GIAC LevelTwo
Intrusion Detection in Depth
SANS 2001
May 13-17, 2001
GCIA Practical Assignment Version 2.8b

Esperanza Lopez-Wilkin

Table of Contents

<u>**Note from the author**</u>: Most of this practical was completed before the outbreak or greatest impact of the CodeRed worm. Statements regarding to Top Ten Vulnerabilities, Top Ten Destination Port, etc., relate to information prior or about July 26, 2001.

Assignment 1 - Network Detects

1. Detect 1

1.1. Network Trace

```
03:09:38.889199 212.223.69.26.2399 > my.domain.smtp: S 784089924:784089924(0) win 16060 (DF)
03:09:38.890294 my.domain.smtp > 212.223.69.26.2399: S 1249254015:1249254015(0) ack
784089925 win 32120 (DF)
03:09:39.110668 212.223.69.26.2399 > my.domain.smtp: . ack 1249254016 win 16060 (DF)
03:09:39.114302 212.223.69.17 > my.domain: icmp: redirect 212.223.69.26 to host 212.223.69.26 [tos
0xc0]
03:09:39.380047 my.domain.smtp > 212.223.69.26.2399: P 1249254016:1249254097(81) ack
784089925 win 32120 (DF)
03:09:39.604213 212.223.69.17 > my.domain: icmp: redirect 212.223.69.26 to host 212.223.69.26 [tos
0xc0]
03:09:39.605360 212.223.69.26.2399 > my.domain.smtp: . ack 1249254097 win 16060 (DF)
03:09:39.839215 212.223.69.26.2399 > my.domain.smtp: P 784089925:784089945(20) ack
1249254097 win 16060 (DF)
03:09:39.840071 my.domain.smtp > 212.223.69.26.2399: . ack 784089945 win 32120 (DF)
03:09:39.840849 my.domain.smtp > 212.223.69.26.2399: P 1249254097:1249254161(64) ack
784089945 win 32120 (DF)
03:09:40.059937 212.223.69.17 > my.domain: icmp: redirect 212.223.69.26 to host 212.223.69.26 [tos
0xc0]
03:09:40.074585 212.223.69.26.2399 > my.domain.smtp: . ack 1249254161 win 16060 (DF)
03:09:40.082645 212.223.69.26.2399 > my.domain.smtp: P 784089945:784089984(39) ack
1249254161 win 16060 (DF)
03:09:40.098926 my.domain.smtp > 212.223.69.26.2399: . ack 784089984 win 32120 (DF) <>
```

<u>Message Mail Headers:</u>
Return-Path:
Received: from 212.223.69.26 ([212.223.69.26])
by my.domain (8.9.3/8.8.7) with SMTP id DAA15998
for someguy@mydomain.com Tue, 10 Apr 2001 03:09:40 -0600
From: handy-land@news-master.de

<u>Additional information:</u>
Date of event: April 10, 2001
Submitting Analyst: Curtis L. Blais
Submitting Analyst's Comments:
The mail appears to be some sort of spam, as the recipient is quite sure it is not legitimate. The ICMP was blocked at the perimeter. It almost appears as though it is some sort of attempt to hide the ICMP amongst the SMTP.

RIPE Information
Server Name: NS1.SOERVER.DE
IP Address: 212.223.69.17
Registrar: NETWORK SOLUTIONS, INC.

Whois Server: whois.networksolutions.com
Referral URL: www.networksolutions.com
person: Erika Zoettlein
address: ratiokontakt GmbH
address: Heganger 18
address: D-96103 Hallstadt
address: Germany

1.2. Source of trace

url: http://www.sans.org/y2k/041301.htm

1.3. Type of event generator

Event logs are assumed to be generated by tcpdump and, mail message headers by local mail program.

Tcpdump log description:

time **src_ip_addr.src_port** traffic_direction **dst_ip_addr.dst_port:** TCP_flags **TCP_sequence_no (payload)** reserved_word window_size <**TCP options**> (Don't fragment flag) **(time_to_live, IP packet id)**

Example:

```
03:09:38.889199 212.223.69.26.2399 > my.domain.smtp: S 784089924:784089924(0) win 16060 (DF)
```

Message Mail Header:

"Return-Path" is a reserved word indicating the return address for delivery notification
"Received" is a reserved word indicating beginning of message header
"from" is followed by source address
"by" is followed by destination address or name
"(8.9.3/8.8.7)" indicate Sendmail version and configuration file version, respectively
"with SMTP" indicates protocol used
"id" is followed by unique id number for the message
"for" is followed by message recipient's e-mail address followed by date and time.
"From:" is followed by sender's e-mail address

Example:

```
Return-Path:
Received: from 212.223.69.26 ([212.223.69.26]) by my.domain (8.9.3/8.8.7) with SMTP id DAA15998
for someguy@mydomain.com Tue, 10 Apr 2001 03:09:40 -0600
Received is a reserved word and it indicates the beginning of a header line
From: handy-land@news-master.de
```

1.4. Probability the source address was spoofed

The detect shows two IP addresses from the attacker's site. The probability of IP 212.223.69.26 address being spoofed is small since the attacker sustained a SMTP connection with host my.domain running the SMTP daemon.

The message mail headers information submitted with the report depicts the message received at host my.domain.smtp. The Received: header shows the source IP address but no host or domain name resolution for such IP address. Therefore, there is some probability that the 212.223.69.17 address could have been spoofed since the ICMP packet looks crafted.

1.5. Description of attack

Lines 1 through 3 show the source address (212.223.69.26) on port 2399 establishing a successful TCP connection to host my.domain on port 25 (SMTP). Since destination host allowed the handshake to be completed using port 25, the attacker would assume that Sendmail is running, therefore host my.domain should be some mail server.

After the three-way handshake, line 4 shows IP address 212.223.69.17 sending an ICMP redirect message. This activity was also repeated on lines 6 and 11. Lines 5, 7-10 and 12-14 show more of the SMTP connection between the two hosts.

The SMTP connection between 212.223.69.26 and host my.domain seems normal and a message was successfully sent to someguy@mydomain.com, verified by the message's mail headers provided with the trace.

There is only one mail Received: header in the message header supplied with the trace. The header does not include any host name and the IP address can not be resolved via DNS .

1.6. Attack mechanism

Port 25 running SMTP service was targeted on this trace. The attacker initiated a SMTP connection to host my.domain. This activity may have been preceded by some reconnaissance work to find out if port 25 (the default port to run SMTP services) on host my.domain was opened.

Note: ICMP redirect message are issued by routers to notify a host of a more efficient route for IP datagram packet exchange with the destination host. This data results in alteration of the initial sending host's routing table. ICMP redirect messages are sent by a router connected to the same network as the sending host.

The attacker sent a crafted ICMP redirect message to host my.domain. The ICMP packets sent on this trace seem crafted for various reasons: it tries to redirect traffic to a host (212.223.69.26), not a router; it duplicates the IP address of the source (212.223.69.26) device as the target device; and it does not show which destination host is sending this redirect message on behalf of. IP address 212.223.69.17 tried to impersonate a router by sending this crafted message.

It was considered possible reconnaissance activity since sending a SMTP message to a bogus e-mail address on a valid domain such as my.domain, would produce a non-delivery notification to the original message sender with information about SMTP servers (hostname, IP addresses, operating system type and version, SMTP service type and version) in the organization that could be used on future activity to exploit the Sendmail application. Based on the message mail header provided with the trace, it was concluded that a non-delivery notification would not reach the original sender since the Return-Path: mail header has no value to be used to address the non-delivery notification message.

It was also considered that the attacker targeted the Sendmail program on exploit attempt since Sendmail has many known vulnerabilities. A novice attacker may have confused a vulnerability for ICMP redirect messages with Sendmail's redirect vulnerability, which relates to the possibility of the SMTP server relaying messages for another domain if running Sendmail on promiscuous mode.

The Type of Service(TOS) value 0xc0 was researched since the value is higher than those recommended for some applications as listed in the Stevens [1] book. The value 0xc0 uses the two higher-order bits in the 3-bit precedence field. This analyst found a document in the IDFAQ (http://www.sans.org/newlook/resources/IDFAQ/egress_benefits.htm) which lists ICMP port unreachable messages generated by a firewall with this TOS value as 0xc0. Therefore, the assumption of this value being invalid was discarted.

1.7. Correlation

The attacker may have preceded this activity with some reconnaissance work to identify host my.domain as a mail server since it had port 25 opened, which is the default port for SMTP service. Establishing a successful SMTP connection with host my.domain actually confirmed that the SMTP daemon was listening on port 25. A denial of service attack on a mail server by attempting to crash the server with ICMP redirect messages, seems more rewarding given its usual importance to any organization.

The attacker sent the ICMP redirect messages as a denial of service attack.

Reference:
CVE-1999-0265: ICMP redirect messages may crash or lock up a host.

Some research was done on possible vulnerabilities related to Sendmail version 8.9.3, but no CVE were found.

1.8. Evidence of active targeting

Host my.domain was targeted under this attack since the attacker not only opened a successful SMTP connection with the targeted host, but also sent crafted ICMP redirect messages to the same host running SMTP service on default port 25.

1.9. Severity

(Criticality + Lethality) – (System + Network Countermeasures) = Severity
(5 + 5) - (4 + 5) = 1

System Criticality = 5 (The SMTP server was targeted.)
Attack Lethality = 5 (The attacker attempted a denial of service attack.)
System Countermeasures = 4 (Sendmail running does provide some vulnerabilities. Host my.domain running version is 8.9.3, not the most recent, but seems stable. Host

my.domain accepted SMTP connection and message from attacker, but it is normal behavior for a listening SMTP server.)
Network Countermeasures = 5 (The submitting analyst recorded that the ICMP redirect message was blocked at the perimeter.)

The attacker targeted a server with port 25 opened and listening for SMTP messages.

1.10. Defensive Recommendations

The first defensive countermeasure seems to be already in place: blocking incoming ICMP redirect messages at the perimeter for internal host. OS implementation and patch updates on host my.domain can assure that ICMP redirect messages are checked before modifying its routing table and invalid messages dropped silently. If Sendmail is running on the server, update the application with most updated release.

1.11. Multiple choice test question

ICMP redirect messages are generated by routers to notify:

a) a host on a remote network that a more direct route is available
b) a host on the network directly connected to the router that a more direct route is available for the destination address
c) another router not to send message to the proxy server
d) a proxy server about host up/down condition of an internal host

Answer B

2. Detect 2

2.1. Network Trace

```
Jun 27 08:42:07 210.52.214.15:111 -> a.b.c.4:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.c.18:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.c.26:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.c.27:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.c.51:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.c.59:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:636 -> a.b.c.62:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.c.101:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.c.194:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.c.195:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.c.209:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.c.212:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.c.237:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.d.52:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.d.204:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.d.215:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.d.218:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.d.220:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.d.221:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.d.233:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.d.245:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.d.250:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.d.249:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.d.251:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.d.254:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.e.42:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.e.48:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.e.79:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.e.88:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.e.97:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.e.106:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.e.116:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.e.128:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.e.165:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.e.176:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.e.179:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.e.184:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.e.186:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.e.195:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.e.217:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.e.229:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.e.238:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.e.241:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.f.11:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.f.28:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.f.32:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.f.39:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.f.41:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.f.49:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.f.74:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.f.132:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.f.145:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.f.164:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.f.166:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.f.176:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.f.183:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.f.190:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.f.192:111 SYN ******S*
Jun 27 08:42:07 210.52.214.15:111 -> a.b.f.244:111 SYN ******S*
```

Syslog:

```
Jun 27 08:42:08 hosth portmap[392]: connect from 210.52.214.15 to dump(): request from unauthorized host
Jun 27 08:42:08 hosth snort: RPC Info Query: 210.52.214.15:636 -> a.b.c.62:111
```

Other submitted information:

```
    Server used for this query: [ whois.apnic.net ]
    inetnum:     210.52.128.0 - 210.52.255.255
    netname:     CNCNET
    descr:       China Netcom Corp.
    descr:       New Telecommunication Carrier Based on IP Backbone
    country:     CN
```

2.2. Source of trace

E-mail sent from Laurie Zirkle to intrusions@incidents.org mailing list on 6/28/01 with Subject: June 27, 2001 probes (part2)

2.3. Type of event generator

Not exactly sure, but seems reasonable to assume Snort for the network sensor and PortSentry for the host.

Snort log format description:

month day time **src_addr:src_port** traffic_direction **dst_addr:dst_port** message_type tcp_flags

Example:

Jun 27 08:42:07 **210.52.214.15:111 -> a.b.f.244:111** SYN ******S*

Syslog format description:

Date and time **hostname** logger application: **description of alert**

Example:

```
Jun 27 08:42:08 hosth portmap[392]: connect from 210.52.214.15 to dump(): request from unauthorized host
```

2.4. Probability the source address was spoofed

The probability of the source address being spoofed is small since the attacker attempted to gather information about portmapper service running on targeted host. The attacked would not have worked with a spoofed IP address.

2.5. Description of attack

Source IP address 210.52.214.15 on reserved port 111 scanned various subnetworks on targeted site by sending SYN packets to hosts on port 111. Time stamp shows a very fast scan to various hosts on the same network.

The seventh line of the trace shows the attacker attempting a RPC dump to list all registered RPC programs running on host a.b.c.62. Reconnaissance activity was blocked by PortSentry at the host a.b.c.62, as shown in the syslog excerpt from the submitting analyst.

2.6. Attack mechanism

The trace shows only traffic destined to the targeted network. All traffic is stimulus from the same source IP address: from source port 111 sending SYN packets and from source port 636 to attempt a RPC query.

Given the intensity of the scan, the attacker may have been trying to deviate the attention from the real reconnaissance activity targeted at host a.b.c.62. Since Windows NT/2000 uses TCP port 135 for the portmap service, we assume that the attacker either performed some previous reconnaissance to determine that host a.b.c.62 is running Unix where portmap service runs on port 111. The attacker may have just assumed Unix OS since Unix is more prevalent in educational institutions such as the targeted site. Previous reconnaissance activity could have also gathered that host a.b.c.62 had port 111 opened.

The RPC dump function provides program number and port where RPC services run on the host server. RPC dump is done on Unix hosts by executing command:
rpcinfo –p a.b.c.62, where a.b.c.62 is the targeted IP address, like in the trace shown above.

The attacker sent crafted packets since the source is port 111 which is a reserved port or compromise the source host.

2.7. Correlation

Source IP address is from Asia Pacific block, where significant reconnaissance and attack activity is generated.

Various network traces submitted by Laurie Zirkle at different times, show port 111 opened on host a.b.c.62 (assuming that Ms. Zirkle always replaces the octets with the same letter values). Even though network traces show only incoming traffic, initial SYN packets addressed to host a.b.c.62 on port 111 are often followed by UDP packets addressed to the same host on the same port. Refer to e-mail from Laurie Zirkle on 7/9/01 with subject “July 6, 2001 probes (part 2)”, as an example.

Same source IP address performed a SYN scan on same target network on 4/14/01, as reported on:
http://www.incidents.org/archives/y2k/042401.htm

Attacks on port 111 appear consistently on the Top 10 Attacks at www.incidents.org: as second most popular within the last 7 days and third in last 30 and 60 days. This reconnaissance activity targeted to port 111 on host a.b.c.62 can be used to further exploit RPC vulnerabilities, such as the following CVEs:

CVE-1999-0168: The portmapper may act as a proxy and redirect service requests from an attacker, making the request appear to come from the local host…

CVE-1999-0212: Solaris rpc.mountd generates error messages that allow a remote attacker to determine what files are on the server.

Other Cert Advisories related to RPC vulnerabilities:
http://www.cert.org/incident_notes/IN-98.02.html
http://www.cert.org/incident_notes/IN-98.04.html
http://www.cert.org/advisories/CA-98.12.mountd.html

2.8. Evidence of active targeting

The network was targeted. Host a.b.c.62 was further targeted with reconnaissance activity.

2.9. Severity

(Criticality + Lethality) – (System + Network Countermeasures) = Severity
(4 + 4) - (5 + 0) = 3

System Criticality = 4 (Attacker targeted server running portmap or with port 111 opened, if correlation assumptions are correct.)
Attack Lethality = 4 (This was a reconnaissance activity, still it targeted portmap service which has known vulnerabilities.)
System Countermeasures = 5 (RPC dump was blocked at the host.)
Network Countermeasures = 0 (Packet reached the host.)

The attacker targeted a server with port 111 opened and running RPC services on port 111.

2.10. Defensive Recommendations

At border gateway, block inbound TCP and UDP traffic destined to port 111.
For further reference:
http://www.cert.org/tech_tips/packet_filtering.html

Also, limit/block access to TCP and UDP port 111 with TCP Wrappers or local IPchains firewall at the host. Another alternative is to run Wietse's secure portmap software if the host most run portmap service.

2.11. Multiple choice test question

Portmap service running on port 111 is used:

a) to provide direct connectivity to other services such as SMTP
b) to exploit known vulnerabilities related to DNS BIND configuration
c) to keep track of RPC services running on the local host
d) create ping sweeps against other host on the local network

Answer C

3. Detect 3

3.1. Network Trace

```
23:04:56.357826 210.208.142.135 > my.host: icmp: echo request (ttl 14, id 6843)
23:04:56.382474 my.host > 210.208.142.135: icmp: echo reply (ttl 128, id 2644)
23:04:57.052298 210.208.142.135.1186 > my.host.161: |30|2c|02|01|04|0aC=islmonitor
|a0|1bGetRequest(7)|02|03|02|01[|snmp] (ttl 110, id 6979)
23:04:57.052380 my.host > 210.208.142.135: icmp: my.host udp port 161 unreachable (ttl 128, id 2645)
23:04:59.044465 210.208.142.135.1186 > my.host.161: |30|2c|02|01|04|0aC=islmonitor
|a0|1bGetRequest(7)|02|03|02|01[|snmp] (ttl 110, id 7333)
23:04:59.044550 my.host > 210.208.142.135: icmp: my.host udp port 161 unreachable (ttl 128, id 2646)
23:05:05.056079 210.208.142.135.1186 > my.host.161: |30|2c|02|01|04|0aC=islmonitor
|a0|1bGetRequest(7)|02|03|02|01[|snmp] (ttl 110, id 7515)
23:05:05.056148 my.host > 210.208.142.135: icmp: my.host udp port 161 unreachable (ttl 128, id 2647)
23:05:07.956246 210.208.142.135.1186 > my.host.161: |30|2c|02|01|04|0aC=islmonitor
|a0|1bGetRequest(7)|02|03|02|01[|snmp] (ttl 110, id 7683)
23:05:07.956327 my.host > 210.208.142.135: icmp: my.host udp port 161 unreachable (ttl 128, id 2648)
23:05:10.263322 210.208.142.135.1186 > my.host.161: |30|2c|02|01|04|0aC=islmonitor
|a0|1bGetRequest(7)|02|03|02|01[|snmp] (ttl 110, id 8033)
23:05:10.263406 my.host > 210.208.142.135: icmp: my.host udp port 161 unreachable (ttl 128, id 2649)
23:05:12.293912 210.208.142.135.1186 > my.host.161: |30|2c|02|01|04|0aC=islmonitor
|a0|1bGetRequest(7)|02|03|02|01[|snmp] (ttl 110, id 8305)
23:05:12.293983 my.host > 210.208.142.135: icmp: my.host udp port 161 unreachable (ttl 128, id 2650)
```

3.2. Source of trace

My pc connected to @HOME network via DSL.

3.3. Type of event generator

Windump 2.02 was used to capture network packets.

Windump log description:

time **src_ip_addr.src_port** traffic_direction **dst_ip_addr.dst_port:** truncated UDP payload for SNMP transaction **(time_to_live, IP packet id)**

Example:

23:05:12.293912 **210.208.142.135.1186** > **my.host.161**: |30|2c|02|01|04|0aC=islmonitor |a0|1bGetRequest(7)|02|03|02|01[|snmp] **(ttl 110, id 8305)**

3.4. Probability the source address was spoofed

The probability of the source being spoofed is small since the attacker attempted to make SNMP calls to gather information from my host. Reconnaissance activity would not work with a spoofed IP address.

3.5. Description of attack

The attacker sent an echo request to host my.host and my.host responded with an echo reply acknowledging that the host is active. The attacker sent multiple attempts to gather SNMP configuration information with the “GetRequest” SNMP call to my host at UDP port 161. My host responded with “port 161 unreachable” since SNMP services are not running and port is not opened.

3.6. Attack mechanism

SNMP Get Request calls are sent to SNMP agents to gather information from the host running the SNMP agent. The trace shows stimulus from the attacker trying to gather value of SNMP community string with the Get Request command. If the targeted host has an unencrypted or common community string, such as “public” or “private”, the attacker can further exploit this information and, gather SNMP configuration information, change SNMP parameters and gain control of the system. If the atttacker’s host and targeted host share the community string, the attacker can manipulate the targeted host with SNMP commands.

3.7. Correlation

SNMP is known to be a unsecured protocol since it transmits services password in the clear and many implementations use the default community string “public” or “private” facilitating exploits by attackers. SNMP vulnerability appears listed on SANS “How to Eliminate the Ten Most Critical Internet Security Threats.”

Some Common Vulnerabilities and Exposures related to SNMP service and community string access are:
CAN-1999-0517: An SNMP community name is the default (e.g. public), null, or missing.

CAN-1999-0516: An SNMP community name is guessable.

CAN-2001-0046: The default permissions for the SNMP Parameters registry key in Windows NT 4.0 allows remote attackers to read and possibly modify the SNMP community strings to obtain sensitive information or modify network configuration, aka one of the "Registry Permissions" vulnerabilities.

CAN-1999-0499: NETBIOS share information may be published through SNMP registry keys in NT.

CAN-1999-0254: A hidden SNMP community string in HP OpenView allows remote attackers to modify MIB tables and obtain sensitive information.

CAN-1999-0186: In Solaris, an SNMP subagent has a default community string that allows remote attackers to execute arbitrary commands as root, or modify system parameters.

From Security Focus website:
Microsoft Windows NT & 2000 SNMP Registry Key Modification Vulnerability
http://www.securityfocus.com/archive/1/149939

SNMP reconnaissance on WinNT and WINS DoS:
http://www.securityfocus.com/archive/1/7756

SNMP vulnerability also on wireless LAN:
http://www.securityfocus.com/archive/1/192301

There has been scans and hostile activity from Asia Pacific block. This trace was initiated from Taiwan:
Server used for this query: [whois.apnic.net]

```
inetnum                210.208.128.0 - 210.208.159.255
netname                ISL-TW
descr                  Internet Solution Lab.
descr                  Taipei
country                TW
```

For more information regarding SNMP vulnerabilities in Microsoft networks:
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/fq00-095.asp

3.8. Evidence of active targeting

Because my host's connection to the ISP network does not allow to collect packets besides those emanated from or destined to my.host, the trace is limited and does not show reconnaissance targeted to other segments of the network. Still, rapid changes in the IP id numbers of the various packets received, leads to believe that other reconnaissance on other hosts was being performed around that time.

Since my.host does not run the SNMP service and the attacker only checked for the host to be active before attempting SNMP information gathering, it is assumed that the SNMP service was targeted on many hosts for reconnaissance purposes to further exploit SNMP vulnerabilities.

3.9. Severity

(Criticality + Lethality) – (System + Network Countermeasures) = Severity
(3 + 3) - (5 + 0) = 1
System Criticality = 3 (SNMP service was targeted. No specific host was targeted but scan moved quickly.)
Attack Lethality = 3 (Since scan targeted vulnerable port and service.)
System Countermeasures = 5 (My host was not running SNMP service and port was not opened.)

Network Countermeasures = 0 (The ISP network does not filter external SNMP traffic.)

3.10. Defensive Recommendations

Make sure personal firewall is enabled since my host is connected to @HOME network where possible malicious traffic from external sources is not filtered. Disable SNMP services (already done) if not needed.

3.11. Multiple choice test question

SNMP community strings are used by SNMP manager to:

a) control services on hosts running SNMP agent
b) exploit ICMP redirect messages
c) run over TCP because it is a more reliable transport protocol
d) exploit other Windows NT vulnerabilities related to IIS Internet Information Server

Answer A

4. Detect 4

4.1. Network Trace

```
Jul  5 02:05:13 203.106.224.87:64117 -> MY.SUB.NET.4:21 SYN ******S*
Jul  5 02:05:13 203.106.224.87:64119 -> MY.SUB.NET.4:23 SYN ******S*
Jul  5 02:05:13 203.106.224.87:64129 -> MY.SUB.NET.4:79 SYN ******S*
Jul  5 02:05:12 203.106.224.87:64076 -> MY.SUB.NET.19:21 SYN ******S*
Jul  5 02:05:13 203.106.224.87:64163 -> MY.SUB.NET.19:79 SYN ******S*
Jul  5 02:05:13 203.106.224.87:64141 -> MY.SUB.NET.29:23 SYN ******S*
Jul  5 02:05:14 203.106.224.87:64198 -> MY.SUB.NET.29:79 SYN ******S*
Jul  5 02:05:12 203.106.224.87:64114 -> MY.SUB.NET.30:21 SYN ******S*
Jul  5 02:05:15 203.106.224.87:64217 -> MY.SUB.NET.30:79 SYN ******S*
Jul  5 02:05:13 203.106.224.87:64164 -> MY.SUB.NET.31:79 SYN ******S*
Jul  5 02:05:13 203.106.224.87:64161 -> MY.SUB.NET.31:21 SYN ******S*
Jul  5 02:05:13 203.106.224.87:64162 -> MY.SUB.NET.31:23 SYN ******S*
Jul  5 02:05:13 203.106.224.87:64128 -> MY.SUB.NET.5:21 SYN ******S*
Jul  5 02:05:13 203.106.224.87:64130 -> MY.SUB.NET.25:23 SYN ******S*
Jul  5 02:05:14 203.106.224.87:64197 -> MY.SUB.NET.25:79 SYN ******S*
Jul  5 02:05:13 203.106.224.87:64134 -> MY.SUB.NET.32:79 SYN ******S*
Jul  5 02:05:14 203.106.224.87:64177 -> MY.SUB.NET.32:21 SYN ******S*
Jul  5 02:05:14 203.106.224.87:64192 -> MY.SUB.NET.32:23 SYN ******S*
Jul  5 02:05:13 203.106.224.87:64137 -> MY.SUB.NET.14:79 SYN ******S*
Jul  5 02:05:13 203.106.224.87:64154 -> MY.SUB.NET.35:79 SYN ******S*
Jul  5 02:05:13 203.106.224.87:64159 -> MY.SUB.NET.36:21 SYN ******S*
Jul  5 02:05:13 203.106.224.87:64165 -> MY.SUB.NET.36:23 SYN ******S*
Jul  5 02:05:14 203.106.224.87:64167 -> MY.SUB.NET.36:79 SYN ******S*
Jul  5 02:05:13 203.106.224.87:64035 -> MY.SUB.NET.6:21 SYN ******S*
Jul  5 02:05:14 203.106.224.87:64172 -> MY.SUB.NET.37:23 SYN ******S*
Jul  5 02:05:14 203.106.224.87:64174 -> MY.SUB.NET.37:79 SYN ******S*
Jul  5 02:05:14 203.106.224.87:64180 -> MY.SUB.NET.38:23 SYN ******S*
Jul  5 02:05:14 203.106.224.87:64182 -> MY.SUB.NET.38:79 SYN ******S*
Jul  5 02:05:14 203.106.224.87:64187 -> MY.SUB.NET.39:23 SYN ******S*
Jul  5 02:05:14 203.106.224.87:64200 -> MY.SUB.NET.40:21 SYN ******S*
Jul  5 02:05:14 203.106.224.87:64208 -> MY.SUB.NET.40:79 SYN ******S*
Jul  5 02:05:15 203.106.224.87:64225 -> MY.SUB.NET.37:23 SYN ******S*
Jul  5 02:05:16 203.106.224.87:64270 -> MY.SUB.NET.37:21 SYN ******S*
Jul  5 02:05:16 203.106.224.87:64290 -> MY.SUB.NET.37:79 SYN ******S*
Jul  5 02:05:15 203.106.224.87:64239 -> MY.SUB.NET.39:21 SYN ******S*
Jul  5 02:05:15 203.106.224.87:64241 -> MY.SUB.NET.39:23 SYN ******S*
Jul  5 02:05:16 203.106.224.87:64287 -> MY.SUB.NET.38:79 SYN ******S*
Jul  5 02:05:16 203.106.224.87:64288 -> MY.SUB.NET.38:21 SYN ******S*
Jul  5 02:05:15 203.106.224.87:64247 -> MY.SUB.NET.29:21 SYN ******S*
Jul  5 02:05:15 203.106.224.87:64250 -> MY.SUB.NET.29:23 SYN ******S*
Jul  5 02:05:15 203.106.224.87:64248 -> MY.SUB.NET.32:23 SYN ******S*
Jul  5 02:05:15 203.106.224.87:64249 -> MY.SUB.NET.32:79 SYN ******S*
Jul  5 02:05:15 203.106.224.87:64251 -> MY.SUB.NET.25:23 SYN ******S*
Jul  5 02:05:15 203.106.224.87:64252 -> MY.SUB.NET.25:79 SYN ******S*
Jul  5 02:05:16 203.106.224.87:64258 -> MY.SUB.NET.40:23 SYN ******S*
Jul  5 02:05:16 203.106.224.87:64259 -> MY.SUB.NET.40:79 SYN ******S*
Jul  5 02:05:16 203.106.224.87:64262 -> MY.SUB.NET.40:21 SYN ******S*
Jul  5 02:05:16 203.106.224.87:64257 -> MY.SUB.NET.35:21 SYN ******S*
Jul  5 02:05:16 203.106.224.87:64263 -> MY.SUB.NET.35:79 SYN ******S*
Jul  5 02:05:16 203.106.224.87:64260 -> MY.SUB.NET.30:21 SYN ******S*
Jul  5 02:05:16 203.106.224.87:64272 -> MY.SUB.NET.30:79 SYN ******S*
Jul  5 02:05:16 203.106.224.87:64271 -> MY.SUB.NET.42:23 SYN ******S*
Jul  5 02:05:16 203.106.224.87:64274 -> MY.SUB.NET.42:79 SYN ******S*
Jul  5 02:05:16 203.106.224.87:64273 -> MY.SUB.NET.36:79 SYN ******S*
Jul  5 02:05:16 203.106.224.87:64275 -> MY.SUB.NET.36:21 SYN ******S*
Jul  5 02:05:16 203.106.224.87:64165 -> MY.SUB.NET.36:23 SYN ******S*
Jul  5 02:05:17 203.106.224.87:64300 -> MY.SUB.NET.35:21 SYN ******S*
Jul  5 02:05:18 203.106.224.87:64349 -> MY.SUB.NET.40:79 SYN ******S*
Jul  5 02:05:18 203.106.224.87:64350 -> MY.SUB.NET.40:21 SYN ******S*
Jul  5 02:05:18 203.106.224.87:64348 -> MY.SUB.NET.40:23 SYN ******S*
Jul  5 02:05:18 203.106.224.87:64356 -> MY.SUB.NET.42:23 SYN ******S*
Jul  5 02:05:18 203.106.224.87:64357 -> MY.SUB.NET.42:79 SYN ******S*
Jul  5 02:05:17 203.106.224.87:64312 -> MY.SUB.NET.36:79 SYN ******S*
Jul  5 02:05:18 203.106.224.87:64339 -> MY.SUB.NET.36:23 SYN ******S*
Jul  5 02:05:17 203.106.224.87:64321 -> MY.SUB.NET.37:23 SYN ******S*
Jul  5 02:05:19 203.106.224.87:64371 -> MY.SUB.NET.36:23 SYN ******S*
Jul  5 02:05:20 203.106.224.87:64400 -> MY.SUB.NET.31:21 SYN ******S*
Jul  5 02:05:22 203.106.224.87:64437 -> MY.SUB.NET.190:21 SYN ******S*
Jul  5 02:05:23 203.106.224.87:64439 -> MY.SUB.NET.190:79 SYN ******S*
Jul  5 02:05:24 203.106.224.87:64463 -> MY.SUB.NET.192:79 SYN ******S*
Jul  5 02:05:24 203.106.224.87:64460 -> MY.SUB.NET.192:21 SYN ******S*
Jul  5 02:05:24 203.106.224.87:64450 -> MY.SUB.NET.203:23 SYN ******S*
Jul  5 02:05:24 203.106.224.87:64451 -> MY.SUB.NET.203:79 SYN ******S*
Jul  5 02:05:24 203.106.224.87:64452 -> MY.SUB.NET.196:21 SYN ******S*
Jul  5 02:05:24 203.106.224.87:64453 -> MY.SUB.NET.204:21 SYN ******S*
Jul  5 02:05:24 203.106.224.87:64455 -> MY.SUB.NET.204:79 SYN ******S*
Jul  5 02:05:25 203.106.224.87:64471 -> MY.SUB.NET.204:23 SYN ******S*
Jul  5 02:05:25 203.106.224.87:64473 -> MY.SUB.NET.196:79 SYN ******S*
Jul  5 02:05:25 203.106.224.87:64475 -> MY.SUB.NET.211:23 SYN ******S*
Jul  5 02:05:26 203.106.224.87:64512 -> MY.SUB.NET.211:21 SYN ******S*
Jul  5 02:05:25 203.106.224.87:64478 -> MY.SUB.NET.190:79 SYN ******S*
Jul  5 02:05:25 203.106.224.87:64477 -> MY.SUB.NET.205:23 SYN ******S*
Jul  5 02:05:25 203.106.224.87:64479 -> MY.SUB.NET.212:21 SYN ******S*
Jul  5 02:05:25 203.106.224.87:64480 -> MY.SUB.NET.192:21 SYN ******S*
Jul  5 02:05:26 203.106.224.87:64491 -> MY.SUB.NET.210:79 SYN ******S*
Jul  5 02:05:26 203.106.224.87:64505 -> MY.SUB.NET.203:23 SYN ******S*
Jul  5 02:05:27 203.106.224.87:64522 -> MY.SUB.NET.241:21 SYN ******S*
Jul  5 02:05:28 203.106.224.87:64556 -> MY.SUB.NET.244:21 SYN ******S*
Jul  5 02:05:27 203.106.224.87:64531 -> MY.SUB.NET.244:23 SYN ******S*
Jul  5 02:05:28 203.106.224.87:64560 -> MY.SUB.NET.244:79 SYN ******S*
Jul  5 02:05:28 203.106.224.87:64561 -> MY.SUB.NET.245:21 SYN ******S*
Jul  5 02:05:27 203.106.224.87:64534 -> MY.SUB.NET.245:23 SYN ******S*
Jul  5 02:05:28 203.106.224.87:64543 -> MY.SUB.NET.211:79 SYN ******S*
Jul  5 02:05:28 203.106.224.87:64550 -> MY.SUB.NET.248:23 SYN ******S*
Jul  5 02:05:28 203.106.224.87:64551 -> MY.SUB.NET.248:79 SYN ******S*
Jul  5 02:05:28 203.106.224.87:64552 -> MY.SUB.NET.249:21 SYN ******S*
Jul  5 02:05:28 203.106.224.87:64555 -> MY.SUB.NET.249:79 SYN ******S*
Jul  5 02:05:29 203.106.224.87:64315 -> MY.SUB.NET.42:21 SYN ******S*
Jul  5 02:05:29 203.106.224.87:64566 -> MY.SUB.NET.248:21 SYN ******S*
Jul  5 02:05:30 203.106.224.87:64590 -> MY.SUB.NET.248:79 SYN ******S*
Jul  5 02:05:30 203.106.224.87:64591 -> MY.SUB.NET.244:21 SYN ******S*
Jul  5 02:05:29 203.106.224.87:64580 -> MY.SUB.NET.244:79 SYN ******S*
Jul  5 02:05:30 203.106.224.87:64592 -> MY.SUB.NET.243:23 SYN ******S*
Jul  5 02:05:30 203.106.224.87:64594 -> MY.SUB.NET.243:79 SYN ******S*
Jul  5 02:05:30 203.106.224.87:64593 -> MY.SUB.NET.249:23 SYN ******S*
Jul  5 02:05:30 203.106.224.87:64595 -> MY.SUB.NET.249:79 SYN ******S*
Jul  5 02:05:30 203.106.224.87:64582 -> MY.SUB.NET.240:79 SYN ******S*
Jul  5 02:05:31 203.106.224.87:64606 -> MY.SUB.NET.248:23 SYN ******S*
```

4.2. Source of trace

E-mail sent from Paul Asadoorian to intrusions@incidents.org mailing list on 7/6/01 with Subject: Interesting Scan Pattern

4.3. Type of event generator

It is assume Snort.

Snort log format description:
month day time **src_addr:src_port** traffic_direction **dst_addr:dst_port** message_type tcp_flags

Example:
Jul 5 02:05:31 **203.106.224.87:64606** -> **MY.SUB.NET.248:23** SYN ******S*

4.4. Probability the source address was spoofed

The probability of the source address being spoofed is small since reconnaissance activity requires responses to get back to the source.

4.5. Description of attack

Host 203.106.224.87 from the Asia-Pacific block sent TCP packets with SYN flags as a reconnaissance mechanism to port 21 (ftp), 23 (telnet) and 79 (finger).

4.6. Attack mechanism

The attacker sent reconnaissance stimulus to the network of interest. There are scanning tools that can target multiple ports on multiple hosts running in parallel, such as Spidermap which is a collection of PERL scripts to perform tuned scans:
http://www.digitaloffense.net/spidermap/spidermap-0.1/SPIDERMAP.README

Reconnaissance scan was targeted port 21, 23 and 79 which are the default ports for FTP, Telnet and Finger, respectively.

4.7. Correlation

Telnet and Finger can be used in combination to query a server on the Internet about local accounts. Telnet and FTP services send password over the network in clear text. The attacker could be using this reconnaissance activity to further compromise an internal server and exploit this Telnet and FTP feature vulnerability.

Port 21 has known vulnerabilities ranging from denial of service exploits to an attacker gaining root access and using the FTP service to access other systems. Mitre.org website lists many vulnerabilities associated with the FTP service, for example:
CVE-1999-0017: FTP servers can allow an attacker to connect to arbitrary ports on machines other than the FTP client, aka FTP bounce.

CVE-1999-0054: Sun's ftpd daemon can be subjected to a denial of service.

CVE-1999-0080: wu-ftp FTP server allows root access via "site exec" command.

CVE-1999-0201: A quote cwd command on FTP servers can reveal the full path of the home directory of the "ftp" user.

CVE-1999-0879: Buffer overflow in WU-FTPD and related FTP servers allows remote attackers to gain root privileges via macro variables in a message file.

The telnet service also has many known vulnerabilities: root access exploits, including buffer overflow, and denial of service. Here are some vulnerabilities listed in the Mitre.org website for the telnet service:
CVE-1999-0073: Telnet allows a remote client to specify environment variables including LD_LIBRARY_PATH, allowing an attacker to bypass the normal system libraries and gain root access.

CVE-1999-0087: Denial of service in AIX telnet can freeze a system and prevent users from accessing the server.

CVE-1999-0749: Buffer overflow in Microsoft Telnet client in Windows 95 and Windows 98 via a malformed Telnet argument.

CVE-2000-0834: The Windows 2000 telnet client attempts to perform NTLM authentication by default, which allows remote attackers to capture and replay the NTLM challenge/response via a telnet:// URL that points to the malicious server, aka the "Windows 2000 Telnet Client NTLM Authentication" vulnerability.

Some Common Vulnerabilities and Exposures and candidate entries listed in Mitre.org related to the finger service:
CVE-1999-0150: The Perl fingerd program allows arbitrary command execution from remote users.

CVE-1999-0612: A version of finger is running that exposes valid user information to any entity on the network.

CVE-1999-0797: NIS finger allows an attacker to conduct a denial of service via a large number of finger requests, resulting in a large number of NIS queries.

CVE-2000-0128: The Finger Server 0.82 allows remote attackers to execute commands via shell metacharacters.

CAN-1999-0106: Finger redirection allows finger bombs.

No other correlation was found in incidents.org website relating to this address or address block.

4.8. Evidence of active targeting

Active targeting was focused on port numbers rather than specific hosts. Ports 21 (FTP), 23 (Telnet) and 79 (Finger) were the target of reconnaissance on this trace's activity. Because so many hosts were scanned over short period of time, it indicates a fast scanning tool with simultaneous processes such as the one described above.

4.9. Severity

(Criticality + Lethality) – (System + Network Countermeasures) = Severity
(3 + 3) - (3 + 0) = 3

System Criticality = 3 (This was a scan at large, no specific servers were targeted.)
Attack Lethality = 3 (This activity was done for reconnaissance but it has potential for exploits if targeted services are running.)
System Countermeasures = 3 (Don't know much details on this one since trace recorded only incoming traffic.)
Network Countermeasures = 0 (Scan reached the targeted host, assuming that sensors are inside border gateway.)

4.10. Defensive Recommendations

Block incoming TCP traffic to port 21, 23 and 79 at border gateway.

Unless absolutely needed, disable FTP, Telnet and Finger on internal hosts. If the FTP service is needed, TCP Wrappers can provide further host security or FTP server can be placed on a DMZ, separating it from the internal network. Use SSH instead of Telnet if access is needed for remote host management from internal hosts to external host, and disable telnet service on external hosts. Since Finger gives out too much information, it should always be disabled.

4.11. Multiple choice test question

```
Jul  5 02:05:28 203.106.224.87:64556 -> MY.SUB.NET.244:21 SYN ******S*
Jul  5 02:05:27 203.106.224.87:64531 -> MY.SUB.NET.244:23 SYN ******S*
Jul  5 02:05:28 203.106.224.87:64560 -> MY.SUB.NET.244:79 SYN ******S*
Jul  5 02:05:28 203.106.224.87:64561 -> MY.SUB.NET.245:21 SYN ******S*
Jul  5 02:05:27 203.106.224.87:64534 -> MY.SUB.NET.245:23 SYN ******S*
Jul  5 02:05:28 203.106.224.87:64543 -> MY.SUB.NET.211:79 SYN ******S*
Jul  5 02:05:28 203.106.224.87:64550 -> MY.SUB.NET.248:23 SYN ******S*
Jul  5 02:05:28 203.106.224.87:64551 -> MY.SUB.NET.248:79 SYN ******S*
Jul  5 02:05:28 203.106.224.87:64552 -> MY.SUB.NET.249:21 SYN ******S*
Jul  5 02:05:28 203.106.224.87:64555 -> MY.SUB.NET.249:79 SYN ******S*
```

The trace above shows a scan for services:

a) SMTP, FTP and RPC
b) SMTP, RPC and Finger
c) FTP, DNS and SSH
d) FTP, Telnet and Finger

Answer D

5. Detect 5

5.1. Network Trace

```
21:55:44.907444 142.177.227.30.4865 > my.host.27374: S 2616624640:2616624640(0) win 16384 <mss
1380,nop,nop,sackOK> (DF) (ttl 114, id 35852)
21:55:44.907528 my.host.27374 > 142.177.227.30.4865: R 0:0(0) ack 2616624641 win 0 (ttl 128, id 2641)
21:55:45.457993 142.177.227.30.4865 > my.host.27374: S 2367511908:2367511908(0) win 16384 <mss
1380,nop,nop,sackOK> (DF) (ttl 114, id 35865)
21:55:45.458076 my.host.27374 > 142.177.227.30.4865: R 0:0(0) ack 4045854565 win 0 (ttl 128, id 2642)
21:55:46.059972 142.177.227.30.4865 > my.host.27374: S 1464179629:1464179629(0) win 16384 <mss
1380,nop,nop,sackOK> (DF) (ttl 114, id 35882)
21:55:46.060054 my.host.27374 > 142.177.227.30.4865: R 0:0(0) ack 3142522286 win 0 (ttl 128, id 2643)
23:51:57.486243 142.177.225.131.4848 > my.host.27374: S 1164592741:1164592741(0) win 16384 <mss
1380,nop,nop,sackOK> (DF) (ttl 114, id 36168)
23:51:57.498905 my.host.27374 > 142.177.225.131.4848: R 0:0(0) ack 1164592742 win 0 (ttl 128, id
2658)
23:51:58.044089 142.177.225.131.4848 > my.host.27374: S 1214947453:1214947453(0) win 16384 <mss
1380,nop,nop,sackOK> (DF) (ttl 114, id 36181)
23:51:58.044170 my.host.27374 > 142.177.225.131.4848: R 0:0(0) ack 50354713 win 0 (ttl 128, id 2659)
23:51:58.544420 142.177.225.131.4848 > my.host.27374: S 1759036000:1759036000(0) win 16384 <mss
1380,nop,nop,sackOK> (DF) (ttl 114, id 36195)
23:51:58.544504 my.host.27374 > 142.177.225.131.4848: R 0:0(0) ack 594443260 win 0 (ttl 128, id 2660)
00:17:18.348603 24.43.214.15.4883 > my.host.27374: S 70729798:70729798(0) win 4288 <mss
1460,nop,nop,sackOK> (DF) (ttl 114, id 18209)
00:17:18.374997 my.host.27374 > 24.43.214.15.4883: R 0:0(0) ack 70729799 win 0 (ttl 128, id 2661)
00:17:18.900952 24.43.214.15.4883 > my.host.27374: S 70729798:70729798(0) win 4288 <mss
1460,nop,nop,sackOK> (DF) (ttl 114, id 18240)
00:17:18.901034 my.host.27374 > 24.43.214.15.4883: R 0:0(0) ack 1 win 0 (ttl 128, id 2662)
00:17:19.403379 24.43.214.15.4883 > my.host.27374: S 70729798:70729798(0) win 4288 <mss
1460,nop,nop,sackOK> (DF) (ttl 114, id 18246)
00:17:19.403461 my.host.27374 > 24.43.214.15.4883: R 0:0(0) ack 1 win 0 (ttl 128, id 2663)
01:10:07.245105 172.163.200.78.1813 > my.host.27374: S 21076316:21076316(0) win 8192 <mss
1432,nop,nop,sackOK> (DF) (ttl 106, id 14637)
01:10:07.264622 my.host.27374 > 172.163.200.78.1813: R 0:0(0) ack 21076317 win 0 (ttl 128, id 2665)
01:10:07.979938 172.163.200.78.1813 > my.host.27374: S 21076316:21076316(0) win 8192 <mss
1432,nop,nop,sackOK> (DF) (ttl 106, id 38189)
01:10:07.980019 my.host.27374 > 172.163.200.78.1813: R 0:0(0) ack 1 win 0 (ttl 128, id 2666)
01:10:08.722122 172.163.200.78.1813 > my.host.27374: S 21076316:21076316(0) win 8192 <mss
1432,nop,nop,sackOK> (DF) (ttl 106, id 45869)
01:10:08.722206 my.host.27374 > 172.163.200.78.1813: R 0:0(0) ack 1 win 0 (ttl 128, id 2667)
```

```
01:18:19.049485 172.131.42.123.3835 > my.host.27374: S 3862855:3862855(0) win 4288 <mss
536,nop,nop,sackOK> (DF) (ttl 107, id 44877)
01:18:19.049571 my.host.27374 > 172.131.42.123.3835: R 0:0(0) ack 3862856 win 0 (ttl 128, id 2668)
01:18:19.643002 172.131.42.123.3835 > my.host.27374: S 3862855:3862855(0) win 4288 <mss
536,nop,nop,sackOK> (DF) (ttl 107, id 58957)
01:18:19.643087 my.host.27374 > 172.131.42.123.3835: R 0:0(0) ack 1 win 0 (ttl 128, id 2669)
01:18:20.476554 172.131.42.123.3835 > my.host.27374: S 3862855:3862855(0) win 4288 <mss
536,nop,nop,sackOK> (DF) (ttl 107, id 64845)
01:18:20.476634 my.host.27374 > 172.131.42.123.3835: R 0:0(0) ack 1 win 0 (ttl 128, id 2670)
01:18:21.148908 172.131.42.123.3835 > my.host.27374: S 3862855:3862855(0) win 4288 <mss
536,nop,nop,sackOK> (DF) (ttl 107, id 6734)
01:18:21.148993 my.host.27374 > 172.131.42.123.3835: R 0:0(0) ack 1 win 0 (ttl 128, id 2671)
02:02:16.034356 24.232.121.53.1821 > my.host.27374: S 18392564:18392564(0) win 8192 <mss
1460,nop,nop,sackOK> (DF) (ttl 108, id 49207)
02:02:16.054779 my.host.27374 > 24.232.121.53.1821: R 0:0(0) ack 18392565 win 0 (ttl 128, id 2672)
02:02:16.684409 24.232.121.53.1821 > my.host.27374: S 18392564:18392564(0) win 8192 <mss
1460,nop,nop,sackOK> (DF) (ttl 108, id 51511)
02:02:16.684490 my.host.27374 > 24.232.121.53.1821: R 0:0(0) ack 1 win 0 (ttl 128, id 2674)
02:02:17.283632 24.232.121.53.1821 > my.host.27374: S 18392564:18392564(0) win 8192 <mss
1460,nop,nop,sackOK> (DF) (ttl 108, id 54583)
02:02:17.283708 my.host.27374 > 24.232.121.53.1821: R 0:0(0) ack 1 win 0 (ttl 128, id 2676)
02:02:17.883503 24.232.121.53.1821 > my.host.27374: S 18392564:18392564(0) win 8192 <mss
1460,nop,nop,sackOK> (DF) (ttl 108, id 59191)
02:02:17.883585 my.host.27374 > 24.232.121.53.1821: R 0:0(0) ack 1 win 0 (ttl 128, id 2678)
04:15:05.324616 4.54.37.238.2114 > my.host.27374: S 117551630:117551630(0) win 4288 <mss
536,nop,nop,sackOK> (DF) (ttl 113, id 61248)
04:15:05.339103 my.host.27374 > 4.54.37.238.2114: R 0:0(0) ack 117551631 win 0 (ttl 128, id 2696)
04:15:06.754302 4.54.37.238.2114 > my.host.27374: S 117551630:117551630(0) win 4288 <mss
536,nop,nop,sackOK> (DF) (ttl 113, id 10305)
04:15:06.754384 my.host.27374 > 4.54.37.238.2114: R 0:0(0) ack 1 win 0 (ttl 128, id 2697)
04:15:07.934107 4.54.37.238.2114 > my.host.27374: S 117551630:117551630(0) win 4288 <mss
536,nop,nop,sackOK> (DF) (ttl 113, id 13377)
04:15:07.934189 my.host.27374 > 4.54.37.238.2114: R 0:0(0) ack 1 win 0 (ttl 128, id 2698)
04:15:08.943916 4.54.37.238.2114 > my.host.27374: S 117551630:117551630(0) win 4288 <mss
536,nop,nop,sackOK> (DF) (ttl 113, id 27969)
04:15:08.943985 my.host.27374 > 4.54.37.238.2114: R 0:0(0) ack 1 win 0 (ttl 128, id 2699)
```

5.2. Source of trace

My pc connected to @HOME network via DSL.

5.3. Type of event generator

Windump 2.02.

Windump log description:

time **src_ip_addr.src_port** traffic_direction **dst_ip_addr.dst_port:** TCP_flags **TCP_sequence_no (payload)** reserved_word window_size <**TCP options**> (Don't fragment flag) **(time_to_live, IP packet id)**

Example:

04:15:08.943916 **4.54.37.238.2114** > **my.host.27374**: S **117551630:117551630(0)** win 4288 <**mss 536,nop,nop,sackOK**> (DF) (ttl 113, id 27969)

5.4. Probability the source address was spoofed

The probability of the source address being spoofed is small since its seems activity resulting from the w32-leave.worm. If activity is from a worm trying to spread to other systems, a TCP connection with a valid address must be completed for the spreading activity to be effective. If activity would be resulting from reconnaissance work, it would work with a spoofed IP address.

5.5. Description of attack

Multiple hosts scanned for port 27374 which is used by Subseven and Ramen Trojans.

Source addresses for this trace are:
142.177.227.30
142.177.225.131
24.43.214.15
172.163.200.78
172.131.42.123
24.232.121.53
4.54.37.238

My host sent a RST to all SYN packets received from source addresses listed above.

5.6. Attack mechanism

W32-leave.worm spreads using Subseven 2.1 port 27374 if host has been infected with Subseven trojan. This new worm deletes some files from the infected host, installs itself as a service and scans specific hosts and netblock addresses, such as @HOME. Scanning allows it to find other host infected with Subseven trojan to spread and infect other hosts.

5.7. Correlation

This trace was captured on July 4th. This is a lot of activity for one holiday!

An article on Hackinthebox.org website from 6/27/01 cites:
"Over the weekend we have been working to analyze a new MS Windows worm named W32.leave.worm. Although the ultimate intent of this worm has not yet been discovered, there are indications that it may be used as part of Zombie DDoS agents. Network traffic collected by the Internet Storm Center and its partners indicates that there is widespread activity. It is assumed that the worms ability to synchronize the system time, to download additional code, and to listen to IRC channels make it a very dangerous DDOS tool.

DESCRIPTION
This program propagates itself via connections to Sub7 port 27374. It then issues the default Sub7 password, and if successful, tells the computer to download and execute f.exe from l4l4l4l4.spites.com (which has been shutdown). After executing, it does several things:…"

The article also adds:

“Finally, it starts scanning port 27374 within predetermined netblocks associated with @Home and Earthlink. It also connects to IRC, creates a random name, and connects to a predetermined channel and waits.” This would explain various source addresses scanning my host connected to @HOME network.

More information at:
http://www.hackinthebox.org/print.php?sid=2434

Advisory 01-014 was issued on NIPC (www.nipc.gov) relating to w32-leave.worm and port 27374 activity:
http://www.nipc.gov/warnings/advisories/2001/01-014.htm

Correlation to targeted block addresses, using whois service from www.arin.net:
Source addresses
Netblock for ip 142.177.227.30 and 142.177.225.131:

```
Stentor National Integrated Communications Network
NET-STENTOR19)
   410 Laurier Avenue West, Room 730
   Ottawa, ON K1P6H5
   CA
   Netname: STENTOR19
   Netblock: 142.177.0.0 - 142.177.255.255
```

Netblock for ip 24.43.214.15:

```
Rogers@Home (NETBLK-ROGERS-4-BLOCK)
ROGERS-4-BLOCK   24.42.0.0 - 24.43.255.255
Rogers@Home Wlfdle (NETBLK-ON-ROG-23-2WLFDLE-1)
ON-ROG-23-2WLFDLE-1
24.43.214.0 - 24.43.214.255
```

Netblock for ip 172.163.200.78 and 172.131.42.123:

```
America Online, Inc. (NETBLK-AOL-172BLK)
   12100 Sunrise Valley Drive
   Reston, VA 20191
   US
   Netname: AOL-172BLK
   Netblock: 172.128.0.0 - 172.191.255.255
```

Netblock for 24.232.121.53:

```
CABLEVISION S.A. (NETBLK-CVTCI-BLK-1)
CVTCI-BLK-1  24.232.0.0 - 24.232.191.255
CABLEVISION S.A. (NETBLK-MOTOROLA-SANFERNANDO)
MOTOROLA-SANFERNANDO
24.232.120.0 - 24.232.124.255
```

Netblock for 4.54.37.238:

```
BBN Planet (NET-SATNET)
   150 Cambridge Park Dr.
   Cambridge, MA 02138
   US
   Netname: SATNET
   Netblock: 4.0.0.0 - 4.255.255.255
```

5.8. Evidence of active targeting

Because my host's connection to the ISP's network does not allow to collect packets besides those emanating from or destined to my.host, the trace is limited and does not show reconnaissance activity targeted to other hosts.

The host was not specifically targeted, but the netblock address was targeted, per article cited above.

5.9. Severity

(Criticality + Lethality) – (System + Network Countermeasures) = Severity
(5 + 5) - (5 + 0) = 5
System Criticality = 5 (High because this worm uses any host already infected by Subseven trojan as a launch pad to infect other hosts.)
Attack Lethality = 5 (This worm spreads quickly over the Internet and sources cite that removal is difficult. At least, seven source addresses infected over 6 hour period.)
System Countermeasures = 5 (My host has updated anti-virus signatures, was not infected with Subseven Trojan and does not have port 27374 opened.)
Network Countermeasures = 0 (The network does not block activity destined to this port.)

5.10. Defensive Recommendations

Enable personal firewall software on hosts connected to Internet via DSL and update anti-virus software with latest signatures.

5.11. Multiple choice test question

This trace was suspicious because:
a) it used a reserved port as the destination port
b) it had many source addresses in a relative short period of time
c) the host was infected with Subseven Trojan
d) port 27374 is used by various scanning tools

Answer B

6. Detect 6

6.1. Network Trace

```
Jul 24 00:47:36 128.143.47.11:21 -> a.b.c.11:21 SYN ******S*
Jul 24 00:47:36 128.143.47.11:21 -> a.b.c.27:21 SYN ******S*
Jul 24 00:47:36 128.143.47.11:21 -> a.b.c.51:21 SYN ******S*
Jul 24 00:47:36 128.143.47.11:3449 -> a.b.c.62:21 SYN ******S*
Jul 24 00:47:36 128.143.47.11:21 -> a.b.c.67:21 SYN ******S*
Jul 24 00:47:36 128.143.47.11:21 -> a.b.c.71:21 SYN ******S*
Jul 24 00:47:36 128.143.47.11:21 -> a.b.c.82:21 SYN ******S*
Jul 24 00:47:36 128.143.47.11:21 -> a.b.c.101:21 SYN ******S*
Jul 24 00:47:36 128.143.47.11:21 -> a.b.c.121:21 SYN ******S*
Jul 24 00:47:36 128.143.47.11:21 -> a.b.c.182:21 SYN ******S*
Jul 24 00:47:36 128.143.47.11:21 -> a.b.c.192:21 SYN ******S*
Jul 24 00:47:36 128.143.47.11:21 -> a.b.c.194:21 SYN ******S*
Jul 24 00:47:36 128.143.47.11:21 -> a.b.c.195:21 SYN ******S*
Jul 24 00:47:36 128.143.47.11:21 -> a.b.c.212:21 SYN ******S*
Jul 24 00:47:36 128.143.47.11:21 -> a.b.c.237:21 SYN ******S*
Jul 24 00:47:36 128.143.47.11:21 -> a.b.d.52:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.d.215:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.d.221:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.d.222:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.d.228:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.d.233:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.d.250:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.d.253:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.d.254:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.e.42:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.e.79:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.e.88:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.e.97:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.e.101:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.e.107:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.e.116:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.e.128:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.e.164:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.e.175:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.e.176:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.e.179:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.e.184:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.e.195:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.e.217:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.e.219:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.e.229:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.e.233:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.f.10:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.f.14:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.f.18:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.f.21:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.f.30:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.f.32:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.f.39:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.f.41:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.f.54:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.f.79:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.f.128:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.f.133:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.f.145:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.f.149:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.f.163:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.f.165:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.f.164:21 SYN ******S*
Jul 24 00:47:37 128.143.47.11:21 -> a.b.f.183:21 SYN ******S*
Jul 24 00:47:38 128.143.47.11:21 -> a.b.f.190:21 SYN ******S*
Jul 24 00:47:38 128.143.47.11:21 -> a.b.f.192:21 SYN ******S*
Jul 24 00:47:38 128.143.47.11:21 -> a.b.f.248:21 SYN ******S*
```

Additional information supplied with the trace:

```
Server used for this query: [ whois.arin.net ]
        University of Virginia (NET-VIRGINIA)
        Charlottesville, VA 22903 US
        Netname: VIRGINIA
        Netblock: 128.143.0.0 - 128.143.255.255
```

6.2. Source of trace

E-mail sent from Laurie Zirkle to intrusions@incidents.org mailing list on 6/25/01 with Subject: June 24, 2001 probes

6.3. Type of event generator

It is assume that log was generated by Snort.

Snort log format description:
month day time **src_addr:src_port** traffic_direction **dst_addr:dst_port** message_type tcp_flags

Example:

```
Jul 24 00:47:38 128.143.47.11:21 -> a.b.f.248:21 SYN ******S*
```

6.4. Probability the source address was spoofed

The probability of the source IP address being spoofed is small, but the system may have been compromised.

6.5. Description of attack

Source IP address 128.143.47.11 sent TCP packets with the SYN flag from reserved port 21 to port 21 on various host on the submitting analyst's network. The fourth line on the trace shows ephemeral port 3449 also sending a TCP packet with the SYN flag to destination port 21 on host a.b.c.62 .

Packets were received in a very short time as shown by the recorded time.

No other log activity was provided.

6.6. Attack mechanism

The trace shows only incoming traffic to the targeted network. All traffic is stimulus from the same source IP address. The targeted network may have multiple connections to the Internet for load-balancing or redundancy. Not knowing sensor placement on the targeted network and if the trace shows all incoming traffic, it is possible that reconnaissance traffic was sent to destination IP a.b.c.62, and to other IP addresses, but not recorded on this trace. Therefore, the fourth line of this trace may have been stimulus generated after a response from host a.b.c.62 on port 21 was sent back to IP 128.143.47.11.

6.7. Correlation

Assuming that Ms. Zirkle always replaces the octects with the same letter values, various network traces submitted to the same mailing list by Laurie Zirkle show host a.b.c.62 as one targeted for various services. Because the network trace only shows one side of the traffic, it is not certain if the response generated by the host were sent back to the attacker, if any.

FTP port 21 is a port commonly used as target for attacks. Port 21 shows as one of the top ten destination ports in www.incidents.org.

Port 21 has many known vulnerabilities which can explain that the source port used is a reserve port, port 21 in this case. Compromised systems could use this port to initiate other attacks and reconnaissance activity. Common Vulnerabilitities and Exposures related to the FTP service:

CVE-1999-0017: FTP servers can allow an attacker to connect to arbitrary ports on machines other than the FTP client, aka FTP bounce.

CVE-1999-0080: wu-ftp FTP server allows root access via "site exec" command.

CVE-1999-0082: CWD ~root command in ftpd allows root access.

CVE-1999-0879: Buffer overflow in WU-FTPD and related FTP servers allows remote attackers to gain root privileges via macro variables in a message file.

Bugtraq ID 126: Multiple Vendor FTP Bounce Attack Vulnerability.

Other CERT Advisories related to FTP vulnerabilities:
http://www.cert.org/advisories/CA-1997-27.html

More information at:
http://www.networkice.com/advice/Exploits/Services/FTP/default.htm

6.8. Evidence of active targeting

A wide range of hosts on the network was targeted. Port 21 was targeted. Host a.b.c.62 was targeted with different traffic to possibly exploit FTP vulnerability.

6.9. Severity

(Criticality + Lethality) – (System + Network Countermeasures) = Severity
(5 + 4) - (3 + 0) = 5

System Criticality = 5 (Attacker targeted server possibly running FTP or with port 21 opened, if correlation assumptions are correct.)
Attack Lethality = 4 (This was reconnaissance activity, still it targeted the FTP service which has known vulnerabilities.)
System Countermeasures = 3 (Don't know what was blocked by the host since no syslog file was provided with the trace.)

Network Countermeasures = 0 (Packet reached the host.)

6.10. Defensive Recommendations

Block all TCP connections to port 21 unless they are destined to the FTP servers located in the DMZ.

6.11. Multiple choice test question

Correlation analysis can be improved by:
a) analyzing various logs depicting traffic for different TCP activity
b) analyzing various logs depicting traffic for the same TCP activity
c) analyzing only one log depicting event of interest
d) analyzing multiple logs from the corporate firewall compared to security policies

Answer B

<u>Assignment 2 – The State of Intrution Detection</u>

IDS Security with Managed Security Services

Outsourcing has become a popular solution to satisfying technical and security needs for government and commercial organizations. In the security arena, this has resulted from increased importance and demand in security while organizations' technical expertise and resources are focused in other initiatives. Government downsizing and the inability to offer competitive salaries to highly skilled personnel has even influenced government agencies to seek outsourcing solutions. The emerge of Managed Security Services (MSS) as the outsourced solution for some of the organization's IT security needs raise security risks and concerns that need to be addressed and understood by organizations seeking such services.

The basis of security is the protection of confidentiality, integrity and availability (CIA) of information, data and systems. With this in mind, an organization seeking MSS as its IDS solution needs to understand each aspect of how CIA is implemented for its intrusion detection system (IDS).

<u>Confidentiality</u>

Confidentiality relates to secrecy of information and the protection of such information. Therefore, contracting a Managed Security Services Provider (MSSP) company to manage a layer of the organization's defense in-depth strategy, such as the IDS implementation, results in the MSSP learning about many of the organization's technical and operational information: policies, network infrastructure, business functions, applications and hosts housing critical components. A non-disclosure agreement signed by the MSSP is the first step to protect the confidentiality of the organization's critical information shared with the MSSP to deliver services for which it has been contracted. The non-disclosure agreement can also contain a clause that requires the information about the organization being part of the MSSP's customer base to be kept as a confidential as well, therefore keeping some inherent or future weakness supplied by the MSSP and the IDS implementation from public knowledge. Contracts from the Department of Defense are handled in similar manner as they may handle data and information that relates to national security.

Identifying personnel to act as key points of contact for the organization and the MSSP will provide protection against clever social engineering reconnaissance tactics as well as provide common sense operational control. Key personnel is critical during the information-gathering phase and the continuing information sharing with the MSSP on the organization's infrastructure changes as well as intrusion incidents handling. The MSSP must have good personnel screening practices. Some MSSP maintain personnel with Secret and Top Secret clearances and perform thorough background checks. The organization may insist on contracting with a MSSP with a whitehat-only hiring policy, which should not be hard to find since most reputable and/or emerging companies enroll good technicians without hiring from the "black-side of the Force."

Now let's consider the sensors' confidentiality, specifically network-based sensors, since sensors are the focal point of automated information gathering for the IDS system and they are also nodes on the network. There are sniffer-detection tools available to detect a computer running in promiscuous mode, capturing packets not destined to its hardware address, in this case, the network-based sensor. If a hacker was to compromise a computer on the same network as the sensor, he/she could run a sniffer-detection tool from the compromised system to detect other systems running in promiscuous mode in an effort to find IDS sensors and to further target the IDS. Hacker ingenuity can not be underestimated. Network-based sensors should have two network interfaces: one with no IP address and no IP stack which makes it invisible to sniffer-detection tools; another to communicate with the IDS console where data is gathered from multiple sensors on the organization's network before is transferred to the MSSP' network and databases. As an aside, sniffer-detection tools can work to the security specialist's advantage when conducting the organization's vulnerability assessment by detecting unauthorized network sniffing activity.

Communications between the IDS console and the MSSP's network must be secured since this traffic travels over the Internet, in most cases, to avoid the additional expense of dedicated/leased lines. A MSSP will support a VPN solution to provide authentication and encryption to secure the confidentiality of IDS traffic between organization and the MSSP. Automated alert such as Simple Network Management Protocol (SNMP) console alerts should also be transmitted over a secure channel. MSSPs offer web portals to provide the organization with incident notification and statistical information reporting which are secured with Secure Socket Layer (SSL) encryption. Authenticated access to web portal is achieved with via strong passwords or e-token, which the MSSP must also manage.

Outsourcing the organization's IDS solution implies a database store at the MSSP's site or the Security Operations Center (SOC), where data normalization, data mining and analysis take place. Therefore the MSSP shall ensure the organization's database confidentiality, separating it from other customers' data, besides assuring its integrity and availability for real-time analysis. In addition, the MSSP must have strong physical security to their SOC, with a minimum of two-level personnel authentication for access control, including biometrics.

Integrity

The aspect of security that is most cumbersome in an IDS implementation is integrity. Most integrity concerns are present whether the IDS is managed by a MSSP or by the organization's in-house resources. Accuracy of the IDS operational goals can be evaluated by how does it collect and analyze the collected data:

- ◊ Can the IDS capture all the data accurately?
  - o Can it capture all the packets?
  - o Can it capture packets destined to other network segments as well at its own?
- ◊ Can it gather data on a centralized database?
  - o Can it transmit sensor data accurately?
  - o Can it combine data from multiple sensors?
  - o Can it store data for long-term analysis?
- ◊ Can it analyze the data collected and identify malicious events?
  - o Can it normalize the collected data so that traffic is only counted once?
  - o Can it differentiate between IDS traffic and other traffic?
  - o Can it verify that traffic/activity is allowed by its destination, either a network node or an application within the host?
  - o Can it match collected data to known malicious or suspicious patterns and behavior?
  - o Can it "learn" about new patterns or malicious activity in a timely fashion?
  - o Can it use data mining for correlation of events beyond real-time or short-term pattern matching?
- ◊ Can analysts provide accurate analysis beyond the systems capabilities?
- ◊ Can it alert accurately about malicious events?
- ◊ Can it report to the interested parties on a timely fashion?

The primary objective of the sensor is to capture all events so IDS components and analysis can be focused on those of interest. Host events are mostly self-contained: if the traffic can reach the host, it can be logged. Network traffic, on the other hand, presents different challenges since changes in network technology and design have a dramatic impact on what the network sensor can capture. Faster network speed means that network traffic needs to be captured at a faster rate. Switched network technology affects the IDS design implementation because all traffic on the network segment must be captured and not only the one destined to the sensor's interface. The MSSP must provide a technical solution that is suitable to the organization's network infrastructure, especially if using high-speed networks and switching technology, and flexibility to revise the IDS design and refresh the implemented technology to achieve accurate and adequate performance levels as the network infrastructure and technical needs change.

IDS evasion and insertion techniques threaten the quantity and quality of data recorded by network IDS components. At the heart of the problem is the disparity of what is an acceptable packet to the IDS versus what is an acceptable to the destination host, which depends on its implemented TCP/IP stack. Operating system developers do not make this job easier by designing variances of TCP/IP stacks, therefore increasing the gap between the IDS packet interpretation and the host's. If the MSSP also manages the organization's border gateways and stateful devices, this would aid in the interpretation of IDS traffic by integrating and correlating gathered data to provide a whole picture view. Encryption also affects IDS analysis since header information, but not the packet payload, can be analyzed. In addition, TCP flags can be set by traffic relating to other emerging network standards, such as Explicit Network Congestion (ECN), which can confuse the IDS signature matching and analysis creating more false positives.

The lag time from when a vulnerability is targeted and exploited and when a signature is developed for its accurate detection is especially critical before an effective countermeasure is developed and deployed. Many MSSP may favor one IDS vendor over another, even when it considers itself to be vendor independent, because it has in-house developers to design/code signatures for newly discovered attacks in an attempt to narrow the detection gap. This value-added develops from MSSP partnerships with IDS vendor companies. The MSSP must deploy a signature quickly to offer maximum detection and protection. Also, the MSSP's support for products from multiple vendors can be in the organization's best interest if a vulnerability was to be found for any IDS component already deployed in the organization.

Even thought the organization will not have control over the MSSPs internal process for data analysis, still should inquire about it and understand it at a high level. This includes the process for sensor data normalization and correlation analysis and value added by the MSSP's research and knowledge base, if any.

Integrity of malicious event identification goes beyond the computer system to include the human factor: the staff' experience level. The MSSP must provide around-the-clock coverage at the SOC and experienced analysts/engineers available to assist less experienced analysts for event identification. The MSSP must also have training initiatives to keep staff updated with technology. This is critical since the analyst will ascertain the validity of alerts generated at the SOC's console and perform correlation of events with other sites or activity. Some MSSPs will offer a global view of security threats because of their customer base or association with other organizations beyond the national boundaries.

Incident handling procedures are the next step to accurate event identification, to make sure the appropriate staff is notified timely for appropriate course of action. MSSPs can provide further vulnerability analysis and countermeasures information via their web-portal reporting to assist the organization.

Physical security must be in place at the MSSP and the organization. Background checks must be performed by the MSSP, as mentioned before, and physical access controls and activity logs of SOC activities should be in place. The integrity of sensor themselves placed in the organization's network is accomplished by system hardening and running the IDS software from the CD-Rom to minimize a system compromise.

Availability

Availability of IDS monitoring services start from sensors and the console, to local network and Internet connectivity, and the MSSP's SOC uninterrupted operations. Redundancy of resources need to be in place at the organization's as well as the MSSP's SOC for the continuous 24x7 IDS operations, analysis and alerts. The SOC must be built with redundancy for continued services and operations on its present location or an alternate site in case of a natural disaster, loss of power and physical compromise. There must be redundant Internet connectivity via different Internet Service Providers (ISPs) for the organization and the SOC. Sensors and the local console, or centralized collection point shall continue to collect data even when connectivity to the SOC is interrupted so that it can be synchronized when connectivity resumes. Part of the contingency planning shall include procedures and prior training to the organization's personnel, even if limited, in case of sensor malfunction and remote sensor management failure.

The MSSP must have financial stability for continued service to the company and future expansion of services, if desired. The MSSP must guarantee a level of staff availability to in support of services provided to the organization as well.

Final Thoughts

Organizations seek outsourcing solutions to reduce overhead and improve coverage and quality of service for the deployment, monitoring and management their intrusion detection system. Outsourcing security services introduce security risks to the organization.

Most articles used as source for this assignment raise "red flags" about outsourcing security at any level. Still, the organization has some possible countermeasures to the risks involve by evaluating the MSSP's CIA protection capabilities and by finding a MSSP that fits the organization's security requirements in addition to satisfying operational goals. Reputable MSSPs have built confidentiality, integrity and availability safeguards to provide a secured IDS solution to organizations that may otherwise not be able to implement.

References:

Counterpane Internet Security; "Managed Security Monitoring"; Webpage on company's website; paper accessed on March 1, 2001, but no longer available.
url: www.counterpane.com/whitepaper.html

Harreld, Heather; "Outsourcing opens security risks"; Federal Computer Week; January 5, 1998 url: http://208.201.97.5/pubs/fcw/1998/0105/fcw-risks-1-5-1998.html

Hulme, George V.; "Use Caution When Choosing A Managed Security Vendor"; InformationWeek; July 16, 2001.
url: http://www.informationweek.com/thisweek/story/IWK20010713S0006

Internet Security Systems; "How to Select a Managed Security Provider"; April 2001.
url: http://documents.iss.net/whitepapers/MSP.pdf

Internet Security Systems; "Intrusion Detection for the Millennium."
url: http://documents.iss.net/whitepapers/int_detect.pdf

McClure, Stuart and Scambray, Joel; "Once-promising intrusion detection systems stumble over switched networks"; InfoWorld; December 11, 2000.
url: http://www.inquiry.com/pubs/infoworld/vol22/issue50/001211opswatch.asp

Ploskina, Brian; "Managed Security Deals Leave Networks Vulnerable"; Interactive Week; July 9, 2001.
url: http://www.zdnet.com/intweek/stories/news/0,4164,2783606,00.html

Miller, Toby; "ECN and it's impact on Intrusion Detection"; The SANS Institute; May 31, 2001.
url: http://www.incidents.org/detect/ecn.php

Security Software Technologies; "The Goal and Purpose of AntiSniff."
url: http://www.securitysoftwaretech.com/antisniff/purpose.html

Tuesday, Vince; "Security Outsourcing: Don't Bet on It – Yet"; ComputerWorld; June 11, 2001.
url: http://www.computerworld.com/cwi/story/0,1199,NAV47_STO61232,00.html

Assignment 3 – Analyze This Scenario

7. Alerts Analysis

7.1. Total number of alerts by type

No. of alerts	Alert Signature
148246	UDP SRC and DST outside network
5388	Watchlist 000222 NET-NCFC
3779	Possible RAMEN server activity
3702	Watchlist 000220 IL-ISDNNET-990517
1112	SYN-FIN scan!
590	connect to 515 from inside
507	Attempted Sun RPC high port access
210	Queso fingerprint
191	WinGate 1080 Attempt
111	Tiny Fragments - Possible Hostile Activity
72	Null scan!
60	TCP SRC and DST outside network
19	ICMP SRC and DST outside network
12	NMAP TCP ping!
5	SNMP public access
4	TCP SMTP Source Port traffic
4	SUNRPC highport access!
1	Russia Dynamo - SANS Flash 28-jul-00

7.2. Top destination addresses with alert activity

No. of alerts	Destination address
143275	224.2.127.254
5337	MY.NET.6.47
2186	MY.NET.207.226
1133	224.0.1.41
1074	24.48.226.183
605	169.254.255.255
573	216.181.129.185
387	162.129.112.40
362	MY.NET.223.254
354	233.28.65.242
321	MY.NET.222.94
273	MY.NET.204.22
260	MY.NET.224.34
232	193.231.10.13
208	MY.NET.217.98
201	224.0.1.1
179	233.28.65.50
176	172.16.1.103
162	MY.NET.211.74
152	MY.NET.225.186
135	MY.NET.221.246

7.3. Top source addresses involved with alert activity

No. of alerts	Source address
31105	155.101.21.38
13779	130.235.133.92
10950	171.69.248.71
7678	129.116.65.3
6734	171.68.98.109
6715	128.223.83.33
6559	128.249.104.243
6487	128.249.104.246
6401	130.161.180.141
5894	171.68.43.192
5603	130.240.64.20
5481	152.1.1.79
5362	159.226.81.1
4070	140.142.19.72
3196	128.223.83.35
3091	130.225.127.87
2186	212.179.21.179
1819	24.48.226.183
1366	63.105.122.6
1302	129.89.125.91
1169	130.240.4.100
1133	171.69.33.40
1108	211.248.112.67
1087	128.178.10.2

7.4. Single alert analysis

7.4.1. UDP SRC and DST outside network

Top source addresses:

No. of alerts	Destination address
31105	155.101.21.38
13779	130.235.133.92
10950	171.69.248.71
7678	129.116.65.3
6734	171.68.98.109
6715	128.223.83.33
6559	128.249.104.243
6487	128.249.104.246
6401	130.161.180.141
5894	171.68.43.192
5603	130.240.64.20
5481	152.1.1.79
4070	140.142.19.72
3196	128.223.83.35
3091	130.225.127.87

Top destination addresses:

No. of alerts	Source address
143270	224.2.127.254
1133	224.0.1.41
605	169.254.255.255
387	162.129.112.40
354	233.28.65.242
232	193.231.10.13
201	224.0.1.1
179	233.28.65.50
176	172.16.1.103
72	10.1.11.101
67	5.0.0.4
55	24.3.0.37
45	24.3.0.38
42	164.124.101.2

7.4.2. Watchlist 000220 IL-ISDNNET-990517

All Watchlist 000220 IL-ISDNNET-990517 alerts were generated from ISP netblock in Israel:

Subnets are assigned to different organizations, for example:

inetnum: 212.179.21.160 - 212.179.21.191
netname: MEGIDO
descr: MEGIDO-LAN
country: IL

inetnum: 212.179.41.128 - 212.179.41.255
netname: KIBUTZ-GEVA
descr: Kibutz-Geva-LAN
country: IL
source: RIPE

Top source addresses:

2186	212.179.21.179
321	212.179.42.21
277	212.179.79.2
272	212.179.47.83
260	212.179.58.193
152	212.179.40.132
133	212.179.28.66
81	212.179.27.6
15	212.179.41.220

Top destination addresses

No. of alerts	Destination address
2186	MY.NET.207.226
321	MY.NET.222.94

272	MY.NET.204.22
260	MY.NET.224.34
207	MY.NET.217.98
152	MY.NET.225.186
133	MY.NET.211.74
81	MY.NET.204.78
55	MY.NET.97.30
15	MY.NET.206.94
11	MY.NET.97.62

7.4.3. Watchlist 000222 NET-NCFC

All Watchlist 000222 NET-NCFC alerts, even the ones not listed on this table, were generated on the same netblock:

The Computer Network Center Chinese Academy of Sciences (NET-NCFC)
P.O. Box 2704-10,
Institute of Computing Technology Chinese Academy of Sciences
Beijing 100080, China
CN

Netname: NCFC
Netblock: 159.226.0.0 - 159.226.255.255

Top source addresses:

No. of alerts	Source address
5362	159.226.81.1
6	159.226.39.4
4	159.226.114.1
2	159.226.126.85
2	159.226.111.1

Top destination addresses:

No. of alerts	Destination address
5337	MY.NET.6.47
27	MY.NET.253.43
8	MY.NET.60.17
5	MY.NET.6.35
5	MY.NET.100.230

7.4.4. Possible RAMEN server activity

Ramen is a worm affecting Red Hat Linux 6.2 and 7.0. The Ramen worm is targeted to compromise web servers and self-propagate to other vulnerable systems by scanning on destination port 21 to infect other systems. Wu-ftpd site_exec() and rpc.statd exploits are used by the Ramen worm on Red Hat 6.2 which can enable an attacker to execute code as root, and the LPRng vulnerability is associated with Red Hat 7.0. The alerts analyzed here show port 27374 which is the port the worm uses to propagate itself.

Further information can be found at:
http://www.sans.org/infosecFAQ/malicious/ramen.htm
http://www.whitehats.com/library/worms/ramen/

Top external source addresses:

No. of alerts	Source address
1819	24.48.226.183
13	24.48.121.105
10	134.29.48.235
9	203.79.69.182
8	203.106.99.237
6	24.23.131.82
4	212.14.255.107

Top two external source addresses come from a ISP in Pennsylvania:

Adelphia Cable Communications (NETBLK-ADELPHIA-CABLE)
Main at Water Street
Coudersport, PA 16915
US
Netname: ADELPHIA-CABLE
Netblock: 24.48.0.0 - 24.51.255.255

Top internal source addresses. Number of alerts have been grouped by first three octects.

No. of alerts	Destination subnet
535	MY.NET.253
74	MY.NET.225
50	MY.NET.217
39	MY.NET.97
37	MY.NET.209
36	MY.NET.224
35	MY.NET.207
32	MY.NET.223
31	MY.NET.227

Top destination addresses:

No. of alerts	Destination address
1074	24.48.226.183
36	MY.NET.225.66
22	MY.NET.217.202
15	24.48.121.105
10	24.23.131.82
10	203.79.69.182
10	203.106.99.237
9	64.231.218.26
8	MY.NET.202.222
7	MY.NET.60.17

7.4.5. SYN-FIN scan!

SYN and FIN flags sent on a single TCP segment can be used to fingerprint the receiving operating system as a predecessor to more targeted attacks. The tables below show source address 211.248.112.67 scanning many hosts on MY.NET on port 53 (DNS). Port 53 is possibly targeted because traffic to and from that port is sometimes not blocked at the border gateways and has known vulnerabilities that the attacker can further attempt to exploit it the port is opened on an active host.

Top source addresses:

No. of alerts	Source address
1108	211.248.112.67
1	63.252.15.242
1	4.35.4.244
1	24.50.25.5
1	209.255.180.130

Scanning source address comes from South Korea:

```
inetnum           211.232.0.0 - 211.255.255.255
netname           KRNIC-KR
descr             KRNIC
descr             Korea Network Information Center
country           KR
```

Top destination addresses. Number of alerts have been grouped by first three octects.

No. of alerts	Destination subnet
23	MY.NET.165
23	MY.NET.163
22	MY.NET.156
21	MY.NET.162
20	MY.NET.161
19	MY.NET.21
18	MY.NET.9
18	MY.NET.152
17	MY.NET.170
17	MY.NET.12
16	MY.NET.2
16	MY.NET.18
16	MY.NET.159
16	MY.NET.158
16	MY.NET.154

These scans seem evenly spread to addresses at MY.NET. Variances in total number of alerts calculated per IP address down to the three octet could be explained by number of active hosts at the time of the scan.

Ports targeted on SYN-FIN scans:

No. of alerts	Destination port
1108	53 (DNS)
1	6346 (Ephemeral port)
1	443 (SSL)

1	259 (ESRO)
1	1415 (Ephemeral port)

7.4.6. connect to 515 from inside

LPR service runs on Unix systems TCP port 515. There are known vulnerabilities that could result in root access to a compromised system or a denial of service.

SANS published an alert on Nov 2000 due to increasing probes to TCP port 515:
http://www.sans.org/newlook/alerts/port515.htm

CVEs and advisories related to LPR service vulnerabilities:
CAN-2000-0917: Format string vulnerability in use_syslog() function in LPRng 3.6.24 allows remote attackers to execute arbitrary commands.
CAN-2000-0839: WinCOM LPD 1.00.90 allows remote attackers to cause a denial of service by sending a large number of LPD options to the LPD port (515).
CVE-1999-0032: Buffer overflow in BSD-based lpr package allows local users to gain root privileges.
CVE-1999-0335: Buffer overflow in BSD and linux lpr command allows local users to execute commands as root through the classification option.
CA-2000-22: Input Validation Problems in LPRng.

Top source addresses:

No. of alerts	Source address
514	MY.NET.98.190
59	MY.NET.97.88
15	MY.NET.7.20
1	MY.NET.201.170
1	MY.NET.162.71

Top destination addresses:

No. of alerts	Destination address
573	216.181.129.185
15	216.88.97.58
1	209.50.66.2
1	209.249.182.79

7.4.7. Attempted Sun RPC high port access

Because of RPC services known vulnerabilities, related RPC ports (32770-34000 range) can be targets to scans and exploits. Two common scanned ports related to RPC services are 32771 and 32776, corresponding to rpc6 and rpc15, respectively.

The files analyzed here show port 32771 targeted in all attempted RPC access.

Top source addresses:

No. of alerts	Source address
362	64.244.10.40

134	205.188.153.97
6	205.188.153.108
5	205.188.153.107

Top destination addresses:

No. of alerts	Destination address
362	MY.NET.223.254
134	MY.NET.221.246
6	MY.NET.105.115
5	MY.NET.97.217

7.4.8. Queso fingerprint

Queso is a port scanning tool used for OS fingerprinting. It allows the attacker to spoof its address and to choose ports to scan.

For more information:
http://www.securityfocus.com/frames/?focus=ids&content=/focus/ids/articles/portscan.html

There has been some discussion on the two reserved bits in the TCP header, used by Queso as part of the OS fingerprinting activity and the proposed ECN (Explicit Congestion Notification) traffic, as published in the SANS' website:
http://www.sans.org/y2k/ecn.htm

Top source addresses:

No. of alerts	Source address
167	141.30.228. netblock
10	209.85.60.179
7	134.109.185.77
6	204.42.254.5
4	209.85.37
4	207.96.122
3	194.87.39
2	194.154.201

The files studied here show netblock 141.30.x.x with the highest indicents of this alert type. This netblock is assigned to an university in Germany:

inetnum: 141.30.0.0 - 141.30.255.255
netname: TUDR
descr: Technische Universitaet Dresden
country: DE
source: RIPE

Top destination addresses:

No. of alerts	Destination address
25	MY.NET.203.50
20	MY.NET.206.30
19	MY.NET.211.74

15	MY.NET.229.22
10	MY.NET.220.14
10	MY.NET.208.90
9	MY.NET.202.158

Top targeted ports for this alert type:

No of alerts	Port number
119	6346 (Gnutella)
53	6355 (ephemeral)
9	25 (SMTP)
5	113 (idend)
2	5500 (securid/Hotline)
2	12506 (ephemeral)

Also port 1 (tcpmux) was targeted with only one recorded alert.

7.4.9. WinGate 1080 Attempt

Wingate, a proxy service for Windows computers, run on port 1080. There are known vulnerabilities related to this service:

Bugtraq ID 509: WinGate's Winsock redirector service is susceptible to a buffer overflow vilnerability that will crash all WinGate services.

CVE-1999-0290: The WinGate telnet proxy allows remote attackers to cause a denial of service via a large number of connections to localhost.
CVE-1999-0291: The WinGate proxy is installed without a password, which allows remote attackers to redirect connections without authentication.
CVE-1999-0441: Remote attackers can perform a denial of service in WinGate machines using a buffer overflow in the Winsock Redirector Service
CVE-1999-0494: Denial of service in WinGate proxy through a buffer overflow in POP3.

Source addresses:

No. of alerts	Source address
29	24.1.201.200
21	128.121.244.217
18	199.173.178.2
15	216.179.0.32
14	63.151.165.130
12	204.117.70.5
11	212.73.162.30
5	209.212.128.47
4	216.234.161.197

Destination addresses:

No. of alerts	Destination address
29	MY.NET.221.30
21	MY.NET.15.178
14	MY.NET.98.118
9	MY.NET.203.234

7	MY.NET.202.138
6	MY.NET.60.8
5	MY.NET.218.86
4	MY.NET.98.156
4	MY.NET.60.17

7.4.10. Tiny Fragments – Possible Hostile Activity

IP fragmentation is used by attackers as an IDS evasion technique. This traffic needs to be investigated, but may not be malicious.

Top source addresses:

No. of alerts	Source address
73	64.80.90.36
6	202.205.5.10
5	64.80.88.99
5	202.96.96.3
3	64.80.90.84
2	64.80.90.55
2	64.80.89.149
2	61.140.75.5
2	61.136.61.68
2	61.134.9.133
2	202.101.43.220
2	111.111.111.111
1	61.155.13.3
1	61.140.75.3
1	61.134.9.134
1	210.12.160.130
1	127.0.0.1

Last entry on table shows reserved IP address 127.0.0.1.

Top destination addresses:

No. of alerts	Destination addresses
53	MY.NET.98.117
20	MY.NET.97.231
16	MY.NET.1.8
7	MY.NET.1.10
5	MY.NET.206.254
5	MY.NET.160.109
3	MY.NET.20.10
1	MY.NET.228.10
1	MY.NET.206.58

7.4.11. Null scan!

Setting a different combination of TCP flags, as a reconnaissance technique, elicit response from a host with an opened targeted TCP port. In the case of null scanning, no TCP flags are set, and it is a stealth scanning techniques to avoid detection.

For more information:
Port scanning:
http://www.networkice.com/advice/Underground/Hacking/Methods/Technical/Port_Scan/default.htm

Top source addresses:

No. of alerts	Destination address
11	63.253.x.x netblock
2	24.9.203.188
2	24.180.66.185
2	24.17.73.154
2	128.40.224.18
1	66.27.9.70
1	65.2.140.248
1	65.0.74.188
1	64.48.75.35
1	64.48.75.1
1	64.48.239.17
1	64.48.221.224
1	63.91.244.71
1	63.91.237.227
1	63.91.234.62
1	63.91.222.118
1	63.255.0.30
1	62.59.138.146
1	62.29.70.109
1	62.180.210.55

UUNET Technologies, Inc. (NETBLK-UUNET63)
3060 Williams Drive, Suite 601
Fairfax, Virginia 22031
US

Netname: UUNET63
Netblock: 63.64.0.0 - 63.127.255.255

Internet Allegiance, Inc. (NETBLK-IALG-ALGX-3)
1950 Stemmons Freeway Suite 3026
Dallas, TX 75207
US

Netname: IALG-ALGX-3
Netblock: 64.48.0.0 - 64.48.255.255

SplitRock Services, Inc (NETBLK-SPLITROCK99)
8665 New Trails Drive
The, 77381
US

Netname: SPLITROCK99
Netblock: 63.252.0.0 - 63.255.255.255

Top destination addresses:

No. of alerts	Destination address
9	MY.NET.211.74
5	MY.NET.60.8
4	MY.NET.60.11
3	MY.NET.60.38
3	MY.NET.5.29
3	MY.NET.224.102
3	MY.NET.201.234
2	MY.NET.220.14
2	MY.NET.165.129

Top destination ports for all Null scan alerts:

No. of alerts	Port number
23	0 (reserved)
12	6346 (Gnutella)
5	6688 (Napster client)
5	21504 (unknown)
4	6144 (unknown)
2	900 (OMG Initial Refs)
2	8960 (unknown)
2	6699 (Napster)
2	427 (Service location protocol)
2	20545 (unknown)
2	17746 (unknown)

7.4.12. TCP SRC and DST outside network

Top source addresses:

No. of alerts	Source address
16	169.254.101.152
9	65.9.177.76
8	192.168.1.51 (reserved)
6	3.0.0.2
4	10.10.5.3 (reserved)
4	0.0.0.0 (reserved)
3	24.23.55.21

Top destination addresses:

No. of alerts	Destination address
6	3.0.0.2
5	208.184.216.22
4	205.188.48.122
3	64.12.24.32
3	24.216.130.185
3	205.188.48.123
3	10.31.226.10

7.4.13. ICMP SRC and DST outside network

Top source addresses:

No. of alerts	Source address
3	10.10.5.3 (reserved)
2	172.128.249.145
2	140.120.93.254
2	140.120.80.254
1	65.9.177.76
1	172.182.21.112
1	172.174.12.110
1	172.167.26.248
1	172.167.120.189
1	172.159.72.255
1	172.128.235.48
1	172.128.196.159
1	172.128.122.7
1	140.120.29.254

Some source addresses from AOL netblock:

```
America Online, Inc. (NETBLK-AOL-172BLK)
   12100 Sunrise Valley Drive
   Reston, VA 20191
   US
   Netname: AOL-172BLK
   Netblock: 172.128.0.0 - 172.191.255.255
```

Top destination addresses:

No. of alerts	Destination address
5	224.2.127.254
3	192.63.42.145
2	24.228.9.100
1	62.224.189.36
1	61.75.17.13
1	4.34.186.8
1	24.66.28.94
1	24.189.144.253
1	211.106.127.235
1	172.168.69.200
1	156.3.140.252
1	146.145.238.234

7.4.14. NMAP TCP ping!

NMAP is a scanning tool used for OS fingerprinting. Below port 53 is the most scanned port. DNS and SMTP are targeted because of their known vulnerabilities, for example:

CVE-1999-0010: Denial of Service vulnerability in BIND 8 Releases via maliciously formatted DNS messages.
CVE-1999-0024: DNS cache poisoning via BIND, by predictable query IDs.

CVE-1999-0274: Denial of service in Windows NT DNS servers through malicious packet which contains a response to a query that wasn't made.

For more information about NMAP:
http://www.insecure.org/nmap/nmap_doc.html

Top source addresses:

No. of alerts	Source address
4	63.119.91.2
4	192.102.197.234
1	208.5.219.131
1	2.2.2.2
1	194.133.58.129
1	12.40.36.194

Top destination addresses:

No. of alerts	Destination address
5	MY.NET.1.8
3	MY.NET.1.5
3	MY.NET.1.3
1	MY.NET.110.39

All destination port numbers:

No. of alerts	Port number
11	53 (DNS)
1	25 (SMTP)

7.4.15. SNMP public access

SNMP services may have the default configuration of "public" or "private" community string, which a remote system can query, change and use to gain control of the local system.

More information has been included on this practical's Assigment 1.

Top source addresses:

No. of alerts	Source address
3	MY.NET.70.42
2	MY.NET.111.156

All the alerts listed above were all targeted to destination address MY.NET.50.154 on port number 161.

7.4.16. TCP SMTP Source Port traffic

Top source addresses:

No. of alerts	Source address
1	200.251.185.30
1	195.211.49.18

1	17.135.218.56
1	11.125.218.156

Top destination addresses:

No of alerts	Destination address
2	MY.NET.60.17
1	MY.NET.158.238
1	MY.NET.139.54

Destination ports:

No. of alerts	Port number
1	979
1	399
1	274
1	1007

7.4.17. SUNRPC highport access!

Top source addresses:

No. of alerts	Source address
2	205.188.5.157
1	24.9.203.188
1	200.233.81.13

Top destination addresses:

No. of alerts	Destination address
2	MY.NET.98.227
1	MY.NET.60.17
1	MY.NET.165.129

All destination ports are 32771. Some SunOS machines run portmapper on this port, therefore the attempted access to this high port.

For more information:
http://www.networkice.com/advice/Exploits/Ports/32771/default.htm
http://www.jammed.com/~jwa/hacks/security/h_rpcinfo/00README
http://www-4.ibm.com/software/security/firewall/about/nsa.htm

7.4.18. Russia Dynamo – SANS Flash 28-jul-00

Only one occurrence of this event appears in the data analyzed with source and destination pair: MY.NET.203.50 – 194.87.6.79, respectively. Source port was 6346 which is associated with Gnutella.

7.5. Top alert counts by source-destination pair and alert type

No. of Alerts	Source-Destination address pair	Alert description
31105	155.101.21.38-224.2.127.254	UDP SRC and DST outside network
13779	130.235.133.92-224.2.127.254	UDP SRC and DST outside network
10950	171.69.248.71-224.2.127.254	UDP SRC and DST outside network

7678	129.116.65.3-224.2.127.254	UDP SRC and DST outside network
6734	171.68.98.109-224.2.127.254	UDP SRC and DST outside network
6715	128.223.83.33-224.2.127.254	UDP SRC and DST outside network
6559	128.249.104.243-224.2.127.254	UDP SRC and DST outside network
6487	128.249.104.246-224.2.127.254	UDP SRC and DST outside network
6401	130.161.180.141-224.2.127.254	UDP SRC and DST outside network
5894	171.68.43.192-224.2.127.254	UDP SRC and DST outside network
5603	130.240.64.20-224.2.127.254	UDP SRC and DST outside network
5481	152.1.1.79-224.2.127.254	UDP SRC and DST outside network
5337	159.226.81.1-MY.NET.6.47	Watchlist 000222 NET-NCFC
4070	140.142.19.72-224.2.127.254	UDP SRC and DST outside network
3196	128.223.83.35-224.2.127.254	UDP SRC and DST outside network
3091	130.225.127.87-224.2.127.254	UDP SRC and DST outside network
2186	212.179.21.179-MY.NET.207.226	Watchlist 000220 IL-ISDNNET-990517
1366	63.105.122.6-224.2.127.254	UDP SRC and DST outside network
1302	129.89.125.91-224.2.127.254	UDP SRC and DST outside network
1169	130.240.4.100-224.2.127.254	UDP SRC and DST outside network
1133	171.69.33.40-224.0.1.41	UDP SRC and DST outside network
1087	128.178.10.2-224.2.127.254	UDP SRC and DST outside network

8. Portscan Analysis

8.1. Number of portscans by type

No. of Scans	Scan Type
454374	UDP
54916	SYN
17165	SYNFIN
1186	NOACK
801	INVALIDACK
405	UNKNOWN
361	NULL
275	XMAS
270	VECNA
157	FIN
68	FULLXMAS
21	NMAPID
16	SPAU

8.2. Top external source addresses

No. of scans	Source address
12129	169.226.202.234
3322	65.9.212.74
3006	194.27.160.5
1948	24.240.136.245
1702	212.162.240.66
1377	130.161.38.55
1240	134.169.9.201
1189	210.98.83.145
1108	211.248.112.67
847	199.108.40.107

835	24.141.226.62
397	24.4.196.167
267	200.188.18.69
229	24.112.112.204
208	24.112.150.147
184	202.237.13.70
174	24.3.0.37
147	212.64.74.145
144	212.209.164.3
115	24.9.203.188
102	24.43.169.231

8.3. Top internal source addresses

No. of scans	Source address
34472	MY.NET.218.90
17808	MY.NET.150.220
13376	MY.NET.202.50
12619	MY.NET.204.66
10397	MY.NET.150.133
9656	MY.NET.210.250
9299	MY.NET.228.54
9223	MY.NET.212.206
6948	MY.NET.203.234
6804	MY.NET.209.238
6773	MY.NET.150.143
6553	MY.NET.150.225
6552	MY.NET.217.58
6326	MY.NET.206.78
6269	MY.NET.217.142
6247	MY.NET.100.230
6140	MY.NET.98.176
6087	MY.NET.225.198
5199	MY.NET.224.238
4830	MY.NET.224.74
4616	MY.NET.217.222
4578	MY.NET.97.13
4570	MY.NET.218.118
4561	MY.NET.203.214
4324	MY.NET.98.150

8.4. Top internal destination addresses

No. of scans	Destination address
857	MY.NET.60.8
435	MY.NET.211.118
433	MY.NET.219.154
422	MY.NET.217.94
415	MY.NET.207.178
398	MY.NET.218.86
175	MY.NET.98.172
144	MY.NET.218.118

140	MY.NET.211.74
116	MY.NET.165.129
68	MY.NET.206.30
43	MY.NET.98.198
39	MY.NET.160.109
38	MY.NET.5.29
38	MY.NET.253.114
38	MY.NET.218.38
37	MY.NET.204.30
35	MY.NET.156.112
32	MY.NET.143.80
31	MY.NET.220.14
30	MY.NET.202.158

8.5. Top external destination addresses

No. of scans	Destination address
2533	129.2.246.94
2040	216.19.133.116
2010	172.132.71.130
1832	24.91.199.203
1722	63.21.61.147
1635	172.141.108.212
1578	172.169.147.76
1455	66.24.125.138
1423	63.14.172.15
1364	142.177.198.96
1360	24.19.99.230
1341	194.251.249.182
1323	24.6.245.220
1309	24.113.23.115
1272	66.30.167.225
1221	24.181.62.57
1210	24.183.99.210
1207	142.103.36.176
1143	199.17.65.223
1086	24.41.40.14
1057	63.29.200.59
1030	165.247.4.159
1019	24.165.188.50

9. Methodology for Assignment 3 Analysis

This analyst did not have available a server with IIS to run Snortsnarf. Thanks to Chris Kuethe, GCIA, for the perl scripts published in the practical. Some modifications to the scripts were made to run the script under Windows 2000 and to change the way output was printed for ease of sorting.

9.1. Alert analysis methodology

Alert files analyzed were: SnortA6.txt, SnortA3.txt, SnortAle.txt, SnortA35.txt and SnortA25.txt corresponding to dates Jan. 30, Feb. 3, Feb. 4, Feb. 6 and Feb. 11, respectively. SnortA36.txt was not included in the analysis because it duplicates data included in SnortA35.txt.

Refer to script alertcount.pl for flags to get total counts, counts by source and destination addresses for each type of alert recorded. Flag settings were added to the script for tracking purposes. Flags $a, $i and $l were not used by this analyst but were kept in the script. Only input lines with alerts were analyzed by this script. The modified version of this script is included in this practical for future reference.

Subncount.pl was written and used to combine counts by IP addresses' second or third octects when individual counts were small and made sense to present the information in this fashion. This script is included in this practical for future reference.

9.2. Port scan analysis methodology

Scan files analyzed were: SnortS8.txt, SnortS7.txt, SnortS26.txt, SnortS2.txt, SnortS2ca.txt, SnortS34.txt, SnortS32.txt, SnortS28.txt and SnortS27.txt corresponding to dates Jan. 21, Jan. 30, Feb. 1, Feb. 4, Feb. 5, Feb. 6, Feb. 7, Feb. 9 and Feb. 10, respectively.

Scanalyze perl script, from Chris Kuethe, was used to parse data before counting.

Scancount.pl was changed to create output easier to sort. The modified version of this script is included in this practical for future reference.

9.3. Alertcount.pl script

```
#  Modified alertcount.pl, original from Chris Kuethe
#  Uncomment needed flags
#$d="-d";
#$s="-s";
#$p="-p";
#$t="-t";
#$q="-q";
#$v="-v";
#$a="-a";

unless (defined($d) ||defined($s) ||defined($q) ||defined($p) ||defined($t) ||defined($v)){
        print "you need to specify at least one action flag\n";
        print "\t-d \tprint the destination hosts\n";
        print "\t-s \tprint the source hosts\n";
        print "\t-p \tprint the attacker/target pair\n";
        print "\t-t \tprint the attack types\n";
        print "\t-q \tbe quiet and print the total number of detects\n";
        print "\t-v \tbe verbose and print everything\n";
        print "\t-a \tprocess all (don't ignore portscans)\n";
```

```
    print "\t-i=file\tread a list of patterns to skip from \n";
    print "\t-l=n\tthreshold before printing\n";
    exit 1;
    }
if (defined($v) && defined($q)){
    print "the '-q' and '-v' flags are mutually exclusive.\n";
    exit 1;
    }

#the skip list contains case-sensitive patterns, one per line
#of strings, which, if found in the alert, cause processing of
#that alert to be skipped.
if (defined($i)) {
    open(SKIPLIST,$i) || die "can't open skip list \"$s\" ! ($!)\n";
    while (){
        chomp;
        push(@skiplist,$_)
    }
    close SKIPLIST;
}
while (<>){
    chomp;

    #make sure we have a log line
    unless (/\Q [**] \E/){ next };

#assuming that there are any alerts we're not interested in, we skip them
#here. portscans shouldn't be that interesting, since we have all the
#output from the portscan logger.
    $skipthis=0;
    if (( /spp_portscan/i ) && ( !defined($a) )){ next; }
    foreach $s (@skiplist){
        if ( $_ =~ /$s/ ){ $skipthis=1; }
    }; if ($skipthis) { next; }

    ($timestamp,$desc,$ip)=split(/\Q [**] \E/, $_);
    ($src, $arrow, $dst) = split(/ /, $ip);
    ($s_h,$s_p)=split(/:/,$src);
    ($d_h,$d_p)=split(/:/,$dst);
    $pkey = "${s_h}-${d_h}XXX$desc";
    $skey = "${s_h}XXX$desc";
    $dkey = "${d_h}XXX$desc";

    $atype{$desc} += 1 ;
    $pair{$pkey} += 1 ;
    $asrc{$skey} += 1 ;
    $adst{$dkey} += 1 ;

    $at2{$desc} += 1 ;
    $pr2{"${s_h}-${d_h}"} += 1 ;
    $as2{"${s_h}"} += 1 ;
    $ad2{"${d_h}"} += 1 ;

}
if (((!$q)&&($t))||($v)){
    foreach $key (sort keys(%atype)){
        printf "%8d\t$key\n", $atype{$key};
    }
}
if (((!$q)&&($d))||($v)){
    foreach $key (sort keys(%adst)){
        ($connection,$crime) = split(/XXX/, $key);
        printf "$crime\t%6d\t$connection\n", $adst{$key};
    }
}
if (((!$q)&&($s))||($v)){
    foreach $key (sort keys(%asrc)){
        ($connection,$crime) = split(/XXX/, $key);
        printf "$crime\t%6d\t$connection\n", $asrc{$key};
    }
}
if (((!$q)&&($p))||($v)){
    foreach $key (sort keys(%pair)){
        ($connection,$crime) = split(/XXX/, $key);
        printf "$crime\t%6d\t$connection\n", $pair{$key};
    }
}
if (($t)&&($q)){
    foreach $key (sort keys(%at2)){
        printf "%7d\t$key\n", $at2{$key};
    }
}
if (($d)&&($q)){
    foreach $key (sort keys(%ad2)){
        printf "%7d\t$key\n", $ad2{$key};
    }
}
if (($s)&&($q)){
    foreach $key (sort keys(%as2)){
        printf "%7d\t$key\n", $as2{$key};
    }
}
```

```
if (($p)&&($q)){
	foreach $key (sort keys(%pr2)){
		printf "%7d\t$key\n", $pr2{$key};
	}
}
```

9.4. Subncount.pl script

```
#number of significant octects, valid numbers 2 or 3
$no=3;

# expected input line, separated by tabs:
# Queso fingerprint	36	MY.NET.211.74

while (<>){
  chomp;
  ($desc, $count, $ip) = split(/\t/, $_);
  ($fst, $snd, $trd, $fth) = split(/\./, $ip);

# put together the octects to act as key
  if (($no==2) || ($no==3)) {
    if ($no==2) {
    $subn = join(".",$fst,$snd);
      } else {
      $subn = join(".",$fst,$snd,$trd);
      }
  }
  $subnkey = "${subn}XXX$desc";
  $asubnet{$subnkey} += $count ;
}
foreach $key (sort keys(%asubnet)){
  ($connection,$crime) = split(/XXX/, $key);
  printf "$crime\t%6d\t$connection\n", $asubnet{$key};
}
```

9.5. Scancount.pl script

```
#$d="-d";
#$s="-s";
#$p="-p";
#$t="-t";
#$f="-f";
#$v="-v";

unless (defined($d) ||defined($s) ||defined($p) ||defined($t) ||defined($v)){
	print "you need to specify at least one action flag\n";
```

```
        print "\t-d \tprint the target hosts\n";
        print "\t-s \tprint the attacking hosts\n";
        print "\t-p \tprint the attacker/target pair\n";
        print "\t-t \tprint the attack type\n";
        print "\t-f \twatch for fingerprinting attempts\n";
        print "\t-v \tbe verbose and print everything\n";
        print "\t-l=n\tconnection threshold before printing\n";
        exit 1;
        }

$l = 0 unless defined($l);

while (<>) {
        chomp;

        ($date, $time, $src, $dst, $scantype, @scanopts) = split;
        $pkey = "$src-$dst";

        unless (($scantype =~ /SYN/)||($scantype =~ /UDP/)||($scantype =~ /FIN/)){
                ++$fsrc{$src};
                ++$fdst{$dst};
                ++$fpr{$pkey};
                ++$ftyp{$scantype};
                }
        ++$asrc{$src};
        ++$adst{$dst};
        ++$type{$scantype};
        ++$pair{$pkey};
#       }
}
if (($t)||($v)){
        print "\n\nUnique Scan Types\n=================\n\n" if ($v) ;
        foreach $key (sort keys(%type)){
                if(($f)&&($ftyp{$key}>0)){ $fp="\t(fp)"; }else{ $fp=""; }
                printf "%8d\t$key $fp\n", $type{$key} unless ($type{$key} < $l);
        }
}
if (($d)||($v)){
        print "\n\nUnique Targets\n==============\n\n" if ($v) ;
        foreach $key (sort keys(%adst)){
                if(($f)&&($fdst{$key})){$fp="\t(fp)";}else{$fp="";}
                printf "%8d\t$key $fp\n", $adst{$key} unless ($adst{$key} < $l);
        }
}
if (($s)||($v)){
        print "\n\nUnique Attackers\n================\n\n" if ($v) ;
        foreach $key (sort keys(%asrc)){
                if(($f)&&($fsrc{$key})){$fp="\t(fp)";}else{$fp="";}
                printf "%8d\t$key $fp\n", $asrc{$key} unless ($asrc{$key} < $l);
        }
}
if (($p)||($v)){
        print "\n\nUnique Attacks/Targets\n=====================\n\n" if ($v) ;
        foreach $key (sort keys(%pair)){
                if(($f)&&($fpr{$key})){$fp="\t(fp)";}else{$fp="";}
                printf "%8d\t$key $fp\n", $pair{$key} unless ($pair{$key} < $l);
        }
}
```

10. References

[1] Stevens, W. Richard; “TCP/IP Illustrated, Volume 1”; Pages 34-35; 1994.

This is a partial list of Internet references used during the preparation of this practical:

http://www.sans.org
http://cve.mitre.org
http://www.securityfocus.com
http://www.cert.org
http://www.yahoo.com
http://advice.networkice.com/advice/Exploits/Ports
http://www.sans.org/newlook/resources/IDFAQ/oddports.htm
http://iana.org/assignments/port-numbers
http://www.arin.net
http://www.apnic.net
http://www.ripe.net
http://www.activestate.com
http://www.networkice.com/advice/Intrusions/2001705/default.htm
http://www.sans.org/y2k/practical/chris_kuethe_gcia.html
http://www.sans.org/y2k/practical/Paul_Asadoorian_GIAC.doc
http://www.sans.org/y2k/practical/PJ_Goodwin_GCIA.doc

Any omissions to this list are unintentional. Many websites are already referenced in the practical. After a lot of web searching, the author can not remember all visited websites.