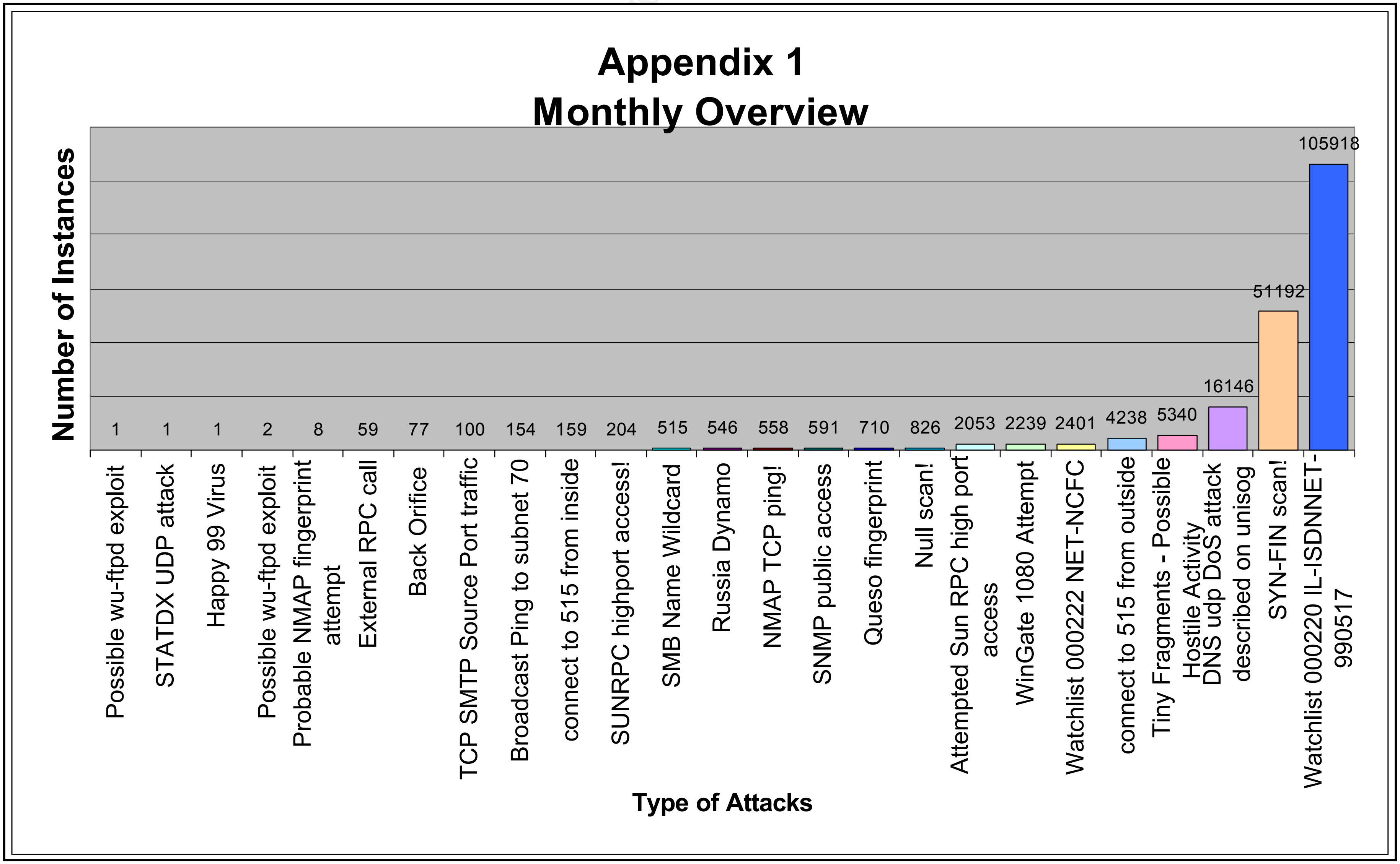
Defensive recommendations:

Your firewall should be set up so that the following incidents can be controlled:

External RPC calls from outside your network.
TCP SMTP Source Port traffic should be blocked at the firewall.
SUNRPC highport access should be denied.
The attempts at Sun RPC high port access should be blocked.
WinGate settings should be tightened.
Connections to port 515 from outside your network should not be allowed.
Tiny Fragments that could indicate possible Hostile Activity should be blocked.

Immediate recommendations:

Block access to port 515 through your firewall
Check hosts MY.NET.1.10 and MY.NET.1.8 for possible compromises.


Appendix 1
Monthly Overview
Number of Instances
Type of Attacks
1
1
1
2
8
59
77
100
154
159
204
515
546
558
591
710
826
2053
2239
2401
4238
5340
16146
51192
105918
Possible wu-ftpd exploit
STATDX UDP attack
Happy 99 Virus
Possible wu-ftpd exploit
Probable NMAP fingerprint attempt
External RPC call
Back Orifice
TCP SMTP Source Port traffic
Broadcast Ping to subnet 70
connect to 515 from inside
SUNRPC highport access!
SMB Name Wildcard
Russia Dynamo
NMAP TCP ping!
SNMP public access
Queso fingerprint
Null scan!
Attempted Sun RPC high port access
WinGate 1080 Attempt
Watchlist 000222 NET-NCFC
connect to 515 from outside
Tiny Fragments - Possible Hostile Activity
DNS udp DoS attack described on unisog
SYN-FIN scan!
Watchlist 000220 IL-ISDNNET-990517