Global Information Assurance Certification Paper

GCIA Certification Practical
SANS 2000 Monterey, CA
Graham Stork

ALL FRIENDLY NETWORK ADDRESSES SANITIZED
ALL NAMES NETWORK AND OTHERWISE SANITIZED

Assignment 1 – 4 Network Detects

Detect 1

08/08/00 13:43:52.248749 61.11.234.17.domain > MY.NETWORK.3.domain: SF **929211681:929211681**(0) win 1028
08/08/00 13:43:52.291378 61.11.234.17.domain > MY.NETWORK.5.domain: SF **929211681:929211681**(0) win 1028
08/08/00 13:43:52.410059 61.11.234.17.domain > MY.NETWORK.25.domain: SF **929211681:929211681**(0) win 1028
08/08/00 13:43:52.490101 61.11.234.17.domain > MY.NETWORK.30.domain: SF **929211681:929211681**(0) win 1028
08/08/00 13:43:54.413638 61.11.234.17.domain > MY.NETWORK.31.domain: SF 315751416:315751416(0) win 1028
08/08/00 13:43:54.651412 61.11.234.17.domain > MY.NETWORK.40.domain: SF 315751416:315751416(0) win 1028
08/08/00 13:43:55.848982 61.11.234.17.domain > MY.NETWORK.80.domain: SF 1083777355:1083777355(0) win 1028
08/08/00 13:43:56.602102 61.11.234.17.domain > MY.NETWORK.99.domain: SF **760449468:760449468**(0) win 1028
08/08/00 13:43:56.760634 61.11.234.17.domain > MY.NETWORK.15.domain: SF **760449468:760449468**(0) win 1028
08/08/00 13:43:56.795945 61.11.234.17.domain > MY.NETWORK.67.domain: SF **760449468:760449468**(0) win 1028

08/11/00 19:29:29.447792 61.11.233.126.domain > MY.NETWORK.3.domain: SF **1276756877:1276756877**(0) win 1028
08/11/00 19:29:29.452807 61.11.233.126.domain > MY.NETWORK.5.domain: SF **1276756877:1276756877**(0) win 1028
08/11/00 19:29:29.494527 61.11.233.126.domain > MY.NETWORK.25.domain: SF **1276756877:1276756877**(0) win 1028
08/11/00 19:29:29.542537 61.11.233.126.domain > MY.NETWORK.30.domain: SF **1276756877:1276756877**(0) win 1028

1> Source of trace
My network

2> Detect was generated by
TCPDUMP filter. The filter used was: tcp and (tcp[13] & 0x02 !=0) and (tcp[13] & 0x01 != 0) This filter looks for packets with both the SYN and FIN flags set. SYN/FIN is an illegal TCP flag combination.

Explanation of fields

08/08/00 13:43:52.248749 {timestamp} **61.11.234.17.domain** {source.ip.PORT} > **my.network.3.domain** {destination.ip.PORT} **:SF** {flags} **929211681:929211681(0)** {beginning sequence number : ending sequence number (data bytes) **win 1028** {windows size)

3> Probability the source address was spoofed
Low. The IP address set is registered to Asia Pacific Network.
Address: 61.11.234.17
Address: 61.11.233.126
Netname: APNIC3 (Asia Pacific Network Information Center)
Netblock: 61.0.0.0 - 61.255.255.255
Maintainer: AP
Addresses have been further assigned to Asia-Pacific users. Additional information not available for these IP addresses.

4> Description of attack
This is a SYN/FIN reconnaissance attack. The information gathered might allow the attacked to identify the operating system of a computer, or computers, running DNS. Information gathered could be used to attempt an exploit against a known DNS vulnerability.

Some example exploits by CVE are:
CVE-1999-0010 denial of service against BIND 8 releases
CVE-1999-0274 denial of service against Windows NT machines
CVE-1999-0299 buffer overflow against FreeBSD machines

5> Attack mechanism
The attacker tries to by pass network intrusion detection systems by using an impossible TCP flags combination. Also note the static sequence numbers and the fact that the source and destination ports are always the same. This is the signature of a reflexive scan. The TCP packets must be crafted to execute this attack.

The attacker was looking for systems running DNS, and possibly for information regarding the operating system running the DNS service, if the service is found. For example, the Linux OS will respond to a SYN-FIN packet with a SYN-FIN-ACK. If this response were to come from a box running DNS, well known exploits for Linux DNS could be used against the machine.

6> Correlations
The SYN/FIN attack was described at SANS 2000, Monterey, CA. Information regarding the attack is found on page 114 of text 3.2 and page (ff) of 3.5/3.6

7> Active targeting
NO. The scan was directed at a variety of machines. None of which are running DNS.

8> Severity
(Criticality + Lethality) - (System + Net counter measures) = severity
(4 + 4) - (5 + 1) = 2

A variety of systems, mostly desktops, were targeted. None the targeted machines run DNS.

9> Defensive recommendation
The router defenses were not sufficient to block this attack. The router ACLs should be updated to allow domain traffic only to the DNS server. A firewall would be useful. All Unix systems should be checked to verify DNS is not running. DNS should only be running on the DNS server. The DNS server should be checked for patch level and DNS software version, and updated as necessary.

10> Multiple choice question

This trace is an example of (choose the most descriptive answer).

A> SYN/FIN attack
B> SYN/FIN source port 0 scan
C> SYN/FIN without packet crafting
D> SYN/FIN reflexive scan.

Answer: D

Detect 2

ms5out1.messagemedia.com > ldude.my.network

18:17:53.542731 ms5out1.messagemedia.com.3120 > ldude.my.network.25: S 252762237:252762237(0) win 8192 (DF)
18:17:56.717128 ms5out1.messagemedia.com.3120 > ldude.my.network.25: S 252762237:252762237(0) win 8192 (DF)
18:18:03.282628 ms5out1.messagemedia.com.3120 > ldude.my.network.25: S 252762237:252762237(0) win 8192 (DF)
18:18:16.408676 ms5out1.messagemedia.com.3120 > ldude.my.network.25: S 252762237:252762237(0) win 8192 (DF)

18:18:44.746017 ms5out1.messagemedia.com.1143 > ldude.my.network.25: S 252813443:252813443(0) win 8192 (DF)
18:18:48.018883 ms5out1.messagemedia.com.1143 > ldude.my.network.25: S 252813443:252813443(0) win 8192 (DF)
18:18:54.579214 ms5out1.messagemedia.com.1143 > ldude.my.network.25: S 252813443:252813443(0) win 8192 (DF)
18:19:07.705520 ms5out1.messagemedia.com.1143 > ldude.my.network.25: S 252813443:252813443(0) win 8192 (DF)

18:19:34.773397 ms5out1.messagemedia.com.3156 > ldude.my.network.25: S 252863485:252863485(0) win 8192 (DF)
18:19:38.000651 ms5out1.messagemedia.com.3156 > ldude.my.network.25: S 252863485:252863485(0) win 8192 (DF)
18:19:44.561772 ms5out1.messagemedia.com.3156 > ldude.my.network.25: S 252863485:252863485(0) win 8192 (DF)
18:19:57.681237 ms5out1.messagemedia.com.3156 > ldude.my.network.25: S 252863485:252863485(0) win 8192 (DF)

1> Source of trace
My network

2> Detect was generated by
SHADOW IDS
18:17:53.542731 {time} **ms5out1.messagemedia.com.3120** {source.ip.PORT} > **ldude.my.network.25** {destination.ip.PORT} **: S** {flags} **252762237:252762237(0)** {beg. sequence # : ending sequence # (data bytes)}**win 8192** {window size} **(DF)** {do not fragment flag is set}

3> Probability the source address was spoofed
NONE. Domain Name: MESSAGEMEDIA.COM
Registrar: NETWORK SOLUTIONS, INC.
Whois Server: whois.networksolutions.com
Referral URL: www.networksolutions.com
Name Server: NSIF.0MM.COM
Name Server: NSCOOP.0MM.COM
Updated Date: 18-jul-2000

4> Description of attack
False detect. This was not an attack

5> Attack mechanism
This was a false detect. It was not an attack.

The IDS issued this detect several days after a demonstration project email server was shutdown. To our knowledge the server had not been used for communications outside the local office, but we did not strictly enforce that usage of the system. We were surprised to discover that an outside system knew about and was trying to contact the demonstration server. At first we thought this was a scan directed at the SMTP port of the mail server. But upon closer examination we determined the source IP to be friendly. Eventually we remembered that ldude.my.network had been recently shutdown.

We decided that the "detect" was really a valid attempt by an external mail system to deliver mail to ldude.my.network. The "scan" behavior is what you would expect from a source SMTP server attempting to send/re-send mail to a destination SMTP server that is not accepting connections.

6> Correlations

None

7>Evidence of active targeting

Yes. Although this was not an attack

8> Severity

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

(5 + 0) - (5 + 5) = -5

A targeted attack against a mail server would get a Criticality value of 5.
It was NOT an attack so Lethality is 0.
The system was not in use and had been shutdown. This gives a System Countermeasures value of 5.
The Network performed as expected. This gives a Network Countermeasures value of 5.

9> Defensive recommendation

Turn off all systems, demonstration and otherwise, that are not being used to provide some service.

10> Multiple choice question

Using the scan shown above and knowing that ms5out1.messagemedia.com is a valid friendly source, and that ldude.my.network was a demonstration mail server that has been recently turned off. This scan is an example of:

A> TCP retry
B> Attempted Denial of Service via SYN flood
C> False Detect / False Positive
D> A and C

Answer C

Detect 3

Note: Log file lines alternately **bolded** and not bolded for readability

```
Oct 28 01:35:58 router.10 11084: 1d10h: %SEC-6-IPACCESSLOGP: list 101 denied udp
63.205.41.94(61002) -> MY.NETWORK.20(137), 1 packet
Oct 28 01:36:08 router.10 11086: 1d10h: %SEC-6-IPACCESSLOGP: list 101 denied udp
63.205.41.94(61058) -> MY.NETWORK.21(137), 1 packet
Oct 28 01:36:19 router.10 11087: 1d10h: %SEC-6-IPACCESSLOGP: list 101 denied udp
63.205.41.94(61064) -> MY.NETWORK.22(137), 1 packet
Oct 28 01:36:30 router.10 11088: 1d10h: %SEC-6-IPACCESSLOGP: list 101 denied udp
63.205.41.94(61074) -> MY.NETWORK.23(137), 1 packet
               :
               : information cut for brevity
               :
Oct 28 02:11:37 router.10 11542: 1d10h: %SEC-6-IPACCESSLOGP: list 101 denied udp
63.205.41.94(61868) -> MY.NETWORK.246(137), 1 packet
Oct 28 02:11:48 router.10 11543: 1d10h: %SEC-6-IPACCESSLOGP: list 101 denied udp
63.205.41.94(61870) -> MY.NETWORK.247(137), 1 packet
Oct 28 02:12:00 router.10 11546: 1d10h: %SEC-6-IPACCESSLOGP: list 101 denied udp
63.205.41.94(61872) -> MY.NETWORK.248(137), 1 packet
Oct 28 02:12:09 router.10 11548: 1d10h: %SEC-6-IPACCESSLOGP: list 101 denied udp
63.205.41.94(61874) -> MY.NETWORK.249(137), 1 packet
Oct 28 02:12:20 router.10 11549: 1d10h: %SEC-6-IPACCESSLOGP: list 101 denied udp
63.205.41.94(61876) -> MY.NETWORK.250(137), 1 packet
Oct 28 02:12:31 router.10 11553: 1d10h: %SEC-6-IPACCESSLOGP: list 101 denied udp
63.205.41.94(61878) -> MY.NETWORK.251(137), 1 packet
Oct 28 02:12:41 router.10 11555: 1d10h: %SEC-6-IPACCESSLOGP: list 101 denied udp
63.205.41.94(61880) -> MY.NETWORK.252(137), 1 packet
Oct 28 02:12:52 router.10 11558: 1d10h: %SEC-6-IPACCESSLOGP: list 101 denied udp
63.205.41.94(61882) -> MY.NETWORK.253(137), 1 packet
Oct 28 02:13:01 router.10 11559: 1d10h: %SEC-6-IPACCESSLOGP: list 101 denied udp
63.205.41.94(61884) -> MY.NETWORK.254(137), 1 packet
```

1> Source of trace

My network

2> Detect was generated by

Cisco Access Control Lists. The rule that fired the detect disallows incoming UDP to port 137.

Explanation of fields

Oct 28 02:12:52 {timestamp} **router.10** {hostname/route IP address} **11558: 1d10h: %SEC-6-IPACCESSLOGP:** {Cisco router info} **list 101** {Access list triggered} **denied** {action take} **udp** {protocol} **63.205.41.94(61882)** {source IP address and (port) -> **MY.NETWORK.253(137)** {destination IP address and (port), **1 packet** {number of packets sent}

3> Probability the source address was spoofed

Low. Addresses in this range are assigned to Pacific Bell Internet Services.
Seeking Info on: 63.205.41.94
[whois.arin.net]
Pacific Bell Internet Services,Inc. (NETBLK-PBI-NET-7) PBI-NET-7
63.192.0.0 - 63.207.255.255
LSAN03 ADSL Rback17 PPPoX (NETBLK-SBCIS-10047-14946) SBCIS-10047-14946
63.205.40.0 - 63.205.47.255

4> Description of attack

The attacker attempted to scan one of our class C networks starting at .20 continuing sequentially through .254. The scan employs the UDP protocol. The source IP address is constant and the source port changes during the scan. The destination port is constant and is always port 137. The attack lasted for about 35 minutes. The attack was blocked by the router UDP filter.

Some CVEs that are relevant

CVE-1999-0288	WINS Denial of Service
CVE-1999-0225	Denial of Service via malformed logon
CVE-1999-0391	SMB authentication problems with Win9x

5> Attack mechanism

This attack is a scan looking for NetBIOS name services. Port 137 can provide significant information to an attacker. Once these systems listening on port 137 are identified attacks against port 137 can be launched. These future exploits could be denial of service attacks or attempts to exploit some other aspect of NetBIOS name services. The probe may have been an SMB-Name Wildcard attack, but we can't determine that information because our IDS never show the packets because they were blocked by the router.

6> Correlations

UDP scans of port 137 were discussed at SANS Network Security, Monterey, CA. in 3.5/3.6 IDS Signatures and Analysis.

7> Evidence of active targeting

NO. This was a very systematic scan.

8> Severity

(3 + 3) – (5 + 3) = -2
Desktop systems were targeted. No servers running NetBIOS. Network counter measures were effective, some of the desktops are running older versions of Microsoft Windows.

9> Defensive recommendation

Make sure that the router ACLs are properly maintained. And again, a firewall would be a useful addition to our network infrastructure.

10> Multiple choice question

The scan above is an example of

A> A low and slow scan
B> A scan with some decoy addresses
C> A scan targeting a specific machine on the network
D> A scan looking for Windows NetBIOS services

Answer D

Detect 4

```
Jun 27 16:09:53 router.10 82834: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.10.100(1221) -> WWWServer(80), 1 packet
Jun 27 16:15:23 router.10 82836: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.10.100(1221) -> WWWServer(80), 5 packets
Jun 28 21:03:37 router.10 83220: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.40.10(2168) -> WWWServer(80), 1 packet
Jun 28 21:08:46 router.10 83221: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.40.10(2168) -> WWWServer(80), 5 packets
Jun 29 15:25:24 router.10 83302: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.0.32(52904) -> WWWServer(80), 1 packet
Jun 29 15:25:26 router.10 83303: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.0.32(52908) -> WWWServer(80), 1 packet
Jun 29 15:25:47 router.10 83304: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.0.32(52964) -> WWWServer(80), 1 packet
Jun 29 15:31:00 router.10 83305: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.0.32(52904) -> WWWServer(80), 1 packet
Jun 29 15:36:00 router.10 83306: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.0.32(52904) -> WWWServer(80), 1 packet
Jun 30 20:27:24 router.10 83618: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.16.33(2536) -> WWWServer(80), 1 packet
Jun 30 20:28:41 router.10 83621: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.16.33(2539) -> WWWServer(80), 1 packet
Jun 30 20:34:23 router.10 83624: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.16.33(2539) -> WWWServer(80), 5 packets
Jun 30 21:10:06 router.10 83636: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.16.33(2630) -> WWWServer(80), 1 packet
Jun 30 21:15:24 router.10 83642: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.16.33(2630) -> WWWServer(80), 5 packets
Jun 30 21:32:29 router.10 83648: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.16.33(2716) -> WWWServer(80), 1 packet
Jun 30 22:48:11 router.10 83670: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.16.6(2462) -> WWWServer(80), 1 packet
Jun 30 22:48:23 router.10 83671: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.16.6(2460) -> WWWServer(80), 1 packet
Jun 30 22:50:03 router.10 83673: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.16.6(2478) -> WWWServer(80), 1 packet
Jun 30 22:53:25 router.10 83675: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.16.6(2465) -> WWWServer(80), 6 packets
Jun 30 22:55:25 router.10 83676: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.16.6(2478) -> WWWServer(80), 6 packets
           :
           : Cut for brevity
           :
Jul  3 16:28:27 router.10 91285: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.16.6(2480) -> WWWServer(80), 1 packet
Jul  3 16:34:14 router.10 91296: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.16.6(2475) -> WWWServer(80), 5 packets
Jul  3 17:11:23 router.10 91315: %SEC-6-IPACCESSLOGDP: list 101 denied icmp
172.16.8.2 -> WWWServer (3/1), 1 packet
Jul  3 20:31:11 router.10 91372: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.1.59(1612) -> WWWServer(80), 1 packet
Jul  3 20:36:17 router.10 91375: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.1.59(1612) -> WWWServer(80), 5 packets
```

```
Jul  3 23:49:50 router.10 91437: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.1.142(2801) -> WWWServer(80), 1 packet
Jul  3 23:50:15 router.10 91438: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.1.142(2796) -> WWWServer(80), 1 packet
Jul  3 23:50:17 router.10 91439: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.1.142(2798) -> WWWServer(80), 1 packet
Jul  3 23:50:27 router.10 91440: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.1.142(2800) -> WWWServer(80), 1 packet
Jul  3 23:55:20 router.10 91441: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.1.142(2798) -> WWWServer(80), 5 packets
Jul  3 23:56:20 router.10 91442: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.1.142(2800) -> WWWServer(80), 5 packets
Jul  5 16:43:49 router.10 91894: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.16.6(2786) -> WWWServer(80), 1 packet
Jul  5 16:48:50 router.10 91895: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.16.6(2786) -> WWWServer(80), 5 packets
Jul  5 17:53:29 router.10 91901: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.16.47(1605) -> WWWServer(80), 1 packet
Jul  5 17:58:51 router.10 91902: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.16.47(1605) -> WWWServer(80), 5 packets
Jul  5 18:08:17 router.10 91908: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.16.47(1622) -> WWWServer(80), 1 packet
Jul  5 18:09:01 router.10 91910: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.16.47(1631) -> WWWServer(80), 1 packet
Jul  5 18:13:51 router.10 91911: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.16.47(1622) -> WWWServer(80), 5 packets
Jul  5 18:14:51 router.10 91912: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.16.47(1631) -> WWWServer(80), 5 packets
Jul  5 19:14:24 router.10 91937: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.82.29(1939) -> WWWServer(80), 1 packet
Jul  5 20:31:51 router.10 91948: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.26.15(1462) -> WWWServer(80), 1 packet
Jul  5 20:31:52 router.10 91949: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.26.15(1461) -> WWWServer(80), 1 packet
Jul  5 20:32:15 router.10 91950: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.26.15(1463) -> WWWServer(80), 1 packet
Jul  5 20:32:39 router.10 91951: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.26.15(1465) -> WWWServer(80), 1 packet
Jul  5 20:36:53 router.10 91952: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.26.15(1462) -> WWWServer(80), 4 packets
Jul  6 00:02:34 router.10 91988: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.1.67(1914) -> WWWServer(80), 1 packet
Jul  6 01:19:11 router.10 92007: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.1.67(1913) -> WWWServer(80), 1 packet
Jul  6 01:24:57 router.10 92009: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.1.67(1913) -> WWWServer(80), 5 packets
           :
           : Information cut for brevity
           :
Aug 21 17:05:02 router.10 110411: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.40.10(1294) -> WWWServer(80), 1 packet
Aug 21 17:10:34 router.10 110425: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.40.10(1294) -> WWWServer(80), 5 packets
Aug 21 19:02:16 router.10 110475: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.40.10(1528) -> WWWServer(80), 1 packet
Aug 21 19:07:36 router.10 110477: %SEC-6-IPACCESSLOGP: list 101 denied tcp
172.16.40.10(1528) -> WWWServer(80), 5 packets
```

1> Source of trace
My network

2> Detect was generated by
This detect was generated by Cisco ACLs on the inbound connection with our ISP. The purpose of this filter is to block packets with non-routeable IP addresses.

Explanation of fields
Aug 21 19:07:36 {timestamp} **router.10** {hostname/IP address} **110477: %SEC-6-IPACCESSLOGP:** {Cisco router info} **list 101** {ALC that triggered} **denied** {action taken} **tcp** {protocol flagged} **172.16.40.10(1528)** {source IP(port)} -> **WWWServer(80)**{destination IP (port)}**, 5 packets** {number of packets sent}

3> Probability the source address was spoofed
HIGH. The IP addresses used in this attack, 176.16.xxx.xxx, are reserved by IANA and should never appear as the source address of packets entering the network.

4> Description of attack
The attack was not a scan. The attacker (maybe attackers) is using a variety of reserved addresses in an attempt to connect to port 80 of our WWW server. The attack took place over 20 or so days with only a few probes of the server per day. We continue to see a surprising number of attempts to connect to our network from reserved address sets.

Some potentially relevant CVEs

CVE-1999-0437	WebRamp denial of service via malicious string to HTTP port
CVE-1999-0071	Apache buffer overflow attack
CVE-1999-0867	Denial of Service IISv4.0 HTTP via malformed headers
CAN-1999-0107	Apache buffer overflow attack

5> Attack mechanism
The attacker is trying to initiate TCP connections on port 80 of our WWW server. The attack is a stimulus of some kind, it is not a scan as scans are not effective when reserved addresses are used, because the information gained by the scan is not returned to the attacker. This is not a random occurrence as it repeats over 20 days. It might be an attempt to run a denial of service against the WWW server, but I believe the number of probes is NOT sufficient to do this, at least by way of a SYN flood attack. There just aren't enough packets targeted at the server to make this believable. Other forms of denial of service attacks are possible.

These packets could be decoy packets. A check of the WWW log files may turn up something interesting, but those logs are not available for this time period.

A question that I have is "Can a CGI-bin exploit that allows an attacker to gain access to the server succeed when the attack originates from a reserved IP address?" I don't think this is possible, but if it is possible that opens up many other possibilities.

6> Correlations

A search for the different uses of port 80 on the www.snort.org site shows a variety of attacks known for port 80. Executor, Executor1, Executor2, Hooker, Ring Zero.

7> Evidence of active targeting

HIGH. The system being targeted is our WWW server.

8> Severity

(Criticality + Lethality) – (Network + System countermeasures) = Severity

(5 + 4) – (5 + 5) = -1

Targeted box is a www server (5). Assume lethality of 4, could be 3 or 5. Network counter measures were fully successful (5). The system has been hardened and system maintenance is up to date (5).

9> Defensive recommendation

Make sure that the router ALCs are properly maintained. Keep the WWW server patch set up to date.

10> Question

Which of the statements below are true?

A> Web Servers are vulnerable to CGI-bin attacks
B> Web Servers and FTP servers are vulnerable to CGI-bin attacks
C> Web Servers are vulnerable to denial of service attacks, but not to CGI-bin attacks
D> Web Servers are vulnerable to both A and B

Answer: A

Assignment 2 - Evaluate an Attack

The attack described below occurred on November 15th. It was more of a spoof than an attack, and a poorly constructed spoof at that. But it does serve to illustrate some important points regarding network security, network monitoring and intrusion detection.

Background to the attack

Our organization recently adopted a new email system, moving from a proprietary system to an internet standards system. Many people raised security and privacy related questions regarding messages posted to and received on the new system. The users were uncomfortable with the fact the message headers were stored on the desktop in clear text and could be accessed by anyone with access to the PC (The PCs are Windows 9x and as such do not have file system security). The message store is on a Sun Solaris system and is secure to the full extent of the Sun administrator's knowledge. This information was provided to all employees.

One individual repeatedly claimed, but never demonstrated, that more than message headers were available on the local computer. This user argued that message bodies were kept in cache on the local machine. The mail administrators argued that messages stored on the server were secure and that access to messaging accounts was as secure as the passwords defined on the accounts. Mail administrators also argued that messages are stored locally only if explicitly download for local storage by the user.

Description and intent of the attack

The attack was an insider attack. The attacker managed to get another individual's email password. The attacker used that password to access the victim's email account. From the victim's account the attacker sent messages to a mailing list in a way that the attacker believed he was able to claim that the email system had failed, implying that the failure indicated a breach of system security and integrity. The intent of the attack was to undermine user confidence in the email system and the administration of that system. It turns out that the individual making the security claims is Daniel.R.Attacker mentioned below.

Detailed discussion of the attack

Some of the messages in the exchange have been left out of this document. The messages left out do not change the analysis. In the messages below the moniker 'weatherbear' is the known moniker of Daniel.R.Attacker

We begin by looking at a sequence of emails sent by Ricard.Victim and Daniel.R.Attacker. Relevant information to that attack is displayed in **bold.**

The first message in the sequence starts below. Daniel.R.Attacker has accessed Richard.Victim's mail account using Richard.Victim's uid (user id) and password. The message appears to be legitimately sent from Richard.Victim's account. It was in fact sent from Richard.Victim's account, but not legitimately sent. Richard.Victim did not authorize this access.

---------- first message in sequence
Return-Path: <Richard.Victim@xxx.yyy>
Received: from aaa.xxx.yyy ([127.0.0.1]) by nsmail.aaa.xxx.yyy
(Netscape Messaging Server 4.15) with ESMTP id G42VU600.MLT for
<abc.all.hands@nsmail.aaa.xxx.yyy>; Wed, 15 Nov 2000 **11:10:54 -0700**
From: "Richard Victim" <Richard.Victim@xxx.yyy>
To: abc.all.hands@nsmail.aaa.xxx.yyy
Message-ID: **<4c33a4aeed.4aeed4c33a@aaa.xxx.yyy>**
Date: Wed, 15 Nov 2000 11:10:54 -0700
X-Mailer: Netscape Webmail
MIME-Version: 1.0
Content-Language: en
Subject: New Sandy Cam
X-Accept-Language: en
Content-Type: text/plain; charset=us-ascii
Content-Disposition: inline
Content-Transfer-Encoding: 7bit
Precedence: list
Resent-From: abc.all.hands@nsmail.aaa.xxx.yyy

Hello all...I have added a webcam to the THETA Northern Mosaic Page.
The camera is located in Sandy, and is looking west toward the Oquirrh
Mountains. It is overlooking a trailer park. I didn't even know they
allowed trailer parks in Sandy. :)

weatherbear
---------- end of first message

The second message in the sequence. One minute and 56 seconds later Daniel.R.Attacker sends a message identical to the message above, but this time from the Daniel.R.Attacker account.

---------- second message in sequence
Return-Path: <Daniel.R.Attacker@xxx.yyy>
Received: from aaa.xxx.yyy ([127.0.0.1]) by nsmail.aaa.xxx.yyy
(Netscape Messaging Server 4.15) with ESMTP id G42VXE00.NN0 for
<abc.all.hands@nsmail.aaa.xxx.yyy>; Wed, 15 Nov 2000 **11:12:50 -0700**
From: "Daniel R Attacker" <Daniel.R.Attacker@xxx.yyy>
To: abc.all.hands@nsmail.aaa.xxx.yyy
Message-ID: **<4bef6488cb.488cb4bef6@aaa.xxx.yyy>**
Date: Wed, 15 Nov 2000 11:12:50 -0700
X-Mailer: Netscape Webmail
MIME-Version: 1.0
Content-Language: en
Subject: New Sandy Cam
X-Accept-Language: en
Content-Type: text/plain; charset=us-ascii

Content-Disposition: inline
Content-Transfer-Encoding: 7bit
Precedence: list
Resent-From: abc.all.hands@nsmail.aaa.xxx.yyy

Hello all...I have added a webcam to the THETA Northern Mosaic Page.
The camera is located in Sandy, and is looking west toward the Oquirrh
Mountains. It is overlooking a trailer park. I didn't even know they
allowed trailer parks in Sandy. :)

weatherbear
---------- end of second message

The third message in the sequence: Five minutes and 29 seconds later, Daniel.R.Attacker, still logged into his account sends the third message. This message exclaims disbelief of the earlier sequence, implying that the message from Richard.Victim must be an error attributable to the email system. Notice that Daniel.R.Attacker sequenced the messages incorrectly. The message from Richard.Victim should be the second message in the sequence, not the first message in the sequence.

---------- third message in sequence
Return-Path: <Daniel.R.Attacker@xxx.yyy>
Received: from aaa.xxx.yyy ([127.0.0.1]) by nsmail.aaa.xxx.yyy
(Netscape Messaging Server 4.15) with ESMTP id G42W6J00.6NH;
Wed, 15 Nov 2000 **11:18:19 -0700**
From: "Daniel R Attacker" <Daniel.R.Attacker@xxx.yyy>
To: "Daniel R Attacker" <Daniel.R.Attacker@xxx.yyy>
CC: abc.all.hands@nsmail.aaa.xxx.yyy
Message-ID: **<4c6cf4e734.4e7344c6cf@aaa.xxx.yyy>**
Date: Wed, 15 Nov 2000 11:18:19 -0700
X-Mailer: Netscape Webmail
MIME-Version: 1.0
Content-Language: en
Subject: Re: New Sandy Cam
X-Accept-Language: en
Content-Type: text/plain; charset=us-ascii
Content-Disposition: inline
Content-Transfer-Encoding: 7bit
Precedence: list
Resent-From: abc.all.hands@nsmail.aaa.xxx.yyy

Why did my e-mail go out under Dick's name too? What's going on here?

----- Original Message -----
From: "Daniel R Attacker" <Daniel.R.Attacker@xxx.yyy>
Date: Wednesday, November 15, 2000 11:12 am
Subject: New Sandy Cam

> Hello all...I have added a webcam to the THETA Northern Mosaic
> Page.
> The camera is located in Sandy, and is looking west toward the
> Oquirrh
> Mountains. It is overlooking a trailer park. I didn't even know
> they
> allowed trailer parks in Sandy. :)
>
> **weatherbear**
---------- end of third message

The fourth message in the sequence: Twenty six minutes and 30 seconds after the third message in the sequence Daniel.R.Attacker creates and sends another copy of the "original" message. If Daniel.R.Attacker was hoping to add to the confusion he failed.

---------- fourth message in sequence
Return-Path: <Daniel.R.Attacker@xxx.yyy>
Received: from aaa.xxx.yyy ([127.0.0.1]) by nsmail.aaa.xxx.yyy
(Netscape Messaging Server 4.15) with ESMTP id G42XEP00.0M1 for
<abc.all.hands@nsmail.aaa.xxx.yyy>; **Wed, 15 Nov 2000 11:44:49 -0700**
From: "Daniel R Attacker" <Daniel.R.Attacker@xxx.yyy>
To: abc.all.hands@nsmail.aaa.xxx.yyy
Message-ID: **<4e5c2496fb.496fb4e5c2@aaa.xxx.yyy>**
Date: Wed, 15 Nov 2000 11:44:49 -0700
X-Mailer: Netscape Webmail
MIME-Version: 1.0
Content-Language: en
Subject: New Sandy Cam
X-Accept-Language: en
Content-Type: text/plain; charset=us-ascii
Content-Disposition: inline
Content-Transfer-Encoding: 7bit
Precedence: list
Resent-From: abc.all.hands@nsmail.aaa.xxx.yyy

Hello all...I have added a webcam to the THETA Northern Mosaic
Page. The camera is located in Sandy, and is looking west toward the
Oquirrh Mountains. It is overlooking a trailer park. I didn't even know
they allowed trailer parks in Sandy. :)

weatherbear
---------- end of fourth message

The fifth and last message in the sequence: Eleven minutes and seven seconds after the fourth message, Daniel.R.Attacker has once again logged into Richard.Victim's account. But now he is really confused.

He sends the exclamation of disbelief from the Richard.Victim's account not from Daniel.R.Attackers account.

---------- fifth message in sequence
Return-Path: <Richard.Victim@xxx.yyy>
Received: from aaa.xxx.yyy ([127.0.0.1]) by nsmail.aaa.xxx.yyy
(Netscape Messaging Server 4.15) with ESMTP id G42XX800.KM2 for
<abc.all.hands@nsmail.aaa.xxx.yyy>; **Wed, 15 Nov 2000 11:55:56 -0700**
From: "Richard Victim" <Richard.Victim@xxx.yyy>
To: abc.all.hands@nsmail.aaa.xxx.yyy
Message-ID: <4d8f84da3c.4da3c4d8f8@aaa.xxx.yyy>
Date: Wed, 15 Nov 2000 11:55:56 -0700
X-Mailer: Netscape Webmail
MIME-Version: 1.0
Content-Language: en
Subject: Re: New Sandy Cam
X-Accept-Language: en
Content-Type: text/plain; charset=us-ascii
Content-Disposition: inline
Content-Transfer-Encoding: 7bit
Precedence: list
Resent-From: abc.all.hands@nsmail.aaa.xxx.yyy

Why did my e-mail go out under Dick's name too? What's going on here?

----- Original Message -----
From: "Daniel R Attacker" <Daniel.R.Attacker@xxx.yyy>
Date: Wednesday, November 15, 2000 11:12 am
Subject: New Sandy Cam

> Hello all...I have added a webcam to the THETA Northern Mosaic
> Page.
> The camera is located in Sandy, and is looking west toward the
> Oquirrh
> Mountains. It is overlooking a trailer park. I didn't even know
> they
> allowed trailer parks in Sandy. :)
>
> **weatherbear**
---------- end of fifth message.

Below are portions of the relevant log files relating to the exploit. Notice that the IP address is constant across all log files and all connects to the mail server regardless of the uid that is logged. The IP address in question, **63.228.195.111,** is registered to uswest.net. The address resolves to slkc6400gw3poolB111.slkc.uswest.net. Daniel.R.Attacker has an ISP account with uswest.net.

{From the HTTP log files: The first line shows Richard.Victim logging in. This is really Daniel.R.Attacker using the uid of the victim. One message is sent.}

[15/Nov/2000:**11:04:07** -0700] nsmail **httpd**[28047]: Account Information: login [**63.228.195.111**] **Richard.Victim** [null]
[15/Nov/2000:11:04:07 -0700] nsmail httpd[28047]: Account Information: login: [**63.228.195.111**] **Richard.Victim** plaintext
[15/Nov/2000:**11:10:54** -0700] nsmail httpd[28047]: General Notice: Richard.Victim[**63.228.195.111**] created message **<4c33a4aeed.4aeed4c33a@aaa.xxx.yyy>**
[15/Nov/2000:**11:11:41** -0700] nsmail httpd[28047]: Account Notice: close **63.228.195.111 Richard.Victim** 2000/11/15 11:04:07 0:07:34 0 0 1

{The victim has logged out and Daniel.R.Attacker is logging in. Notice the IP address. Two messages are sent.}

[15/Nov/2000:**11:11:49** -0700] nsmail httpd[28047]: Account Information: login [**63.228.195.111**] **Daniel.R.Attacker** [null]
[15/Nov/2000:**11:11:49** -0700] nsmail httpd[28047]: Account Information: login: [**63.228.195.111**] **Daniel.R.Attacker** plaintext
[15/Nov/2000:**11:12:50** -0700] nsmail httpd[28047]: General Notice: **Daniel.R.Attacker**[**63.228.195.111**] created message **<4bef6488cb.488cb4bef6@aaa.xxx.yyy>**
[15/Nov/2000:**11:18:19** -0700] nsmail httpd[28047]: General Notice: **Daniel.R.Attacker**[**63.228.195.111**] created message **<4c6cf4e734.4e7344c6cf@aaa.xxx.yyy>**
[15/Nov/2000:**11:18:46** -0700] nsmail httpd[28047]: Account Notice: close **63.228.195.111 Daniel.R.Attacker** 2000/11/15 11:11:49 0:06:57 0 0 1

{Daniel.R.Attacker has logged out of (above) and logged back into the mail server (below). Notice the IP address. One message is sent.}

[15/Nov/2000:**11:42:22** -0700] nsmail httpd[28047]: Account Information: login [**63.228.195.111**] **Daniel.R.Attacker** [null]
[15/Nov/2000:11:42:22 -0700] nsmail httpd[28047]: Account Information: login: [63.228.195.111] Daniel.R.Attacker plaintext
[15/Nov/2000:**11:44:49** -0700] nsmail httpd[28047]: General Notice: **Daniel.R.Attacker**[**63.228.195.111**] created message **<4e5c2496fb.496fb4e5c2@aaa.xxx.yyy>**
[15/Nov/2000:11:46:34 -0700] nsmail httpd[28047]: Account Debug: session_cleanup Daniel.R.Attacker:0x3069723a timeout session
[15/Nov/2000:**11:46:34** -0700] nsmail httpd[28047]: Account Notice: close **63.228.195.111** Daniel.R.Attacker 2000/11/15 9:22:16 2:24:18 0 0 1
[15/Nov/2000:**11:47:51** -0700] nsmail httpd[28047]: Account Notice: **close 63.228.195.111 Daniel.R.Attacker** 2000/11/15 11:42:22 0:05:29 0 0 1

{Daniel.R.Attacker has logged out (above) and then logged in as Richard.Victim (below). Notice the IP address. The final message in the sequence is sent.}

[15/Nov/2000:**11:54:10** -0700] nsmail httpd[28047]: Account Information: **login [63.228.195.111] Richard.Victim** [null]
[15/Nov/2000:**11:54:10** -0700] nsmail httpd[28047]: Account Information: login: [63.228.195.111] Richard.Victim plaintext
[15/Nov/2000:**11:55:56** -0700] nsmail httpd[28047]: General Notice: **Richard.Victim[63.228.195.111**] created message **<4d8f84da3c.4da3c4d8f8@aaa.xxx.yyy>**
[15/Nov/2000:**11:56:05** -0700] nsmail httpd[28047]: Account Notice: close **63.228.195.111** Richard.Victim 2000/11/15 11:54:10 0:01:55 0 0 1

{Daniel.R.Attacker logs in as himself one last time. No messages are sent. And again, notice the IP address.}

[15/Nov/2000:**11:56:13** -0700] nsmail httpd[28047]: Account Information: **login [63.228.195.111] Daniel.R.Attacker** [null]
[15/Nov/2000:11:56:13 -0700] nsmail httpd[28047]: Account Information: login: [63.228.195.111] Daniel.R.Attacker plaintext
[15/Nov/2000:**11:56:39** -0700] nsmail httpd[28047]: Account Notice: **close 63.228.195.111 Daniel.R.Attacker** 2000/11/15 11:56:13 0:00:26 0 0 1

Corroboration

The original connection to the mail server from **63.228.195.111** is corroborated by Snort analysis of tcpdump raw data. The cut below shows the initial 3-way handshake. 63.228.195.111 is connecting to the mail server. The time stamps are off by a few seconds because the systems are not running on coordinated time.

****
11/15-11:04:54.931671 63.228.195.111:1509 ->nsmail:80
TCP TTL:112 TOS:0x0 ID:9645 DF
****S***** Seq: 0x43A86B57 Ack: 0x0 Win: 0x4000
TCP Options => MSS: 1460 NOP NOP SackOK

11/15-11:04:54.931900 nsmail:80 -> **63.228.195.111:1509**
TCP TTL:255 TOS:0x0 ID:2883 DF
****S***A*** Seq: 0xB0965FF4 Ack: 0x43A86B58 Win: 0x2238
TCP Options => MSS: 1460

11/15-11:04:55.060308 63.228.195.111:1509 -> nsmail:80
TCP TTL:112 TOS:0x0 ID:9647 DF
******A*** Seq: 0x43A86B58 Ack: 0xB0965FF5 Win: 0x4470
47 45 54 20 2F 53 GET /S
****

The SMTP logfile view of the sending of the first message in the five message sequence is shown below.

****
[15/Nov/2000:**11:10:54** -0700] nsmail **smtpd**[28069]: General Notice: SMTP-Accept:G42VU600.MLT:**<4c33a4aeed.4aeed4c33a@aaa.xxx.yyy>**:[127.0.0.1]:127.0.0.1:<**Richard.Victim**@xxx.yyy>:930:1:<abc.all.hands@nsmail.aaa.xxx.yyy>
****

The relevant times and actions taken in non-log format are shown below.

****
At **11:04:07** MDT Richard.Victim logged into his email account from IP address **63.228.195.111**
At **11:10:54** MDT Richard.Victim sent a message with the Subject: New Sandy Cam
At **11:11:41** MDT Richard.Victim logged out of his email account.

At **11:11:49** MDT Daniel.R.Attacker logged into his email account from IP address **63.228.195.111**
At **11:12:50** MDT Daniel.R.Attacker sent a message with the Subject: New Sandy Cam
At **11:18:19** MDT Daniel.R.Attacker sent a message with the Subject: Re: New Sandy Cam
The above message is the message that questions the integrity of the email system.
At **11:18:46** MDT Daniel.R.Attacker logged out of his email account.

At **11:42:42** MDT Daniel.R.Attacker logged into his email account from IP address 63.228.195.111
At **11:44:49** MDT Daniel.R.Attacker sends message
At **11:47:51** MDT Daniel.R.Attacker logged out of his email account

At **11:54:10** MDT Richard.Victim logged into his email account from IP address 63.228.195.111

At **11:55:56** MDT Richard.Victim sends message
At **11:56:05** MDT Richard.Victim logged out of his email account
****

What does this attack illustrate?

1> Attacks or exploits often originate from insiders

.

2> It is extremely important to keep passwords secure.

3> Passwords need to be changed on a regular basis and should not be shared.

4> Logging, Logging, Logging. Logging is important. Without the hard evidence of log files one can often only speculate as to what really happened. In the this case the mail server HTTP logs showing account changes between UIDs in close proximity in time and from the same IP address sealed the case.

5> Multiple log files can be used to correlate attack activities. In this example, evidence of the exploit is also found in the mail server SMTP log files and in the Intrusion Detection System tcpdump files. This redundant information solidifies the case. Complete redundant logs of the incident not included for brevity.

6> You don't always know how your log files will be used. In this case the IP address 63.228.195.111 was not on a watch list and the activity engaged by attacker did not fall out side the scope of normal TCP/IP traffic. So, the corroborating evidence was buried in 540MB of Snort text data. I accessed information from the file by cat'ing the 540MB text version of the raw tcpdump file to a Perl script.

7> Systems should be time synchronized. This facilitates correlation between log files.

Assignment 3 - Analyze This

The data provided by GIAC Enterprises to Stop'M Inc., for security analysis spans the time period of 08-11-00 to 09-14-00. The data is incomplete due to many problems with the network monitoring and data collection system. The incompleteness of the data makes conclusions very difficult to draw. The bid to provide security services to GIAC Enterprises must include money sufficient to provide uninterrupted data collection.

In general the collected data show a significant (40225 alerts) amount of suspicious traffic on the GIAC Enterprises' network. This is not surprising, but it does underscore the need for greater vigilance regarding network security on the part of GIAC Enterprises.

The SOOS* files

Based on information in the SOOS* files provided by GIAC all machines on the GIAC networks MY.NET.201, .202, .203, .204, .205, .206, .210, .211, .217, .218, .219, .220, .221, .222, .223 and .226 should be considered to be compromised. A total of 55 machines spread across the networks listed above are known to have participated in conversations as shown below.

The conversations show MY.NET.xxx.xxx computers sending crafted packets to the outside world. The evidence of the packet crafting is shown in **bold**. A variety of illegal TCP flag combinations including SYN/FIN, and set reserved bits are shown. This is sufficient reason to believe that all machines on the networks specified above should be considered compromised.

An example conversation from the SOOS file provided is:

```
Initializing Network Interface ep0
snaplen = 68
Entering readback mode....
08/28-00:27:14.211201 128.194.9.94:1575 -> MY.NET.201.190:6699
TCP TTL:116 TOS:0x0 ID:41583  DF
**SFR**U Seq: 0xB3D2B34   Ack: 0x80025E   Win: 0x5010
TCP Options => EOL EOL

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/28-00:47:45.028963 128.194.9.94:1578 -> MY.NET.201.190:6699
TCP TTL:116 TOS:0x0 ID:49529  DF
2*SFRP*U Seq: 0xB3FD0DF   Ack: 0x150278   Win: 0x5010
06 2A 1A 2B 0B 3F D0 DF 00 15 02 78 02 6F 50 10  .*.+.?.....x.oP.
22 36 AF E9 00 00 15 96 DC A6 BC 6F 89 DD E3 FB  "6.........o....
A7 73                                            .s

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/28-08:06:40.609664 129.81.91.167:1763 -> MY.NET.202.174:6699
TCP TTL:111 TOS:0x0 ID:53203  DF
*1SF**A* Seq: 0x55077A   Ack: 0x96950008   Win: 0x5018
TCP Options => EOL EOL
F3 44 32 9D 8A 4B C3 3D 46 CB                    .D2..K.=F.
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/28-09:55:10.288780 MY.NET.202.202:0 -> 128.61.68.140:1694
```

```
TCP TTL:126 TOS:0x0 ID:10013  DF
2*SF*PAU Seq: 0x1A2B005F   Ack: 0x31330343   Win: 0x5010
TCP Options => Opt 32 (32): 2020 2000 0402 0101 080A 0023 0000 0000 0000 0000 0000
0000 0000 0000 0000

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/28-09:55:13.260265 MY.NET.202.202:1694 -> 128.61.68.140:6699
TCP TTL:126 TOS:0x0 ID:56350  DF
2*SF*PAU Seq: 0x5F   Ack: 0x3133034B   Win: 0x5010

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/28-09:57:57.798022 MY.NET.202.202:0 -> 152.7.56.109:1701
TCP TTL:126 TOS:0x0 ID:19359  DF
**SF**A* Seq: 0x1A2B0061   Ack: 0x44830C4D   Win: 0x5010
00 00 06 A5 1A 2B 00 61 44 83 0C 4D 07 13 50 10  .....+.aD..M..P.
07 EC 11 40 20 20 20 20 20 00                    ...@      .

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/28-09:58:34.695272 MY.NET.202.202:1708 -> 128.61.68.140:6699
TCP TTL:126 TOS:0x0 ID:48573  DF
21SF**A* Seq: 0x62   Ack: 0x78CB0326   Win: 0x5010
TCP Options => Opt 32 (32): 2020 2000 A06D 070D 6987 0014 0000 0000 0000 0000 0000
0000 0000 0000 0000 EOL EOL EOL EOL EOL EOL EOL EOL

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

The SnortA* Files

Analysis of the SnortA* files provided by GIAC shows that 40255 alerts, instances of suspicious activity, took place between 08-11-00 and 09-14-00. These 40225 events are distributed across 19 known suspicious signatures (See the table below). Keep in mind that other suspicious activities may be occurring on the MY.NET networks. The 19 signatures shown and discussed below were detected by a "standard" signature detection set. A more carefully constructed, or more restrictive, set of signatures may have detected even more suspicious traffic. GIAC must keep in mind that we may not have seen alerts for all of the suspicious traffic that has occurred on the network. This is especially true when the gaps in the collected data are considered.

The suspicious activities range from network scans/reconnaissance missions, to what might be targeted attacks against specific machines providing specific network services. The activities are directed against both Unix and Microsoft Windows based computer systems.

The following information is taken from the SnortSnarf index.html page. SnortSnarf will be discussed in section 4 of this paper.

****

SnortSnarf start page
All Snort signatures   SnortSnarf v102700.1

40225 alerts found among the files: snorta
Earliest alert at 00:33:44.374672 on 08/11
Latest alert at 23:21:39.338983 on 09/14

Signature	# Alerts	#Sources	# Destinations
Possible wu-ftpd exploit GIAC000623	2	1	2
site exec - Possible wu-ftpd exploit - GIAC000623	6	1	4
Happy 99 Virus	2	2	2
TCP SMTP Source Port Traffic	8	2	2
Tiny Fragments - Possible Hostile Activity	12	5	8
External RPC call	40	6	3
Queso fingerprint	54	11	23
Probable NMAP fingerprint attempt	67	7	28
SUNRPC highport access!	64	5	3
NMAP TCP ping!	138	10	42
Null scan!	181	63	73
SMB Name Wildcard	338	17	15
SNMP public access	922	16	1
Attempted Sun RPC high port access	1990	8	11
Watchlist 000220 IL-ISDNNET-990517	5276	19	21
SYN-FIN scan!	5457	6	3005
WinGate 1080 Attempt	6193	347	2156
Watchlist 000222 NET-NCFC	19478	45	19

SnortSnarf brought to you courtesy of Silicon Defense
Authors: Jim Hoagland and Stuart Staniford
See also the Snort Page by Marty Roesch
Page generated at Sat Nov 11 12:52:04 2000
****

A brief discussion of each of the signatures from the table above follows. Similar alerts may be grouped together.

Possible wu-ftpd exploit GIAC000623
site exec - Possible wu-ftpd exploit - GIAC000623
8 alerts from 1 source address to 6 destination addresses

These alerts were part of a 09-08-00 scan for ftp servers. The address in this alert comes from an ATHOME address block. According to CERT® Advisory CA-1999-13 there are multiple vulnerabilities in WU-FTPD If wu-ftpd is running on the GIAC network it should be patched to meet current standards. There is a good chance that these are false alerts.

The following was taken from a post to SANS regarding the wu-ftpd detect.
(Andy Johnston on the wu-ftpd!)

"After looking at the exploit code at securityfocus, I added these rules to the one in the SANS update:

alert tcp any any -> $HOME_NET 21 (msg:"site exec - Possible wu-ftpd exploit -GIAC000623"; content:"site exec";)
alert tcp any any -> $HOME_NET 21 (msg:"SITE EXEC - Possible wu-ftpd exploit - GIAC000623"; content:"SITE EXEC";)
There will be some false positives, but I'd kind of like to know when site exec commands are used anyway."

Happy 99 Virus
2 Alerts from 2 source address to 2 destination addresses.

This attack is classified as Trojan by some and by others a virus or a worm. The Happy 99 Virus (Trojan) affects Windows PCs. If the computers targeted (MY.NET.6.35 and MY.NET.179.80) are Windows based, look for ska.exe, ska.dll and wsock32.ska. These files indicate that the system has been compromised with the Happy 99 virus.

TCP SMTP Source Port Traffic
8 alerts 2 sources 2 destinations

The sources for this detect appear to be friendly. 206.46.170.21 is smtppop2pub.gte.net and 156.40.66.2 is the National Institute of Health, a U.S. government agency. The destination computers were MY.NET.97.181:25 and MY.NET.253.53:25. Check these machines to see if they are running SMTP. If they are NOT running SMTP these detects should be pursued. If they are running SMTP this is ok traffic.

Tiny Fragments - Possible Hostile Activity
12 alerts 5 sources 8 destinations

Some of the networks I believe to be compromised based on analysis of the SOOS* files were targets of this attack. The attack may have been successful. The attack was staged from computers in foreign countries including Belgium, the former Soviet Union and Poland.

Malicious fragmentation can be used to launch denial of service attacks or can be a method of network

mapping that goes undetected by some firewalls (CAN-1999-0204). The eight destination systems need to be examined to see if they were compromised.

External RPC Call

40 alerts 6 sources 3 destinations

This activity is very suspicious. Although it may not have resulted in system compromise it should be considered to be hostile until proven otherwise. Each of the GIAC systems MY.NET.6.15, MY.NET.100.130 and MY.NET.15.127 were targets of SYN/FIN or hostile activity from the machines making the RPC calls.

18.116.0.75 (MIT)	to MY.NET.6.15		
209.160.238.215 (CA)	to MY.NET.6.15	MY.NET.100.130	MY.NET.15.127
141.223.124.31 (Korea)	to MY.NET.6.15	MY.NET.100.130	
161.31.208.237 (U of Ark)	to MY.NET 6.15		
210.100.199.219 (Asia)	to MY.NET.6.15	MY.NET 100.130	
210.101.101.110 (Asia)	to MY.NET.6.15		

Is RPC from the outside world to the GIAC network required? If not, block the traffic. If this traffic is allowed verify that the source address above are allowed access to the GIAC MY.NET destinations listed above.

Queso Fingerprint

54 alerts 11 sources 23 destinations

Queso is a tool used to identify operating systems. OS determination is a first step in compromising a system. Queso is written so that information is gathered without the 3-way handshake required by TCP.

Probable NMAP fingerprint attempt

64 alerts 7 sources 28 destinations

NMAP is a tool that, among other things, can be used to identify operating systems. NMAP is capable of scanning networks using a variety of techniques. The information gathered from the scan can be used to launch more serious attacks against a finger printed system.

SUNRPC high port access!

64 alerts 5 sources 3 destinations

This sounds serious. And it might be serious. The destination computers MY.NET.211.12, MY.NET.6.15 and MY.NET.210.2 need to be checked. Of the five source addresses two are very suspicious. 193.64.205.17 is from Finland and 212.204.196.241 is from the Netherlands.

I will use this detect to raise the incomplete log file issue once again. I searched both the SnortS* and SOOS* files looking for entries relating to the possibly friendly traffic from sources 209.10.41.242 (Globix Corporation), 207.29.195.22 (Netreach Corporation) and 205.188.4.42. 205.188.4.42 is from a AOL Net-Block and the source port, 5190, is used by America Online Services. The source port from 207.29.195.22 is 2646, the AND License Manager. But, at the times that these detects were issued the SnortS* and SOOS* files have no logged information. This makes complete analysis of the situation impossible.

Finally, does GIAC Enterprises need to have these high ports open to the Internet? Probably not, so access to high ports should be restricted.

Attempted Sun RPC high port access

1990 alerts 8 sources 11 destinations

All but 4 of these detects are false positives triggered by the destination port of 32771. Ports in the range 32770 through 32900 are used by Sun Solaris RPC services. The source port trigger for the detect event was always 4000. This source port is usually an ICQ server. ICQ services have been verified on all sources addresses except 141.213.191.50 (University of Michigan) and 128.211.224.100 (Purdue University). The latter attempts should be considered suspicious and must be investigated. The suspicious activity accounts for 4 of the 1990 detects.

NMAP TCP ping!

138 alerts 10 sources 42 destinations

This is a stealthy way to identify which computers on a network are alive. A destination computer receiving the unsolicited ACK sent by the source should respond with a RESET. This RESET indicates to the hostile source that the destination computer is alive. The live computer may become a targeted system.

Null scan!

181 alerts 63 sources 73 destinations

A likely source of the null scan is NMAP. The null scan might be used to identify a computer's operating system.

SMB Name Wildcard

338 alerts 17 sources 15 destinations

Client PCs on a Windows network connect to services using NetBIOS over TCP/IP. Once that connection is established, clients communicate with servers using SMB protocol. This protocol allows clients to access Windows shares, open, read and write files. SMB (Server Message Block) name wildcard is an attempt to identify NetBIOS resources on the Windows network. Once resources are identified, resource specific exploits can be launched.

SNMP public access

922 alerts 16 sources 1 destination

This looks like a false detect to me. The alert was triggered by internal (MY.NET) traffic to destination port 161, which is the SNMP management port. The most likely explanation is that the 16 sources are servers reporting to a SNMP agent running on the destination. This should be verified.

SYN-FIN scan!

5457 alerts 6 sources 3005 destinations

The SYN-FIN scan is used to probe machines on a network. The goal is to find open ports. The attacker uses the invalid SF flag combination to elude (hopefully) detection by intrusion detection systems, or possibly to fingerprint the operating system. Linux, for example, responds to a SYN-FIN scan with a SYN-FIN-ACK. SYN-FIN is a common probe and indicates hostile activity because the SYN-FIN flag combination does not occur naturally on TCP/IP networks.

WinGate 1080 attempt

6193 alerts 347 sources 2156 destinations

The information below was taken from CERT Vulnerability Note VN-98.03

WinGate allows networked computers to simultaneously share an Internet connection. WinGate also can serves as a firewall, prohibiting intruders from accessing your network, but early (prior to v2.1) versions of WinGate have serious security flaws, and WinGate is often mis-configured. The default configuration for WinGate allows an intruder to use a WinGate server to conceal his or her true location without the need to forge packets. In particular WinGate enables all available network ports or services (this includes FTP, IRC, News, Telnet and WWW). WinGate does not log connections.

Because connections are not logged by default, and because WinGate will accept any incoming connections by default, intruders can use WinGate to launder their IP addresses during an internet-based attack. A victim of an attack using a WinGate server is only able to trace the connection back to the WinGate server. Additionally, a site running a vulnerable WinGate server may be implicated in a security incident when in fact an intruder has used the WinGate server to conceal his or her true location.

There is a chance that some of this WinGate traffic is IRC server related (See 3.5/3.6 IDS Signatures and Analysis, pg 240). But much, if not most of the traffic is not IRC related. Networks 168.187.26.157 (Kuwait) and 168.120.16.250 (Bankok) accounted for 4,426 of the 6,193 alerts.

If the GIAC Enterprises security policy allows WinGate servers they should be upgraded and configured as per information vendor information, otherwise WinGate servers should be identified and shutdown.

Watchlist 000220 IL-ISDNNET-990517

5276 alerts 19 sources 21 destinations

All of these detects came from Israeli networks. The detects from Watchlist 000222 NET-NCFC below came from Chinese networks.

MY.NET.253.41, MY.NET.253.42 and MY.NET.253.43 were very active destinations on both watch lists. Where there is smoke, there is fire. It may be true here. This looks like active targeting. I would take a close look at these machines. One question is, have they been compromised? Another is, what would someone gain if one of these machines was compromised?

Watchlist 000222 NET-NCFC

19478 alerts 45 sources 19 destinations

All of these attacks came from Chinese networks. See Watchlist 000220 above.

Assignment 4 - Analysis Process

I started the analysis process by downloading the approximately 20MB of text data to my Linux box. I manually scanned (using 'more *filename*') the contents of the different groups of files to determine exactly how the information in the files should be classified (Snort Alerts, Snort Scans, and Snort output of read back data).

I decided that I might want to manually correlate the information from a particular day across file sets so I built chart of describing which files held information from specific dates. A portion of the table created is below.

Alerts		Scans		SOOS	
SnortAle	8-11	SnortSca	8-16	SOOS	8-28
SnortA2	8-15	SnortS2	8-18	SOOS2	8-29
:		:		:	

This did not turn out to be particularly useful, but I would probably do this again anyway. I believe that this approach might have greater value with a more complete data set, that is, a data set without large holes where useful data might be found.

It was clear that hand analysis of files was not even remotely possible so I had to decide on a tool, or set of tools, to use on the data. The first tool I downloaded was Snort sort. Snort sort is a Perl script that sorts Snort alert files by alert type. To make Snort sort even remotely useful I had to build a single alerts file from the 20 alerts files that I had downloaded. I did this by cat'ing the individual files, using the append output re-director, to build a composite file I called snorta I ran Snort sort against this file and viola, it listed all of the different alerts in the composite file. Snort sort clearly indicated what alerts existed in the alerts files, but otherwise was not very useful.

Next I downloaded SnortSnarf and ran it against my composite alerts file. In order to get SnortSnarf working properly I had to change every instance of MY.NETWORK in the composite snorta file to 255.255. Any set of numbers will work here. I just happened to choose 255.255. The output of SnortSnarf agreed with the output of Snort sort. This corroboration made it easy to trust the output of SnortSnarf. The output of SnortSnarf was impressive. No question about it, SnortSnart was the primary tool that I used to analyze the data and I would have been in deep yogurt without it.

After I had SnortSnarf running I felt pretty comfortable that I had a way to deal with the huge amount of data in the Snort alerts files. But I was not sure what to do with the Snort scan files or the SOOS* read back files. So I went to the SANS WWW site and downloaded and read through some of the practical exams of previous students. One of the practical exams corroborated my SnortSnarf composite alert file approach, others talked of building databases with Windows desktop applications. I knew that I would not be doing that. It seemed like too much to learn with everything else that was going on. And I had a lot of confidence in SnortSnarf by this time.

But I still did not have a way to use the Snort scan files or the Snort SOOS* files. I spent sometime trying to manually correlate, using cat, grep and more, patterns between Snort alert files and Snort scan files. I was not successful and eventually decided, right or wrong, not to spend much time on the SnortS* scan

files. I did attempt to correlate information from the SnortA* alerts files and the SOOS* to the SnortS* files. But I never found a smoking gun.

I had already observed that the collective size of the scan files was twice that of the collective size of the Snort alerts files. My feeling was, and still is, that the interesting information collected in the scan files should also be in the alerts files. So that left just the SOOS* files. I used cat, grep and more on these files looking for obvious signs of trouble in the packet payload. Some of the strings that I searched for were 'password', 'login' and 'bind'. These are obvious indicators of potential trouble. I tried to correlate addresses of both source and destination systems in the alerts files to source and destination addresses in the SOOS* files. But, alas, these attempts to correlate failed.

I was not prepared to walk away from the SOOS* files. I was sure that some relevant information existed in these files, I just had not identified what it was. I was getting discouraged regarding how to use the SOOS* files. I had spent what seemed like a lot of time on these files. One day I was browsing the files with more and grep, and vi, no longer sure what I was looking for when I finally observed that the TCP flags for each set of transactions were bogus. This observation led to my final analysis of the SOOS* files.

After deciding how to deal with the Snort scan files and Snort SOOS* files I turned my attention back to the SnortA* alert files and the output of SnortSnarf. I printed the index.html generated by SnortSnarf and for each alert listed I scanned and re-scanned the SANS workshop textbooks for discussions of the alert. When a discussion of a particular alert was found wrote the information down on the print out. Next I surfed the web, mostly from www.google.com and www.lycos.com, for alert related information and for source and destination port related information. I made notes regarding where I found what might be relevant information.

Finally I went back and evaluated each alert. What does the alerted activity do? How does the alerted activity work? Is the alerted activity a scan, a denial of service attack, or an exploit that would allow the successful attacker to take over the machine? I looked at the destination and source addresses involved. I looked at the destination and source ports involved. I used nslookup, and the built in lookup capabilities of SnortSnarf and SHADOW to track down source IP addresses. I used the port searcher on www.snort.org to identify how ports might be used. I referenced information in SANS workshop texts and web information. Using all of the above information I wrote my analysis of the alerted activity.

Last Updated: August 17th, 2017

Upcoming Training

SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
Baltimore Fall 2017 - SEC503: Intrusion Detection In-Depth	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Scottsdale SEC503**	Scottsdale, AZ	Oct 02, 2017 - Oct 07, 2017	Community SANS
SANS October Singapore 2017	Singapore, Singapore	Oct 09, 2017 - Oct 28, 2017	Live Event
Community SANS Ottawa SEC503	Ottawa, ON	Oct 16, 2017 - Oct 21, 2017	Community SANS
SANS Berlin 2017	Berlin, Germany	Oct 23, 2017 - Oct 28, 2017	Live Event
SANS Seattle 2017	Seattle, WA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS San Diego 2017	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	Live Event
San Diego Fall 2017 - SEC503: Intrusion Detection In-Depth	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	vLive
SANS Paris November 2017	Paris, France	Nov 13, 2017 - Nov 18, 2017	Live Event
Community SANS Pensacola SEC503	Pensacola, FL	Nov 27, 2017 - Dec 02, 2017	Community SANS
SIEM & Tactical Analytics Summit & Training	Scottsdale, AZ	Nov 28, 2017 - Dec 05, 2017	Live Event
SANS Munich December 2017	Munich, Germany	Dec 04, 2017 - Dec 09, 2017	Live Event
SANS Cyber Defense Initiative 2017	Washington, DC	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS Security East 2018	New Orleans, LA	Jan 08, 2018 - Jan 13, 2018	Live Event
SANS Las Vegas 2018	Las Vegas, NV	Jan 28, 2018 - Feb 02, 2018	Live Event
SANS Dallas 2018	Dallas, TX	Feb 19, 2018 - Feb 24, 2018	Live Event
SANS Northern VA Spring - Tysons 2018	Tysons, VA	Mar 17, 2018 - Mar 24, 2018	Live Event
SANS 2018	Orlando, FL	Apr 03, 2018 - Apr 10, 2018	Live Event
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced