Global Information Assurance Certification Paper

Intrusion Detection In Depth

GCIA
Practical Assignment v2.9

James Manion

July 23, 2001

Assignment #1

Below is a list of 10 detects that were obtained from my home computer dialed into the Internet through a local Internet Service Provider. Windump was used to capture the data and Snort was used to analyze and interpret the packets logged.

Detect #1 (DNS Scan)

07/10-05:49:51.230426 216.3.81.131:53 -> 63.49.116.112:53
TCP TTL:30 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x6CB12E90 Ack: 0x1D895C2 Win: 0x404 TcpLen: 20

1. Source of Trace.

My Network

2. Detect was generated by:

Detect was generated by using Windump Version 3.5.2a to capture the raw packets. Then Snort Version 1.7 was used to read the tcpdump file.
The Windump command used was: windump -w d:\james.log -B 10000 –vvv

-w Writes the raw packets to *file* rather than parsing and printing them out

-B Sets the driver's buffer size in Kilobytes. The default was 1 MB however to avoid any type of packet loss I bumped the buffer size to 10MB.

-vvv Produces an even more verbose output

The Snort command used was snort -r d:\james2.log -l c:\Inetpub\wwwroot\Logs

-r Reads in a tcdump generated file which was obtained through Windump.

-l Logs packets to a directory and sets up a hierarchical directory structure with the log directory as the base starting directory, and the IP address of the remote peer generating traffic as the directory which packets from that address are stored in.

Date/Time Group	07/10-05:49:51.230426
Source Address and Port	216.3.81.131:53
Direction Operator	->
Destination Address and Port	63.49.116.112:53
Protocol and Time to Live (TTL)	TCP TTL:30
Type of Service (TOS)	TOS:0x0
Packet ID Binary	ID:39426
IP header Length	IpLen:20
Total Length	DgmLen:40
TCP Flags Set	******SF
Sequence # in Hex	Seq: 0x6CB12E90
Acknowledgement # in Hex	Ack: 0x1D895C2
Window Size in Hex	Win: 0x404
TCP Header Length	TcpLen: 20

3. Probability the source address was spoofed:
Low. The source address is probably not spoofed. According to www.arin.net the address 216.3.81.131 is registered to the "Business Internet Inc" an ISP in Tampa Florida. Below is an excerpt from www.arin.net

Business Internet, Inc. (NET-ICIX-MD-BLK17)
3625 Queen Palm Drive
Tampa, FL 33619
US

Netname: ICIX-MD-BLK17
Netblock: 216.0.0.0 - 216.5.255.255
Maintainer: IMBI

Coordinator:
Business Internet, Inc. (ZI44-ARIN) ipreq@icix.net
240-616-2000

Domain System inverse mapping provided by:

NS.DIGEX.NET 64.245.20.14
NS2.DIGEX.NET 64.245.43.14

Record last updated on 02-Jan-2001.
Database last updated on 11-Aug-2001 23:03:50 EDT.

4. Description of attack:
IP address 216.3.81.131 scanned 63.49.116.112 using a SYN/FIN scan for services running on port 53. This port is reserved for Domain Name Services (DNS). This is probably a crafted packet since the SYN and FIN flags are both set.

5. Attack mechanism:
There was no attack mechanism since this was a simple reconnaissance attempt for DNS servers on 63.49.116.112

6. Correlations:
Since this was just a port scan there was no attack however log files for the networks DNS servers should be check to help match the time of the attack. If this is a low and slow recon attempt it is only a matter of time before they discover any DNS servers being used on the network. Once the DNS servers are discovered Zone Transfers between the servers can give out lots of critical IP address data for other servers on the network.

7. Evidence of active targeting:
The scanned machine was a Windows-based PC on a Dial-up connection to a local ISP. This was the only traffic from IP address 216.3.81.131 to IP address 63.49.116.112, which leads us to believe that the individual is scanning a block of IP addresses. Log files from the ISP's other machines should show other addresses being scanned.

8. Severity:

System Criticality + Attack Lethality
\- System Countermeasures + Network Countermeasures
= Severity

(5 + 1) - (5 + 2) = -1
System Criticality = 5 – DNS Server
Attack Lethality = 1 – Not Likely to Succeed
System Countermeasures = 5 – Modern Operating System and Service Not Running
Network Countermeasures = 2 – Firewall in place but traffic is allowed to target
Severity = -1

9. Defensive recommendation:

If scans persist adjust firewall to block traffic from this block of IP addresses. Firewall rules can also be established to block inbound traffic to destination Port 53.

10. Multiple choice test question:

07/10-05:49:51.230426 216.3.81.131:53 -> 63.49.116.112:53
TCP TTL:30 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
******SF Seq: 0x6CB12E90 Ack: 0x1D895C2 Win: 0x404 TcpLen: 20

Based on the above log file what field denotes the total size of the packet?

A. DgmLen
B. TcpLen
C. IpLen
D. Win

Answer: A

Detect #2 (IRC Scan)

07/10-02:45:42.816670 207.200.89.246:1028 -> 63.49.116.112:4440
TCP TTL:245 TOS:0x0 ID:54742 IpLen:20 DgmLen:576 DF
A* Seq: 0xD7EB1454 Ack: 0x17E321F Win: 0x83E8 TcpLen: 20

07/10-02:45:42.826394 63.49.116.112:4440 -> 207.200.89.246:1028
TCP TTL:128 TOS:0x0 ID:5280 IpLen:20 DgmLen:40
*****R** Seq: 0x17E321F Ack: 0x17E321F Win: 0x0 TcpLen: 20

1. Source of Trace.

My Network

2. Detect was generated by:
Detect was generated by using Windump Version 3.5.2a to capture the raw packets. Then Snort, Version 1.7 was used to read the tcpdump file.
The Windump command used was: windump -w d:\james.log -B 10000 –vvv

-w Writes the raw packets to *file* rather than parsing and printing them out
-B Sets the driver's buffer size in Kilobytes. The default was 1 MB however to avoid any type of packet loss I bumped the buffer size to 10MB.
-vvv Produces an even more verbose output

The Snort command used was snort -r d:\james2.log -l c:\Inetpub\wwwroot\Logs

-r Reads in a tcdump generated file which was obtained through Windump.
-l Logs packets to a directory and sets up a hierarchical directory structure with the log directory as the base starting directory, and the IP address of the remote peer generating traffic as the directory which packets from that address are stored in.

Date/Time Group	07/10-02:45:42.816670
Source Address and Port	207.200.89.246:1028
Direction Operator	->
Destination Address and Port	63.49.116.112:4440
Protocol and Time to Live (TTL)	TCP TTL:245
Type of Service (TOS)	TOS:0x0
Packet ID Binary	ID:54742
IP header Length	IpLen:20
Total Length	DgmLen:576
Don't Fragment Flag	DF
TCP Flags Set	***A****
Sequence # in Hex	Seq: 0xD7EB1454
Acknowledgement # in Hex	Ack: 0x17E321F
Window Size in Hex	Win: 0x83E8
TCP Header Length	TcpLen: 20

3. Probability the source address was spoofed:
Medium. The source address is probably spoofed. According to www.arin.net the address 207.200.89.246 is registered to Netscape Communications Corp, which is owned by American Online. Since this is an IRC scan what better way to spoof an IP than to try to hide among the 26 million other AOL users out there. Also the fact that the first incoming packet was an ACK packet points to a crafted packet.

Below is an excerpt from www.arin.net

Netscape Communications Corp. (NETBLK-NETSCAPE-CIDR)
501 East Middlefield Rd
Mountain View, CA 94043
US

Netname: NETSCAPE-CIDR
Netblock: 207.200.64.0 - 207.200.127.255
Maintainer: NSCP

Coordinator:
America Online, Inc. (AOL-NOC-ARIN) domains@AOL.NET
703-265-4670

Domain System inverse mapping provided by:

NS.NETSCAPE.COM 198.95.251.10
NS2.NETSCAPE.COM 207.200.73.80

ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

Record last updated on 28-Mar-2001.
Database last updated on 11-Aug-2001 23:03:50 EDT.

4. Description of attack:
IP address 207.200.89.246 scanned 63.49.116.112 for possible IRC connections running on port 4440. This port along with 6667 and 7000 are commonly used Internet Relay Chat (IRC) ports.

5. Attack mechanism:
There was no attack mechanism since this was a simple reconnaissance attempt for IRC connections on 63.49.116.112.

6. Correlations:
An Internet search turned up the following information on port 4440:
http://sol.pace.edu/java/ hosts an IRC server on port 4440
http://www.multimania.com/garmee/dhq/internet.html#IRC hosts an IRC server on ports 4440, 6667 and 7000

7. Evidence of active targeting:
The scanned machine was a Windows-based PC on a Dial-up connection to a local ISP. This was the only traffic from IP address 207.200.89.246 to IP address

63.49.116.112, which leads us to believe that the individual is scanning a block of IP addresses. Log files from the ISP's other machines should show other addresses being scanned.

8. Severity:

System Criticality + Attack Lethality
\- System Countermeasures + Network Countermeasures
= Severity

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

System Criticality = 1 – IRC Server
Attack Lethality = 1 – Not Likely to Succeed
System Countermeasures = 5 – Modern Operating System and Service Not Running
Network Countermeasures = 2 – Firewall in place but traffic is allowed to target
Severity = -5

9. Defensive recommendation:

This was only a reconnaissance attempt and if scans persist adjust firewall to block traffic from this block of IP addresses. Firewall rules can also be established to block inbound traffic to common IRC destinations like port 4440, 6667 and 7000.

10. Multiple choice test question:

The acronym IRC stands for

a. Internet Remote Control
b. Internet Relay Chat
c. Instant Response Connection
d. Intrusion Response Console

Answer: B

Detect #3 (SUN RPC Portmapper)

07/10-00:00:41.543321 65.163.40.247:1885 -> 63.49.116.112:111
TCP TTL:49 TOS:0x0 ID:63919 IpLen:20 DgmLen:60 DF
******S* Seq: 0xB51E52BF Ack: 0x0 Win: 0x7D78 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 4718795 0 NOP WS: 0

07/10-00:00:41.552692 63.49.116.112:111 -> 65.163.40.247:1885
TCP TTL:128 TOS:0x0 ID:54882 IpLen:20 DgmLen:40
***A*R** Seq: 0x0 Ack: 0xB51E52C0 Win: 0x0 TcpLen: 20

1. Source of Trace.

My Network

2. Detect was generated by:

Detect was generated by using Windump Verions 3.5.2a to capture the raw packets. Then Snort, Version 1.7 was used to read the tcpdump file.

The Windump command used was: windump -w d:\james.log -B 10000 –vvv

-w Writes the raw packets to *file* rather than parsing and printing them out

-B Sets the driver's buffer size in Kilobytes. The default was 1 MB however to avoid any type of packet loss I bumped the buffer size to 10MB.

-vvv Produces an even more verbose output

The Snort command used was snort -r d:\james2.log -l c:\Inetpub\wwwroot\Logs

-r Reads in a tcdump generated file which was obtained through Windump.

-l Logs packets to a directory and sets up a hierarchical directory structure with the log directory as the base starting directory, and the IP address of the remote peer generating traffic as the directory which packets from that address are stored in.

Date/Time Group	07/10-00:00:41.543321
Source Address and Port	65.163.40.247:1885
Direction Operator	->
Destination Address and Port	63.49.116.112:111
Protocol and Time to Live (TTL)	TCP TTL:49
Type of Service (TOS)	TOS:0x0
Packet ID Binary	ID:63919
IP header Length	IpLen:20
Total Length	DgmLen:60
Don't Fragment Flag	DF
TCP Flags Set	******S*
Sequence # in Hex	Seq: 0xB51E52BF
Acknowledgement # in Hex	Ack: 0x0
Window Size in Hex	Win: 0x7D78
TCP Header Length	TcpLen: 40

3. Probability the source address was spoofed:

Low. The source address is probably not spoofed. According to www.arin.net the address 65.163.40.247 is registered to Northgate Communications out of Norway, Minnesota. Below is an excerpt from www.arin.net

NORTHGATE COMMUNICATIONS (NETBLK-FON-110121164877474)
723 MAIN ST PO BOX 95
NORWAY, MI 49870
US

Netname: FON-110121164877474
Netblock: 65.163.40.0 - 65.163.40.255

Coordinator:

AUBREY, DON (DA708-ARIN) DAUBREY@NORTHGATEOCM.COM
(906)563-5729

4. Description of attack:
IP address 65.163.40.247 scanned 63.49.116.112 for possible SUN RPC connections. The source attempted to establish a TCP connection to an RPC Portmapper service by issuing an initial SYN connection. Since the victim was a PC running Windows98 the response to the SYN attempt was an ACK with a RST flag set.

5. Attack mechanism:
There was no attack mechanism since this was a simple reconnaissance attempt searching for any open Portmapper services on 63.49.116.112.

6. Correlations:
According to the Network ICE website "RPC (Remote Procedure Call) is a networking technology developed by Sun Microsystems. It is used on most UNIX machines, and is a popular way of building networked applications. (Almost no Windows computers run this form of RPC)…….. Scanning for RPC is the first stage in looking for those particular programs. If you had been running RPC on your system, then the next step the intruder would take would be an RPC portmapper dump, which would list all the RPC programs on your machine and tell the intruder if there are any he/she can exploit (use to break into your system)."
http://advice.networkice.com/advice/Intrusions/2003016/default.htm

7. Evidence of active targeting:
The scanned machine was a Windows-based PC on a Dial-up connection to a local ISP. This was the only traffic from IP address 65.163.40.247 to IP address 63.49.116.112, which leads us to believe that the individual is scanning a block of IP addresses. Log files from the ISPs other machines should show other addresses being scanned.

8. Severity:

$$\frac{\text{System Criticality} + \text{Attack Lethality}}{-\ \text{System Countermeasures} + \text{Network Countermeasures}} = \text{Severity}$$

(5 + 1) - (5 + 2) = -1
System Criticality = 5 – Sun RPC Portmapper
Attack Lethality = 1 – Not Likely to Succeed
System Countermeasures = 5 – Modern Operating System and Service Not Running
Network Countermeasures = 2 – Firewall in place but traffic is allowed to

target
Severity = -1

9. Defensive recommendation:
This was only a reconnaissance attempt and if scans persist adjust firewall to block traffic from this block of IP addresses. Firewall rules can also be established to block inbound traffic to Sun RPC Portmapper services like Port 111. It is also prudent to check any Unix boxes located on the network since this reconnaissance attempt was searching specifically for Sun RPC services.

10. Multiple choice test question:
What operating system commonly use RPCs

a. Windows
b. Macintosh
c. Unix
d. OS/2

Answer: C

Detect #4 (Intuit – False Positive)

```
07/08-20:05:53.363519 63.49.114.61:2999 -> 206.154.102.12:5282
TCP TTL:128 TOS:0x0 ID:33146 IpLen:20 DgmLen:48 DF
******S* Seq: 0xFAF4CF5  Ack: 0x0  Win: 0x2000  TcpLen: 28
TCP Options (4) => MSS: 536 NOP NOP SackOK

07/08-20:05:53.549216 206.154.102.12:5282 -> 63.49.114.61:2999
TCP TTL:239 TOS:0x0 ID:33809 IpLen:20 DgmLen:44 DF
***A**S* Seq: 0xD59AE042  Ack: 0xFAF4CF6  Win: 0x2398  TcpLen: 24
TCP Options (1) => MSS: 536

07/08-20:05:53.553508 63.49.114.61:2999 -> 206.154.102.12:5282
TCP TTL:128 TOS:0x0 ID:33658 IpLen:20 DgmLen:40 DF
***A**** Seq: 0xFAF4CF6  Ack: 0xD59AE043  Win: 0x2180  TcpLen: 20

07/08-20:05:53.557522 63.49.114.61:2999 -> 206.154.102.12:5282
TCP TTL:128 TOS:0x0 ID:33914 IpLen:20 DgmLen:217 DF
***AP*** Seq: 0xFAF4CF6  Ack: 0xD59AE043  Win: 0x2180  TcpLen: 20

07/08-20:05:53.766919 206.154.102.12:5282 -> 63.49.114.61:2999
TCP TTL:239 TOS:0x0 ID:33810 IpLen:20 DgmLen:40 DF
***A**** Seq: 0xD59AE043  Ack: 0xFAF4DA7  Win: 0x2398  TcpLen: 20

07/08-20:05:53.773499 63.49.114.61:2999 -> 206.154.102.12:5282
TCP TTL:128 TOS:0x0 ID:34682 IpLen:20 DgmLen:538 DF
***AP*** Seq: 0xFAF4DA7  Ack: 0xD59AE043  Win: 0x2180  TcpLen: 20
```

07/08-20:05:54.071073 206.154.102.12:5282 -> 63.49.114.61:2999
TCP TTL:239 TOS:0x0 ID:33811 IpLen:20 DgmLen:225 DF
AP Seq: 0xD59AE043 Ack: 0xFAF4F99 Win: 0x2398 TcpLen: 20

07/08-20:05:54.103185 63.49.114.61:2999 -> 206.154.102.12:5282
TCP TTL:128 TOS:0x0 ID:35194 IpLen:20 DgmLen:217 DF
AP Seq: 0xFAF4F99 Ack: 0xD59AE0FC Win: 0x20C7 TcpLen: 20

07/08-20:05:54.107508 63.49.114.61:2999 -> 206.154.102.12:5282
TCP TTL:128 TOS:0x0 ID:35450 IpLen:20 DgmLen:576 DF
A* Seq: 0xFAF504A Ack: 0xD59AE0FC Win: 0x20C7 TcpLen: 20

07/08-20:05:54.113500 63.49.114.61:2999 -> 206.154.102.12:5282
TCP TTL:128 TOS:0x0 ID:35706 IpLen:20 DgmLen:65 DF
AP Seq: 0xFAF5262 Ack: 0xD59AE0FC Win: 0x20C7 TcpLen: 20

07/08-20:05:54.347021 206.154.102.12:5282 -> 63.49.114.61:2999
TCP TTL:239 TOS:0x0 ID:33812 IpLen:20 DgmLen:40 DF
A* Seq: 0xD59AE0FC Ack: 0xFAF504A Win: 0x2398 TcpLen: 20

07/08-20:05:54.377003 206.154.102.12:5282 -> 63.49.114.61:2999
TCP TTL:239 TOS:0x0 ID:33813 IpLen:20 DgmLen:40 DF
A* Seq: 0xD59AE0FC Ack: 0xFAF5262 Win: 0x2398 TcpLen: 20

07/08-20:05:54.417004 206.154.102.12:5282 -> 63.49.114.61:2999
TCP TTL:239 TOS:0x0 ID:33814 IpLen:20 DgmLen:40 DF
A* Seq: 0xD59AE0FC Ack: 0xFAF527B Win: 0x2398 TcpLen: 20

07/08-20:05:54.497098 206.154.102.12:5282 -> 63.49.114.61:2999
TCP TTL:239 TOS:0x0 ID:33815 IpLen:20 DgmLen:225 DF
AP Seq: 0xD59AE0FC Ack: 0xFAF527B Win: 0x2398 TcpLen: 20

07/08-20:05:54.672485 63.49.114.61:2999 -> 206.154.102.12:5282
TCP TTL:128 TOS:0x0 ID:35962 IpLen:20 DgmLen:40 DF
A* Seq: 0xFAF527B Ack: 0xD59AE1B5 Win: 0x200E TcpLen: 20

07/08-20:05:58.945327 63.49.114.61:2999 -> 206.154.102.12:5282
TCP TTL:128 TOS:0x0 ID:36218 IpLen:20 DgmLen:40 DF
AF Seq: 0xFAF527B Ack: 0xD59AE1B5 Win: 0x200E TcpLen: 20

07/08-20:05:59.117883 206.154.102.12:5282 -> 63.49.114.61:2999
TCP TTL:239 TOS:0x0 ID:33816 IpLen:20 DgmLen:40 DF
A* Seq: 0xD59AE1B5 Ack: 0xFAF527C Win: 0x2398 TcpLen: 20

07/08-20:06:01.047724 206.154.102.12:5282 -> 63.49.114.61:2999
TCP TTL:239 TOS:0x0 ID:33817 IpLen:20 DgmLen:40 DF

AF Seq: 0xD59AE1B5 Ack: 0xFAF527C Win: 0x2398 TcpLen: 20

07/08-20:06:01.054256 63.49.114.61:2999 -> 206.154.102.12:5282
TCP TTL:128 TOS:0x0 ID:36474 IpLen:20 DgmLen:40 DF
A* Seq: 0xFAF527C Ack: 0xD59AE1B6 Win: 0x200E TcpLen: 20

1. Source of Trace.
My Network

2. Detect was generated by:
Detect was generated by using Windump Version 3.5.2a to capture the raw packets. Then Snort, Version 1.7 was used to read the tcpdump file.
The Windump command used was: windump -w d:\james.log -B 10000 –vvv

-w Writes the raw packets to *file* rather than parsing and printing them out
-B Sets the driver's buffer size in Kilobytes. The default was 1 MB however to avoid any type of packet loss I bumped the buffer size to 10MB.
-vvv Produces an even more verbose output

The Snort command used was snort -r d:\james2.log -l c:\Inetpub\wwwroot\Logs

-r Reads in a tcdump generated file which was obtained through Windump.
-l Logs packets to a directory and sets up a hierarchical directory structure with the log directory as the base starting directory, and the IP address of the remote peer generating traffic as the directory which packets from that address are stored in.

Date/Time Group	07/08-20:05:53.363519
Source Address and Port	63.49.114.61:2999
Direction Operator	->
Destination Address and Port	206.154.102.12:5282
Protocol and Time to Live (TTL)	TCP TTL:128
Type of Service (TOS)	TOS:0x0
Packet ID Binary	ID:33146
IP header Length	IpLen:20
Total Length	DgmLen:48
Don't Fragment Flag	DF
TCP Flags Set	******S*
Sequence # in Hex	Seq: 0xFAF4CF5
Acknowledgement # in Hex	Ack: 0x0
Window Size in Hex	Win: 0x2000
TCP Header Length	TcpLen: 28

3. Probability the source address was spoofed:

Medium. The source address is probably not spoofed. According to www.arin.net the address 206.154.102.12 is registered to Intuit the makers of Quicken and Turbo Tax. Since this is a normal TCP Handshake connection and a transfer of data and the fact that Quicken is used on this machine leads me to believe that this is not a crafted packet. However, I placed the level of probability at Medium because caution should be used here. This is a perfect address to spoof if you are going to try to obtain a user's financial data. Below is an excerpt from www.arin.net

INTUIT (NETBLK-CW-206-154-102)
1870 ENABARCADERO
MOUNTAIN VIEW, CA 94039
US
Netname: CW-206-154-102
Netblock: 206.154.102.0 - 206.154.102.255

Coordinator:
Yap, Kelvin (KY139-ARIN) kelvin_yap@INTUIT.COM
650-944-6000 (FAX) (650)919-0655

Record last updated on 07-Jul-1997.
Database last updated on 11-Aug-2001 23:03:50 EDT.

4. Description of attack:
IP address 63.49.114.61 established an outbound connection with 206.154.102.12. The Three-Way handshake can be easily seen in the first three entries in the log where the packets start as a SYN then followed by a SYN-ACK and finally an ACK. From there you can see the ACK-PUSH flags denoting the transfer of data. The connection initiated by the host was some type of service or software installed on the PC. You can see that the first and majority of PUSH packets were outbound packets.

5. Attack mechanism:
There was no attack mechanism since this was an outbound connection made to Intuit's server

6. Correlations:
After checking though the files of quicken I found two in particular that stored this websites address and port.
C:\quickenw\inet\common\PATCH\Notify\index.ini, and
C:\quickenw\inet\common\QCHANNEL.DAT
I made backup of these files and then edited the live files however Quicken seems to rewrite these files whenever you start Quicken

7. Evidence of active targeting:

There was no targeting since this was an outbound connection.

8. Severity:

System Criticality + Attack Lethality
\- System Countermeasures + Network Countermeasures
= Severity

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

System Criticality = 5 – Confidential Financial Data
Attack Lethality = 1 – Not Likely to Succeed
System Countermeasures = 5 – Modern Operating System and Service Not Running
Network Countermeasures = 5 – Firewall in place must be outbound traffic
Severity = -4

9. Defensive recommendation:

Most likely the software is checking for updates. Reaffirm that the firewall will only allow outbound connections to be established to this particular site. It would also be wise to alert Quicken of this and perhaps they should release an update with an option to turn off this feature.

10. Multiple choice test question:

Which formula below is correctly written?

a. Dgmlen = IpLen + TcpLen
b. IpLen = DgmLen + TcpLen
c. DgmLen = TcpLen – IpLen
d. IpLen = TcpLen - DgmLen

Answer: A

Detect #5 (Senna Spy)

07/12-20:09:17.350043 164.106.71.90:14415 -> 63.49.136.175:13000
TCP TTL:114 TOS:0x0 ID:14615 IpLen:20 DgmLen:48 DF
******S* Seq: 0xEF2370 Ack: 0x0 Win: 0x4000 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

07/12-20:09:17.355151 63.49.136.175:13000 -> 164.106.71.90:14415
TCP TTL:128 TOS:0x0 ID:58788 IpLen:20 DgmLen:40
***A*R** Seq: 0x0 Ack: 0xEF2371 Win: 0x0 TcpLen: 20

1. Source of Trace.
My Network

2. Detect was generated by:
Detect was generated by using Windump Versions 3.5.2a to capture the raw packets then using Snort, Version 1.7 to read the tcpdump file.
The Windump command used was: windump -w d:\james.log -B 10000 –vvv

-w Writes the raw packets to *file* rather than parsing and printing them out
-B Sets the driver's buffer size in Kilobytes. The default was 1 MB however to avoid any type of packet loss I bumped the buffer size to 10MB.
-vvv Produces an even more verbose output

The Snort command used was snort -r d:\james2.log -l c:\Inetpub\wwwroot\Logs

-r Reads in a tcdump generated file which was obtained through Windump.
-l Logs packets to a directory and sets up a hierarchical directory structure with the log directory as the base starting directory, and the IP address of the remote peer generating traffic as the directory which packets from that address are stored in.

Date/Time Group	07/12-20:09:17.350043
Source Address and Port	164.106.71.90:14415
Direction Operator	->
Destination Address and Port	63.49.136.175:13000
Protocol and Time to Live (TTL)	TCP TTL:114
Type of Service (TOS)	TOS:0x0
Packet ID Binary	ID:14615
IP header Length	IpLen:20
Total Length	DgmLen:48
Don't Fragment Flag	DF
TCP Flags Set	******S*
Sequence # in Hex	Seq: 0xEF2370
Acknowledgement # in Hex	Ack: 0x0
Window Size in Hex	Win: 0x4000
TCP Header Length	TcpLen: 28

3. Probability the source address was spoofed:
Low. The source address is probably not spoofed. According to www.arin.net the address 164.106.71.90i s registered to the Virginia Community College

System. Since the community college system has open labs for students to use this address is probably one of their lab machines. Below is an excerpt from www.arin.net

Virginia Community College System (NET-VCCS)
101 North 14th Street
Richmond, VA 23219
US

Netname: VCCS
Netblock: 164.106.0.0 - 164.106.255.255

Coordinator:
Miller, Dennis (DM444-ARIN) dmiller@UT.CC.VA.US
(703) 323-4070 (FAX) (703) 323-3859

Domain System inverse mapping provided by:

NS1.CC.VA.US	164.106.1.1
NS2.CC.VA.US	164.106.2.1

Record last updated on 03-Mar-1999.
Database last updated on 11-Aug-2001 23:03:50 EDT.

4. Description of attack:
The packets logged were a scan of port 13000. The attacker was probably looking for an installed Trojan created by the SennaSpy Trojan Generator. A SYN packet was received but the attacker and an ACK/RST packet was sent back to the attacker.

5. Attack mechanism:
A Trojan generator that commonly uses Port 13000 is SennaSpy. This software allows the user to customize a Trojan through a few simple mouse clicks. The generator creates a Trojan that is "controlled by telnet" making it possible for just about any operating system to run the Trojan. The default port for SennaSpy is 11000 however port 13000 has become a popular port and really any port can be targeted due to the configurable capabilities of the SennaSpy Trojan generator.

Another feature of this Trojan is the ability to access the infected computer's file system with an ftp client such as cute ftp or Ws ftp , this aspect of SennaSpy is pretty scary because it gives the hacker power to download and upload any file of choice .

6. Correlations:
After searching many newsgroups and websites I was able to find the SennaSpy Trojan Generator at
http://www.megasecurity.org/trojans/sstrojangenerator/SSTG_all.html

7. Evidence of active targeting:

The scanned machine was a Windows-based PC on a Dial-up connection to a local ISP. This was the only traffic from IP address 164.106.71.90 to IP address 63.49.136.175, which leads us to believe that the individual is scanning a block of IP addresses. Log files from the ISPs other machines should show other addresses being scanned.

8. Severity:

System Criticality + Attack Lethality
\- System Countermeasures + Network Countermeasures
= Severity

(5 + 1) - (5 + 5) = -4
System Criticality = 5 – Confidential Financial Data
Attack Lethality = 1 – Not Likely to Succeed
System Countermeasures = 5 – Modern Operating System and Service Not Running
Network Countermeasures = 2 – Firewall in place but traffic is allowed to
target Severity = -1

9. Defensive recommendation:

The SennaSpy Trojan creates an entry in the registry.
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\
Also by using Netstat you can view open ports on your system. Since Telnet is used as the client you can try to telnet to the suspected ports from your own computer back to your own computer using the loop back address.
From the firewall point of view you can block all inbound telnet connection attempts that are destined for these ports (or all inbound telnet attempted sessions for that matter)

http://www.glocksoft.com/trojan_list/Senna_Spy_Trojan_Generator.htm
http://www.networkice.com/advice/Exploits/Ports/13000/default.htm

10. Multiple choice test question:

Which of the following was one of the first Trojan Generator programs, which made creating and customizing unique Trojans simple for the novice user?
a) Back Orrifice
b) SennaSpy
c) 7up
d) Copper

Answer B

Assignment #2
State of Intrusion Detection (Masters Paradise or Hacker's Paradise)

Background:
This Trojan was created to infect Windows 95/98/ME/NT. The original Trojan created in 1997 was known as Hackers Paradise however it was also know as SecretAgentDat2, Paradise Agent srv and finally as Masters Paradise. This relatively new type of Trojan was tailored to Remote Access and stealing passwords and operated on port 456

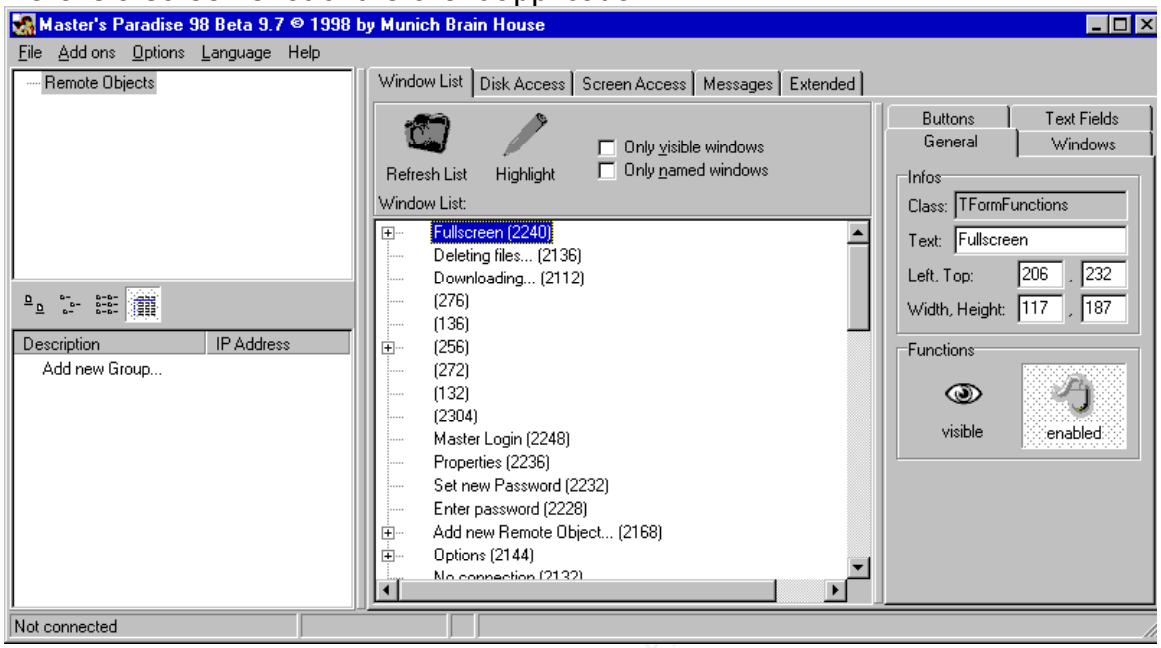
In March 1998, a German programmer decided to resurrect this Trojan and created Masters Paradise98. This mirrored many of the concepts and features of NetBus another Trojan that was widespread back in 1998. The new features in this version allowed the a file to be placed as a backdoor to gain access and control of the system as well as operate on different ports. The common ports where it was found were 31, 3129 40421, 40422, 40423, 40425, and 40426. Since it ran at a lower port of 31 there was even a version called Agent31 floating around the hacker underworld. Which of course led to a boot leg version called Agent40421 after the port that Masters Paradise ran on.

The key to getting this type of program to work was to have an executable installed on the victim's computer to permit remote access to the intended victim's system. Many other Trojans had this same concept following in the footsteps of the Cult of the Dead Cow's Back Orifice and NetBus. The Trojan executable used was propagated through AOL and IRC chat rooms. What made Masters Paradise different was that it was embedded in an actual game that was popular at the time. The game was called Pie Bill Gates, which was timely since Bill Gates had recently been ambushed by two pie wielding pranksters. There was also a Shockwave version of this popular game floating around in chain emails. So the unknowing victim runs the small game oblivious to the fact that they have just installed the Trojan on their machine.

The two files embedded in the "game" were SYSEDIT.exe and KeyHook.dll. SYSEDIT.exe is an actually windows program that is installed with the operating system. KeyHook.dll is also a popular dll that is used with many shareware programs.

How to Works:
This Trojan is triggered by a registry entry, which starts the application when the machine is booted. The startup key is found in [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run that is pointing to the trojan server (sysedit.exe), the subkey name is the name of the file.
Once the server side is started on the victims machine the Masters Paradise client (shown below) can remotely control the machine.

Here is a screen shot of the client application.



The client can control many different features on the remote machine including:

- Open/close the CD-ROM once or in intervals (specified in seconds).
- Show optional image. If no full path of the image is given it will look for it in the installed directory. The supported image-formats are BMP and JPG.
- Swap mouse buttons so that the right mouse button gets the left mouse button's functions and vice versa.
- Start optional applications.
- Play optional sound-files. If no full path of the sound-file is given it will look for it in the installed directory. The supported sound-format is WAV.
- Point the mouse to optional coordinates. You can even navigate the mouse on the target computer with your own!
- Send a message dialog to the victim's computer screen.
- Shutdown the system, logoff the user etc.
- Go to an optional URL within the default web-browser.
- Send keystrokes to the active application on the target computer! The text in the field Message/text will be inserted in the application that has focus.
- Listen for keystrokes and send them back to you!
- Get a screendump
- Return information about the target computer.
- Upload any file from you to the target computer.
- Increase and decrease the sound-volume.
- Record sounds from the computer's microphone.
- Make click sounds every time a key is pressed.

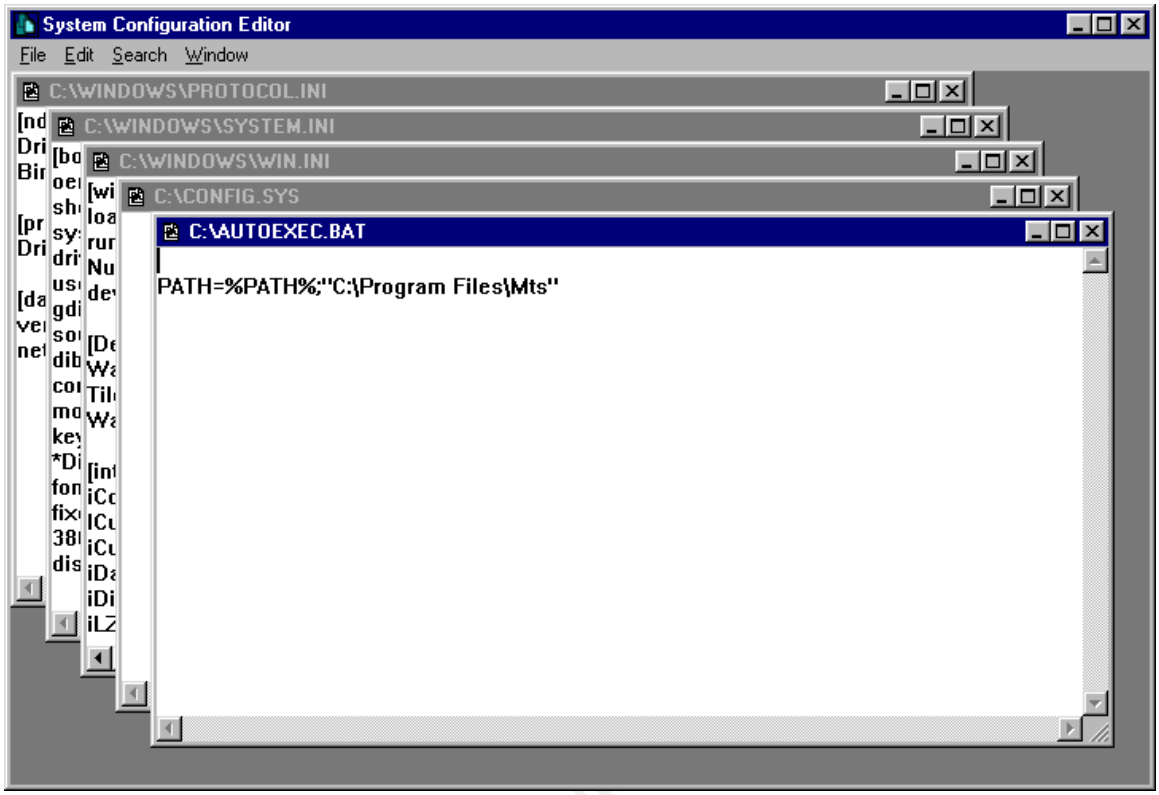
- Download and deletion of any file from the target.
- Keys (letters) on the keyboard can be disabled.
- Password-protection management.
- Show, kill and focus windows on the system.

Of all of these capabilities probably the most dangerous ones are the read/write/delete of files, which can give the hacker access to powerful information or even the ability to erase these files. Snagging keystrokes from the victim can gain insight into accounts and passwords accessed from the victim's machine. And even activating the victim's microphone can "bug" the user's office.

Detection:
So how do you know if you have the virus, aside from running a virus scanner? This server replaces the file C:\Windows\SYSEDIT.EXE and deletes the original. Normally the original is 10-18 Kb, but when you are infected it is over 100 Kb. Also, a file called KeyHook.dll is placed in either your C:\Windows directory or your C:\Windows\System folder. You can replace the original SYSEDIT-file from your Windows-setup disks. The file KeyHook.dll could also have existed before you had the Trojan, if so, reinstall the application using it.

Removal:
Since Masters Paradise used similar techniques that NetBus and Back Orifice many virus scanners have modified signatures to detect all three. As with most Trojans there are edits to the registry that are made by the software. This trojan can add up to 2 entries in the system registry to load on startup.

First Registry Key:
Delete the registry-key:
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run This is pointing to the Trojan server (sysedit.exe)
Keep in mind the real sysedit.exe does not run on startup. This is the program that is used to edit the system files like win.ini, config.sys, autoexec.bat, protocol.ini and system.ini. As note that SYSEDIT should NOT be running unless you manually ran it yourself.



- An updated version of the trojan also writes a key to : HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices Explorer = c:\.....\agent.exe, delete this key, too.
- Reboot your machine.
- Also, a file called KeyHook.dll is placed in either your windows or your windows/system-folder. You can replace the original SYSEDIT-file from your Windows-setup disks.
  The file KeyHook.dll could also have existed before you had the trojan, if so, reinstall the application using it.
- Reboot again.

Sources:

http://www.hackfix.org/miscfix/mp.shtml
http://www.dark-e.com/archive/trojans/mp/
http://www.fortunecity.com/skyscraper/techie/693/
http://home.tiscalinet.be/bchicken/trojans/masterpar/
http://www.simovits.com/trojans/tr_data/y565.html
http://www.simovits.com/trojans/tr_data/y776.html
http://www.multimania.com/cdc/master.html

Assignment # 3 Analyze This Scenario

A University has provided data from a Snort system using a standard rule set. The objective is to analyze the data and detect possible attempts to compromise the systems.

In order to get a good idea of the types of hacking attempts that the University sees on a daily basis I first created daily totals for the entire month of March 2001. This allowed me to gather trend data as well as try to detect any "low-n-slow" reconnaissance. The files used where from 03/01, 03/06, 03/12, 03/17 03/20/. The respective OOS and Snort alert files were also analyzed.

Executive Summary:
The intrusion detection system in place used a standard rule set with snort to log any alerts that passed through the network. There were over 478,903 alerts logged. The alerts varied between the tame port and IP scans to actual targeting of devices and using specific techniques like Operating System Fingerprinting as well as attempting to compromise specific services. There were even instances of spoofed IP addresses and crafted packets. Lastly there were several alerts that were from known hostile IP addresses that are on various "Watch-Lists". These specific addresses should be specified in any security policy

Here are the most common destination ports:

(Sorted by frequency)

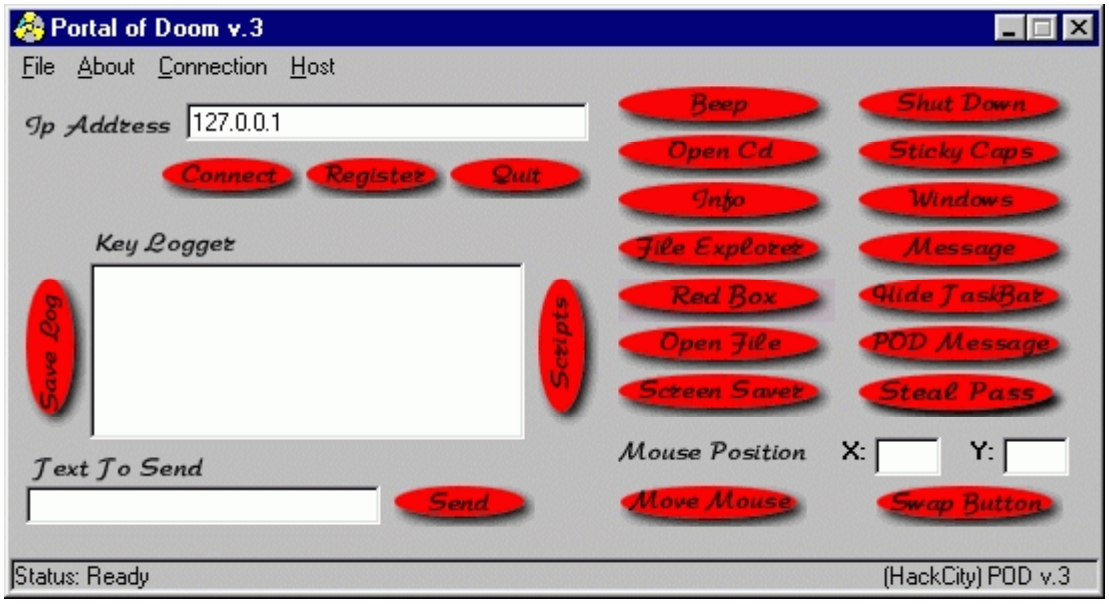
Port	#Hits
9875	283710
9880	140743
5779	69832
32771	7627
1718	7489
67	5223
1080	3414
316	3038
123	2551
137	2347
27374	1135
21	860
4971	638
6667	423
6346	368
38293	255
317	249

Port 9875 Portal of Doom

The Trojan Portal of Doom also known as BackDoor-K.srv, BackDoor-K.cli, and POD, tends to occupy the following ports: 3700, 9872, 9873, 9874, 9875, 10067 (UDP), 10167 (UDP). The Trojan was created in March 1999 and is currently at version 3.0. The Trojan is transmitted through a zip file like Pod.zip (277Kb), Portal of doombeta.zip (280Kb) or Portalofdoom3.0.zip (183Kb). These zipped files contain the files needed to install and activate the Trojan. The executable files needed are named Server.exe (114Kb), Portal.exe(502Kb), and Ljsgz.exe (111Kb) . The Windows controls (ocx files) are Cswsk32.ocx (90Kb), Comdlg32.ocx (140Kb).

The Trojan works very similar to other Remote Access Trojans where the client or Portal component can connect to a computer running the server component. Since Portal of Doom is written in Visual Basic it easily runs on Windows 9.x and Windows NT machines.

The Trojan creates a Registry entry that loads the Trojan when the machine is booted. This entry can be found at :
HKLM\Software\Microsoft\Windows\CurrentVersion\RunServices \
Here is a sample screen shot of the client/controlling application of Portal of Doom.



Port 9880 gtkglarea

This is a port of gtkglarea, a GTK extension that allows use of OpenGL stuff. This port uses libtool and is therefore broken out-of-the-box for 4.0-CURRENT. This can be fixed by pointing the configure script to /usr/local/share/libtool/* which has the One True FreeBSD Libtool so we don't need yet another copy of

the same patch. This port is also a common Portal of Doom port (see above) http://www.student.oulu.fi/~jlof/gtkglarea/

Port 32771 (Ghost Portmapper)

There are lots of attacks on rpc-based services towards solaris machines. RPC ports can be checked for rpc services with the command rpcinfo. On port 32771 you will usually find an RPC service called ypbind listening. According to NetworkIce, “Some SunOS machines listen at this port for portmapper. Since firewalls frequently don't filter out high ports, it can allow the attacker access to portmapper even when port 111 is blocked.” An attacker can try to access this port for operating system fingerprinting as well as network reconnaissance for Unix/Sun RPC services that may be running and a potential path for exploitation.

Port 27374 (Sub7)

Sub7 was written by a hacker who calls himself "Mobman". He maintains a site subseven.slak.org where one can download the Trojan and even enter a contest for sites that have been compromised by Sub7.According to NetworkIce, “this is one of the most commonly probed ports on the Internet right now, due to its inclusion within the Sub7 Trojan.” However, the use of Sub7 allows some customization and changing the port is one of these features. One of the reasons this Trojan is so popular is that it can actually have the victim computer scan for other devices on the network. Other reasons for its popularity is that it is actively maintained and updated which keeps it a step ahead of anti-virus definitions. The Trojan supports "port redirection", so that any attack can be funneled through a victim's machines. Also the Trojan has ways of manipulating Instant-messaging applications like ICQ, MSN, and AOL IM, on the user’s machine

This Trojan does not affect Linux or non-Windows machines however it can wreak havoc on Windows based computers.

The Trojan usually enters the victim’s computer through an email attachment disguised as an image. When the user “opens” the attachment the machine is infected and the Trojan is installed and executed. A common name for the file associated with Sub7 is BackDoor-G2.svr. There are two files that allow a client to contact and communicate with the Trojan. These files are BackDoor-G2.cfg and BackDoor-G2.cli. Since the victimized machine will scan for other devices the client files may be found on the compromised machine.

The Trojan makes several Registry changes including one that alters how .exe files are fun. It causes the Trojan to load when an EXE is launched including executables for changing the registry. Both Norton and McAfee offer data definitions to detect some instances of Sub7 as well as removal tools.

Here is a sample screen shot of the latest version of Sub7 version 2.2

Port 1080 Wingate

Port 1080 is usually used by either a socks server, or a web proxy. If an attacker can hit a SOCKS proxy server they can orchestrate other attacks that look as if they are originating from the SOCKS server. Any machine running a SOCKS proxy should not be considered a “trusted host”. Wingate is not always used to mask potential attackers. Wingate is also susceptible to Denial Of Service attacks in the form of a buffer overflow.

According to the SANS site “Attackers will often look for servers running the Wingate proxy software so that they can bounce HTTP traffic off of a proxy server to attack another web server. It can also be used as a socks proxy, meaning an attacker can telnet to a host and it will look like its coming from your proxy server”

Port 113 identd/auth

The ident daemon (identd) is used to identify the "owner" of a connection. This can reveal a lot of information to hackers. Based on RFC1413, the Identification Protocol is a “connection based application on TCP that listens for TCP connections on TCP port 113 (decimal).” This apparently innocuous port can cause problems with certain Linux and Unix operating systems. BugtraqID: 587 references an issue with the opening of a large number of connections to port 113 that can result in the downing of some SuSE Linux (CVE-1999-0746)

Vulnerable systems include SuSELinux 6.x, 5.x, 4.4.x and Slackware 3.6 and 3.2. These vulnerabilities are rooted in the fact that some versions start the Identification daemon with a “-w -t120” option. This causes the process to wait 120 seconds after answering the first request to answer the next request. Each subsequent request prior to the wait time expiring requires its own process to be started. If a malicious remote attacker starts a large number of ident requests in a short period of time it will force the target machine to start multiple daemons because the initial daemon is in a time wait state. In a short, this causes a Denial of Service when the resources on the machine begin to dwindle.

Tiny Fragments

The SANS site defines Tiny Fragments as “Fragmentation [that] occurs when packets are too large and need to be split up into smaller chunks of data.” Tiny Fragments occur when the chunks of data are smaller than normal. The term “normal” refers to the size of the chunks of data that is fragment by the operating system. According to SANS, “Earlier IDS systems used to miss these packets because they would only check the first fragmented packet and ignore the others. This would allow the attackers to slip by the Intrusion Detection System.” Some operating systems have trouble handling these Tiny Fragments and can crash. When Tiny Fragments are detected this is usually a sign of crafted packets that break up malicious code to avoid common IDS systems that do not reassemble the fragments before it inspects them.

Queso fingerprint

Fingerprinting is a term used when an attacker attempts to determine what type of operating system you are running. Queso is a tool facilitates this process. Carefully selected packets with different flags set in the TCP header can be sent to a machine and by reviewing the machine responses one can determine what operating system is running. Each operating system may respond differently to different packets so in this way the machine can be queried to fingerprint the system. Once the attacker knows the operating system of its intended target, system specific attacks and exploits can be used to compromise the system.

Here are the most common Source IP Addresses

(Number next to IP addresses denotes the number of occurrences)

130.225.127.87	96428
155.101.21.38	67958
194.165.226.27	35311
63.250.208.169	27592
206.190.54.67	23369
171.69.248.71	26883
128.223.83.33	14246
130.240.64.20	11879
140.142.19.72	11010
152.1.1.79	10708
130.161.180.141	10441
63.250.210.72	9421
206.190.54.131	9268
128.223.83.35	8554
172.137.152.251	7609
130.235.133.92	5157
128.178.10.2	5141
10.0.0.1	4733

Using the ARIN lookup here is the information on the common source IP addresses. All of the academic addresses are indicative of the typical arena for malicious traffic to originate. These places have students who are learning new computer techniques and want try them out on other hosts. These addresses are hard to trace back any further than the institution itself due to network policies. Most universities use DHCP or dynamic addressing of the student's computers in order to easily manage a large number of diverse clients. Many of these DHCP addresses have short leases and expire in a few days. Users can also release and renew their addresses manually. If an attack is conducted by the time the alert is analyzed and the packets traced back to the college the user can have already been assigned a new address. Also these institutions may have 24-hour computer labs that give the hacker-to-be a perfect place to initiate an anonymous attack.

130.225.127.87 (Academic Institution)

Danish Computer Centre for Research and Education (NET-DENET-1)
Building 305, DTH
DK-2800 Lyngby
DK
Netname: DENET-1
Netblock: 130.225.0.0 - 130.225.255.255
Coordinator:
Fjerdingstad, Torben (TF47-ARIN) uni-role@UNI-C.DK
+45 35 87 88 89
Domain System inverse mapping provided by:
NS-SOA.DARENET.DK 130.226.1.4
MIMER.SNET.UVM.DK 193.162.240.5

Record last updated on 01-Jul-1998.
Database last updated on 31-Jul-2001 23:07:46 EDT.

155.101.21.38 (Academic Institution)

University of Utah (NET-UTAH-OC-NET)
606 Black Hawk Way
Salt Lake City, UT 84108
US
Netname: UTAH-OC-NET
Netblock: 155.101.0.0 - 155.101.255.255
Coordinator:
University of Utah (ZU27-ARIN) postmaster@ns.utah.edu
+1 801 581 4000
Domain System inverse mapping provided by:
NS.UTAH.EDU 128.110.125.120
FIBER.UTAH.EDU 128.110.132.99
Record last updated on 09-Oct-2000.
Database last updated on 31-Jul-2001 23:07:46 EDT.

194.165.226.27 (Academic Institution)

European Regional Internet Registry/RIPE NCC (NETBLK-RIPE-C2)
These addresses have been further assigned to European users.
Contact info can be found in the RIPE database, via the
WHOIS and TELNET servers at whois.ripe.net, and at
http://www.ripe.net/db/whois.html
NL
Netname: RIPE-CBLK2
Netblock: 194.0.0.0 - 194.255.255.255
Maintainer: RIPE
Coordinator:
Reseaux IP European Network Co-ordination Centre Singel 258 (RIPE-NCC-ARIN) nicdb@RIPE.NET
+31 20 535 4444
Domain System inverse mapping provided by:
NS.RIPE.NET 193.0.0.193
NS.EU.NET 192.16.202.11
AUTH03.NS.UU.NET 198.6.1.83
NS2.NIC.FR 192.93.0.4
SUNIC.SUNET.SE 192.36.125.2
MUNNARI.OZ.AU 128.250.1.21
NS.APNIC.NET 203.37.255.97

% This is the RIPE Whois server.
% The objects are in RPSL format.
% Please visit http://www.ripe.net/rpsl for more information.
% Rights restricted by copyright.
% See http://www.ripe.net/ripencc/pub-services/db/copyright.html

inetnum: 194.165.226.0 - 194.165.226.255
netname: IT-CENTER
descr: IT-Center
descr: Umea
country: SE
admin-c: AL14-RIPE
tech-c: BJ12-RIPE
tech-c: EB78-RIPE
rev-srv: dns.it-center.se
rev-srv: dns2.it-center.se
rev-srv: dns.norrnod.se
status: ASSIGNED PA
mnt-by: UNO1-MNT
changed: Roland.Hedberg@umdac.umu.se 19960129
changed: hostmaster@ripe.net 19960209
changed: Roland.Hedberg@umdac.umu.se 19960313
changed: benny.jonsson@it-center.se 19990222
changed: benny.jonsson@it-center.se 19990301
changed: Bjorn.Isaksson@umdac.umu.se 19990419
source: RIPE

route: 194.165.224.0/19
descr: SE-UMDAC
descr: In case of improper use originating from our network,
descr: please mail customer or abuse@swip.net
origin: AS1257
notify: staff@swip.net
mnt-by: AS1257-MNT
changed: per@swip.net 20010327
source: RIPE

person: Anders Lindberg
address: Umea IT-Center AB
address: Box 150
address: 901 04 Umea
address: Sweden
phone: +46 90 100831
fax-no: +46 90 100819
e-mail: anders.lindberg@it-center.se
nic-hdl: AL14-RIPE
changed: Roland.Hedberg@umdac.umu.se 19960226
changed: erik.bylund@it-center.se 19990728
source: RIPE

person: Benny Jonsson
address: Umea IT-Center AB
address: Box 150
address: 901 04 Umea
address: Sweden
phone: +46 90 100832
fax-no: +46 90 100819
e-mail: benny.jonsson@it-center.se
nic-hdl: BJ12-RIPE
changed: Roland.Hedberg@umdac.umu.se 19960226
changed: erik.bylund@it-center.se 19990728
source: RIPE

person: Erik Bylund
address: Bredbandsbolaget
address: Box 148
address: S-901 04 Umea
address: Sweden
phone: +46 90 100830
fax-no: +46 90 100849
e-mail: erik.bylund@norr.bredband.com
nic-hdl: EB78-RIPE
changed: Roland.Hedberg@umdac.umu.se 19960321
changed: erik.bylund@it-center.se 19990728
changed: erik.bylund@norr.bredband.com 20010210
source: RIPE

63.250.208.169 (Large Internet Service Provider)

Yahoo! Broadcast Services, Inc. (NETBLK-NETBLK2-YAHOOBS)
2914 Taylor st
Dallas, TX 75226
US
Netname: NETBLK2-YAHOOBS
Netblock: 63.250.192.0 - 63.250.223.255
Maintainer: YAHO
Coordinator:
Bonin, Troy (TB501-ARIN) netops@broadcast.com
214.782.4278 ext. 2278
Domain System inverse mapping provided by:
NS.BROADCAST.COM 206.190.32.2
NS2.BROADCAST.COM 206.190.32.3
ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

Record last updated on 29-Jun-2001.
Database last updated on 31-Jul-2001 23:07:46 EDT.

206.190.54.67 (Large Internet Service Provider)

Yahoo! Broadcast Services, Inc. (NET-NETBLK1-YAHOOBS)
2914 Taylor St.
Dallas, TX 75226
US
Netname: NETBLK1-YAHOOBS
Netblock: 206.190.32.0 - 206.190.63.255
Maintainer: YAHO
Coordinator:
Bonin, Troy (TB501-ARIN) netops@broadcast.com
214.782.4278 ext. 2278
Domain System inverse mapping provided by:
NS.BROADCAST.COM 206.190.32.2
NS2.BROADCAST.COM 206.190.32.3
Record last updated on 29-Jun-2001.
Database last updated on 31-Jul-2001 23:07:46 EDT.

171.69.248.71

Bay Area Regional Research Network (NETBLK-BARRNET-BBLOCK)
3801 East Bayshore Road
Palo Alto, CA 94306
US
Netname: NETBLK-BARRNET-BBLOCK
Netblock: 171.68.0.0 - 171.71.255.255
Maintainer: BBNP
Coordinator:
BBN Network Operations Center (BNOC-ARIN) ops@BBNPLANET.COM
800-632-7638 617-873-8730 fax: 617-873-6315
ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
Record last updated on 29-Sep-2000.
Database last updated on 31-Jul-2001 23:07:46 EDT.

128.223.83.33 (Academic Institution)

University of Oregon (NET-UONET)
1225 Kincaid St
Eugene, OR 97403-1212
US
Netname: UONET
Netblock: 128.223.0.0 - 128.223.255.255
Coordinator:
Meyer, David M. (DMM65-ARIN) dmm@antc.uoregon.edu
541.915.0094
Domain System inverse mapping provided by:
PHLOEM.UOREGON.EDU 128.223.32.35
MAGGIE.TELCOM.ARIZONA.EDU 128.196.128.233
RUMINANT.UOREGON.EDU 128.223.21.15
DNS.CS.UOREGON.EDU 128.223.6.9
Record last updated on 27-Aug-1996.
Database last updated on 31-Jul-2001 23:07:46 EDT.

130.240.64.20 (Academic Institution)

University of Lulea (NET-LUTHNET)
Computer Centre
Lulea, 97187
SE
Netname: LUTHNET
Netblock: 130.240.0.0 - 130.240.255.255
Coordinator:
Lulea University of Technology (ZL34-ARIN) abuse@luth.se
+46 920 91262
Domain System inverse mapping provided by:
LUNIC.LUTH.SE 130.240.19.2
ERU.MT.LUTH.SE 130.240.1.1
Record last updated on 30-Aug-2000.
Database last updated on 31-Jul-2001 23:07:46 EDT.

140.142.19.72 (Academic Institution)

NorthWestNet Network Operations Center (NET-UW-SEA)
  Academic Computing Center
  3737 Brooklyn NE
  Seattle, WA 98105
  US
  Netname: UW-SEA
  Netblock: 140.142.0.0 - 140.142.255.255
  Maintainer: UWND
  Coordinator:
    University, Of Washington (OWU2-ARIN) noc@CAC.WASHINGTON.EDU
    206-543-5128
  Domain System inverse mapping provided by:
  HANNA.CAC.WASHINGTON.EDU 140.142.5.5
  MARGE.CAC.WASHINGTON.EDU 140.142.5.13
  NS.UNET.UMN.EDU 128.101.101.101
  Record last updated on 17-Mar-2000.
  Database last updated on 31-Jul-2001 23:07:46 EDT.

University, Of Washington (OWU2-ARIN) noc@CAC.WASHINGTON.EDU
  4545 15th Ave NE
  Seattle, WA 98105
  206-543-5128
  Record last updated on 14-Apr-1995.
  Database last updated on 31-Jul-2001 23:07:46 EDT.

152.1.1.79 (Academic Institution)

North Carolina State University (NET-NCSU)
  NCSU-Computing Center Box 7109
  Raleigh, NC 27695-7109
  US
  Netname: NCSU
  Netblock: 152.1.0.0 - 152.1.255.255
  Coordinator:
    Host, Master (HOS150-ORG-ARIN) Hostmaster@NCSU.EDU
    (919) 515-7571
Fax- (919) 513-1893
  Domain System inverse mapping provided by:
  UNI00NS.UNITY.NCSU.EDU152.1.1.22
  UNI10NS.UNITY.NCSU.EDU152.1.1.208
  Record last updated on 02-Sep-1998.
  Database last updated on 31-Jul-2001 23:07:46 EDT.

Here are the most common Destination IP Addresses

(Number next to IP addresses denotes the number of occurrences)

224.2.127.254	365297
233.28.65.255	27592
233.28.65.197	16058
233.40.70.194	9421
233.40.70.199	9268
224.0.1.41	8013
MY.NET.217.74	7609
10.255.255.255	4119
MY.NET.178.42	3289
224.0.1.1	1951
10.255.255.255	1387
204.62.32.194	711

169.254.255.255	704
MY.NET.222.2	636
192.168.0.255	585
MY.NET.144.54	528
63.162.10.74	375
MY.NET.98.110	127
MY.NET.98.123	123
128.252.25.206	97

Most of the destination IP addresses were multicast address or broadcast addresses (x.255.255.255, x.x.255.255, x.x.x.255 were broadcast and 224.x.x.x and 233.x.x.x are multicast). There were however a few that did turn up to be valid external addresses however many of these are from other Universities or colleges and the network administrator at these entities should be contacted and perhaps a correlation of logs files is in order

204.62.32.194 (Academic Institution)

Towson State University (NETBLK-TSU)
Academic Computing Service
Towson, MD 21204
US
Netname: TSU
Netblock: 204.62.32.0 - 204.62.51.255
Coordinator:
Houston, Samuel (SH1243-ARIN) shouston@BACH.TOWSON.EDU
(410) 830-4084 (FAX) (410) 830-2661
Domain System inverse mapping provided by:

HAL.TOWSON.EDU	204.62.32.10
TRANTOR.UMD.EDU	128.8.10.14

Record last updated on 07-May-1996.
Database last updated on 31-Jul-2001 23:07:46 EDT.

128.252.25.206 (Academic Institution)

Washington University (NET-WASHINGTON-U)
One Brookings Drive, Campus Box 1048
St. Louis, MO 63130
US
Netname: WASHINGTON-U
Netblock: 128.252.0.0 - 128.252.255.255
Coordinator:
Roman, John (JR756-ARIN) jrr@wustl.edu
(314) 935-4865 (FAX) (314) 935-7142
Domain System inverse mapping provided by:

WUGATE.WUSTL.EDU	128.252.120.1
WUARCHIVE.WUSTL.EDU	128.252.135.4
ADMIN.STARNET.NET	199.217.253.10
NEWS.STARNET.NET	199.217.253.11

Record last updated on 25-Apr-2000.
Database last updated on 31-Jul-2001 23:07:46 EDT.

Analysis of Out-Of Spec Files

These logs contained a large variety of packets which show signs of packet craft.

Anomalies were:

SF - SYN FIN flags set
03/12-02:47:04.671991 66.20.28.21:7982 -> MY.NET.201.98:4022
TCP TTL:112 TOS:0x0 ID:30594 DF
21SF**A* Seq: 0xE2295556 Ack: 0xC3022D Win: 0x5018
TCP Options => EOL EOL Opt 168
02 24 90 20 00 00 .$. ..

Source and Destination ports the same and set to 0
03/12-22:23:28.397042 66.9.62.2:0 -> MY.NET.98.249:0
TCP TTL:110 TOS:0x0 ID:41841 DF
*1SFRP** Seq: 0xFF8F Ack: 0xFF8FFF8F Win: 0xFF8F
TCP Options => Opt 236 (40): B690 8431 01D2 16C2 9650 DFFC 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
IP ID is static

TCP sequence number the same.
03/12-04:42:17.242319 209.85.251.2:48395 -> MY.NET.214.6:6355
TCP TTL:50 TOS:0x0 ID:0 DF
21S***** Seq: **0xC49CAD9A** Ack: 0x0 Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 94436437 0 EOL EOL EOL EOL

=+
=+=+=+
03/12-04:42:20.241817 209.85.251.2:48395 -> MY.NET.214.6:6355
TCP TTL:50 TOS:0x0 ID:0 DF
21S***** Seq: **0xC49CAD9A** Ack: 0x0 Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 94436737 0 EOL EOL EOL EOL

=+
=+=+=+
03/12-04:42:26.244872 209.85.251.2:48395 -> MY.NET.214.6:6355
TCP TTL:50 TOS:0x0 ID:0 DF
21S***** Seq: **0xC49CAD9A** Ack: 0x0 Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 94437337 0 EOL EOL EOL EOL

Any insights into internal machines such as compromise or possible dangerous or anomalous activity.

The following IP addresses were common internal addresses that were targeted:
MY.NET.217.74
MY.NET.178.42
MY.NET.222.2
MY.NET.144.54
MY.NET.98.110
MY.NET.98.123

Several were hit with port scans and then followed-up with specific port attacks consisting of the RingZero, Wingate, and Russian Dynamo attacks. These devices should be closely monitored for further malicious traffic. Perhaps a honeypot can be set up that mimics these devices and these addresses isolated from other devices on the network and all activity logged. As mentioned earlier the Sub7 Trojan can scan from the infected machine to other devices on the network so the machines associated with those IP addresses listed should be thoroughly checked.

Defensive recommendations.

Several of the following recommendations are fairly general and may already be in place at this time. However all security policies should be reviewed periodically.

Update Software with latest patches
Closely monitor security web sites for new exploits and security bulletins as well as patches that are released for operating systems and software packages.

Install RPCbind
I would also recommend installing rpcbind by V Wenema, which has similar characteristics for host-access as the famous tcp-wrapper by the same author.

Unnecessary services should be disabled.
Each host should be checked to see if there are any services that are not critical and in use. Check the inetd.conf file as well as the init.d part of any Unix system and comment out or remove the software packages and services that are not needed. Services that are enabled should be logged and periodically compared to see if any services have been activated by malicious activity.

Use Vulnerability scanning to test network
There are many packages out on the market to choose from that can scan your network for weak hosts. The software can even be unleashed on the weak hosts

to detect possible vulnerabilities and suggest ways to harden the device. Vulnerability scans should be done periodically with updated vulnerability software in order to detect any new exploits.

Implement Stateful Firewall.
There may already be a stateful firewall at this site however there are still malicious packets getting passed the network perimeter. ICMP scans should not be allowed as well as any TCP connections that do not originate from inside the firewall.

Install/Update Virus Protection
There was evidence of Trojans originating from inside the network and an updated virus definition and regular scans can help eliminate these problems. Educate your users on how to combat the viruses and malicious attachments that may propagate throughout the email system.

Block Known Hostile Addresses
From the firewall policies should be created to block access to and from the known Hostile addresses.

Description of Analysis Process:
The data that was downloaded from the SANS site consisted of a months worth of scans from the month of March. There were also Out-Of-Spec files as well as Snort Alert files. Using the Excel formulas provided by Mark Gryparis (SANS Network Security 2000 – Monterey CA October 2000) I was able to examine and parse the data into a Spreadsheet format. From there I was able to sort and query the data for further analysis.

Here are the Excel formulas used to parse the data: ta:

- For the Snort Alert spreadsheets, I edited the spreadsheet as follows:

Cel	*Contents/Formula*	*Result*
A1	**09/14-00:05:38.136984 [**] WinGate 1080 Attempt [**] 216.176.130.250:1201 -> MY.NET.98.194:1080**	---
B1	=LEFT($A2,5)	**09/14**
C1	=MID($A2,10,12)	**00:05:38.136984**
D1	=MID($A2,29,FIND("]",$A2,29)-33)	**WinGate 1080Attempt**
E1	=IF(ISERROR(FIND(" -> ",$A2)),"",MID($A2,FIND("] ",$A2,28)+2,FIND(":",$A2,28)-FIND("] ",$A2,28)-2))	**216.176.130.250**
F1	=IF(ISERROR(FIND(" -> ",$A2)),"",MID($A2,FIND(":",$A2,28)+1,FIND(" -> ",$A2)-1-FIND(":",$A2,28)))	**1201**
G1	=IF(ISERROR(FIND(" -> ",$A2)),"",MID($A2,FIND(" -> ",$A2)+4,FIND(":",$A2,FIND(" -> ",$A2))-FIND(" -> ",$A2)-4))	**MY.NET.98.194**

H1	=IF(ISERROR(FIND(" -> ",$A2)),"",RIGHT($A2,LEN($A2)-FIND(":",$A2,FIND(" -> ",$A2))))	**1080**

- Then I did a “Fill Down” on columns B through H, to parse out the Snort log entry in each row of Column A

Each of the days that were analyzed were placed into a separate spreadsheet files. Using the sort, count and subtotals feature I was able to gather information on the number of occurrences on a port as well as IP address basis. This gave me a quick look at common ports that were accessed as well as traffic totals from the more “active” IP addresses. I was also able to determine which “subnet blocks” seemed to be the source of the hostile traffic.

Once I had my list of ports and IP addresses I used the SANS website and other Internet sites to correlate the information and tactics used with other reported instances.

Sources Used:

http://www.sans.org
http://www.isc.org
http://xforce.iss.net
http://neworder.box.sk
http://www.securityfocus.com
http://www.whitehats.com
http://www.hack.co.za
http://www.glocksoft.com/trojan_list/Senna_Spy_Trojan_Generator.htm
http://www.sans.org/newlook/resources/IDFAQ/oddports.htm
http://www.sans.org/newlook/resources/IDFAQ/socks.htm
http://advice.networkice.com/advice/exploits/ports/27374/default.htm
http://subseven.slak.org/
http://www.securityfocus.com/bid/509.html
http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1413.html
http://www.securityfocus.com/bid/587.html
http://advice.networkice.com/advice/Intrusions/2001902/default.htm
http://alcor.concordia.ca/~rich/why-ident.txt
http://www.whitehats.com/IDS/29
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0454
http://www.sans.org/newlook/resources/IDFAQ/fragments.htm