Global Information Assurance Certification Paper

10 detects submitted by Judith M. Ostroot for the practical examination following the Intrusion Detection course attended at SANS2000, San Jose.

DETECT 1 – LinuxConf Scan

From a Korean website for traffic information.
May 30, 2000

```
10:23:08.562718 210.112.192.74.2332 > MyNet.0.98: S 3126558814:3126558814(0) win 32120 DF)
10:23:08.562830 210.112.192.74.2340 > MyNet.8.98: S 3122269501:3122269501(0) win 32120 (DF)
10:23:08.570528 210.112.192.74.2336 > MyNet.4.98: S 3130383268:3130383268(0) win 32120 (DF)
10:23:08.577921 210.112.192.74.2339 > MyNet.7.98: S 3122901546:3122901546(0) win 32120 (DF)
10:23:08.581611 210.112.192.74.2334 > MyNet.2.98: S 3128658365:3128658365(0) win 32120 (DF)
10:23:08.587487 210.112.192.74.2341 > MyNet.9.98: S 3117590815:3117590815(0) win 32120 (DF)
10:23:08.594196 210.112.192.74.2344 > MyNet.12.98: S 3119051516:3119051516(0) win 32120 (DF)
10:23:08.599549 210.112.192.74.2343 > MyNet.11.98: S 3117975763:3117975763(0) win 32120 (DF)
10:23:08.604563 210.112.192.74.2335 > MyNet.3.98: S 3127435303:3127435303(0) win 32120 (DF)
10:23:08.611112 210.112.192.74.2338 > MyNet.6.98: S 3130195997:3130195997(0) win 32120 (DF)
10:23:08.617865 210.112.192.74.2337 > MyNet.5.98: S 3128742624:3128742624(0) win 32120 (DF)
10:23:08.695482 210.112.192.74.2749 > MyNet.14.98: S 3128646712:3128646712(0) win 32120 (DF)
10:23:08.863491 210.112.192.74.2753 > MyNet.18.98: S 3126432355:3126432355(0) win 32120 (DF)
10:23:09.446882 210.112.192.74.2773 > MyNet.20.98: S 3126286835:3126286835(0) win 32120 (DF)
10:23:09.468444 210.112.192.74.2775 > MyNet.22.98: S 3119646408:3119646408(0) win 32120 (DF)
10:23:09.470943 210.112.192.74.2778 > MyNet.25.98: S 3128495330:3128495330(0) win 32120 (DF)
```

(each IP address was actually hit twice, I removed the duplicates to save space.)

1. Source of Trace

   My network

2. Detect was generated by:

   Windump

   Explanation of fields:

   This is tcp traffic:
   10:23:11.548892 *(Timestamp)* **210.112.192.74.2332 > MyNet.0.98:** *(Source IP.port > Dest IP.port)* ***S*** *(Type of packet, S with no Ack is an initial synchronization packet)* **3126558814:3126558814(0)** *(Initial sequence number:Final Sequence number (payload))* **win 32120 (DF)** *(window size (Do not Fragment))*

3. Probability the source address was spoofed.

   Low – the attacker will want the scan responses back in order to better direct his attacks.

4. Description of Attack

   The attacker is scanning a range of IP addresses on the port (98), used by Linuxconf, looking for live hosts.

5. Attack Mechanism

   A series of IP addresses are sent initial SYN packets. If response is received from one them, the attacker knows he has a live host, which he can then attempt to exploit using Linuxconf. Linuxconf is a tool which can be used to remotely administer Linux boxes.

   This is a reconnaissance attack.

6. Correlations:

   This attack happened the day Korea Linuxconf-scanned the world. There were several traces sent in to GIAC. A couple of them were from the same address as this trace. This attack is described on Robert Graham's site (http://www.robertgraham.com/pubs/firewall-seen.html#1.1). It is also described in the SANS GIAC San Jose Coursebook 2.4 & 2.5 page 159.

7. Evidence of active targeting

   None. Due to the fact that others were scanned in exactly the same way, I believe a large range of IP addresses were scanned.

8. Severity = (Criticality of System + Lethality of Attack) – (Host Countermeasures + Network Countermeasures)

   (5 + 5) – (5 + 5) = 0

   - Critical servers were scanned
   - Attack can gain root access
   - No Linux servers
   - Restrictive firewall, only one way in or out

9. Defensive recommendations

   - Make sure any Linux servers are hardened against this type of attack.
   - Ensure firewall blocks attempts on port 98.
   - Block traffic from this site.

10. Multiple choice question:

    Linuxconf is ___.

a) a script used by hackers to scan for Linux boxes
b) on-line conferencing software for the Linux OS
c) a remote administration utility for Linux
d) none of the above

Answer: c

DETECT 2 – Impossible Flag Probe

From to www.bno.nl – the Association of Dutch Designers, Netherlands
May 26, 2000
08:21:18.726493 195.11.224.147.27035 > MyNet.27005: FP 6867710:6869170(1460) win 128 urg 20866 (DF) *(flags are R1-U-P-F)*
08:21:18.843670 195.11.224.147.27035 > MyNet.27005: FP 5522689:5524149(1460) win 128 urg 21890 (DF) *(flags are R1-U-P-F)*
08:21:23.609792 195.11.224.147.1029 > MyNet.5073: P 249079:250539(1460) ack 2016822646 win 8562 (DF) *(flags are A-P)*
08:21:25.465826 195.11.224.147.7777 > MyNet.1041: SRP 5855295:5856771(1476) win 1824 urg 64 (DF) *(flags are U-P-R-S)*
08:21:29.368757 195.11.224.147.7788 > MyNet.1212: SFP 8069222:8070702(1480) win 20 urg 136 (DF) *(flags are R1-R2-U-P-S-F)*
08:21:34.997627 195.11.224.147.1966 > MyNet.53: F 2434225:2435705(1480) win 0 (DF) *(flag is F only)*

1. Source of Trace

   My network

2. Detect was generated by:

   Windump

   Explanation of fields:

   08:21:18.726493 *(Timestamp)* **195.11.224.147.27035 > MyNet.27005:** (Source IP.port > Dest IP.port) **FP** *(Type of packet, this one is a FIN-PUSH)* **6867710:6869170(1460)** *(Initial sequence number:Final Sequence number (payload))* **win 128 urg 20866 (DF)** *(window size (urgent pointer)(Do not Fragment))*
   (flags are A-P) *(this is a field I added to list the flags for each packet)*

3. Probability the source address was spoofed.

   Low - The attacker needs the results back or the attack is useless.

4. Description of Attack

A series of packets – most with impossible flag combinations – were sent to various ports at the same address.

5. Attack Mechanism

This may be OS fingerprinting. In this type of attack, packets with a variety of possible and impossible flag combinations, and source and destination ports are sent are sent to a specific host. The responses (or lack of them) help the attacker to determine the operating system of that host.

6. Correlations:

This attack is an unknown pattern. However, there are similarities to OS fingerprinting tools. Nmap and Queso are both programs that use impossible flag combinations to make OS determinations. In SANS GIAC San Jose Coursebook 2.2 Nmap is described on page 172 and Queso on page 170.

Below are the flag combinations for nmap, Queso, and this detect.

nmap	Queso	Detect 2
SYN	SYN-ACK	FIN-PSH-URG-R1
(none)	FIN only	FIN-PSH-URG-R1
SYN-FIN-URG-PSH	FIN-ACK	PSH-ACK
ACK	SYN-FIN	SYN-RST-PSH-URG
SYN	PSH only	SYN-FIN-PSH-URG-R1-R2
ACK	SYN-R1-R2	FIN only
FIN-PSH-URG		

This attack originated in the address space of Demon Internet, Amsterdam. I've been trying to find similarities in other sets of packets received from their address space and that of Demon Internet, UK. So far, this is what I've got: the packets received are not solicited, not part of any scan, or any legitimate traffic; probes of 4 packets or more include at least 1 to a well-known port, a connection from port 27*nnn* to port 27*nnn* (often 27035 > 27005), and a connection from a port in the 77xx range (where x=x, 7766, for example).

7. Evidence of active targeting

All packets were sent to a specific IP address, which indicates previous reconnaissance.

8. Severity = (Criticality of System + Lethality of Attack) – (Host Countermeasures + Network Countermeasures)

(5 + 2) – (4 + 5) = -2

- Critical servers
- Attack can gain root access
- modern OS, most patches
- Restrictive firewall, only one way in or out

9. Defensive recommendations

- Block source IP.
- Ensure systems are hardened and have the latest updates and patches.

10. Multiple choice question:

Which of the following is a possible flag combination?
a) FIN-ACK
b) FIN only
c) SYN-FIN
d) SYN-Reserved1-Reserved2

Answer: a

DETECT 3 – SYN-FIN Scan for Zone Transfer

From the Housing And Urban Development Corporation, Tokyo, Japan
June 11, 2000.

```
08:34:25.118877 210.196.222.18.53 > MyNet.3.53: SF 907639663:907639663(0) win 1028
08:34:25.140392 210.196.222.18.53 > MyNet.4.53: SF 907639663:907639663(0) win 1028
08:34:25.164549 210.196.222.18.53 > MyNet.5.53: SF 907639663:907639663(0) win 1028
08:34:25.174844 210.196.222.18.53 > MyNet.6.53: SF 907639663:907639663(0) win 1028
08:34:25.193856 210.196.222.18.53 > MyNet.7.53: SF 907639663:907639663(0) win 1028
08:34:25.218688 210.196.222.18.53 > MyNet.8.53: SF 907639663:907639663(0) win 1028
08:34:25.242696 210.196.222.18.53 > MyNet.9.53: SF 907639663:907639663(0) win 1028
08:34:25.271702 210.196.222.18.53 > MyNet.2.53: SF 907639663:907639663(0) win 1028
08:34:25.278866 210.196.222.18.53 > MyNet.11.53: SF 907639663:907639663(0) win 1028
08:34:25.308196 210.196.222.18.53 > MyNet.12.53: SF 907639663:907639663(0) win 1028
08:34:25.341513 210.196.222.18.53 > MyNet.14.53: SF 907639663:907639663(0) win 1028
08:34:25.415109 210.196.222.18.53 > MyNet.18.53: SF 604385641:604385641(0) win 1028
08:34:25.458396 210.196.222.18.53 > MyNet.20.53: SF 604385641:604385641(0) win 1028
08:34:25.478261 210.196.222.18.53 > MyNet.21.53: SF 604385641:604385641(0) win 1028
08:34:25.489865 210.196.222.18.53 > MyNet.22.53: SF 604385641:604385641(0) win 1028
08:34:25.553628 210.196.222.18.53 > MyNet.25.53: SF 604385641:604385641(0) win 1028
08:34:25.642779 210.196.222.18.53 > MyNet.29.53: SF 604385641:604385641(0) win 1028
08:34:30.154098 210.196.222.18.53 > MyNet.255.53: SF 458276388:458276388(0) win 1028
```

1. Source of Trace

My network

2. Detect was generated by:

   Windump

   Explanation of fields:

   June 1 08:34:25.489865 *(Timestamp)* **210.196.222.18.53 > MyNet.22.53:** *(Source IP.port > DestIP.port)* **SF** *(Type of packet, these are all SYN-FIN)* **604385641:604385641(0)** *(Initial sequence number:Final Sequence number (payload))* **win 1028** *(window size)*

3. Probability the source address was spoofed.

   Low – though the packets are clearly crafted, the probability of the source being spoofed is slim because – if I am correct about the source of the scan – this scan came from a web-based tool called NetVehicle on a Japanese website. (Check it out! 210.196.222.20)

4. Description of Attack

   This attack is a SYN-FIN scan for Zone Transfers.

5. Attack Mechanism

   The attacker's goal is to download a host table from the target site with a zone transfer. This attack scans a range of IP addresses looking for a response from DNS. Packets with both the SYN and FIN flags improve the chances of getting a response because some systems will respond to SYNs and some to FINs. In addition SYN-FIN packets may not be logged by some systems. This attack provides both reconnaissance and stealth.

6. Correlations:

   Scan for Zone Transfers is described in the SANS GIAC, San Jose book 2.4 & 2.5, page 288; the SYN-FIN scan on page 289, and the SYN-FIN for Zone Transfer from port 0 on page 290.

7. Evidence of active targeting

   I don't think we were specifically targeted, more likely a range of IP addresses was selected using the tool on the NetVehicle website.

8. Severity = (Criticality of System + Lethality of Attack) – (Host Countermeasures + Network Countermeasures)

   (3 + 3) – (4 + 5) = -3

- Critical servers
- recon for possible denial of service, or host table info
- modern OS, most patches
- Restrictive firewall, only one way in or out

9. Defensive recommendations

   - Use split DNS (separate machines for internal and for public records).
   - Ensure that DNS servers are configured correctly.
   - If possible, make sure that these packets are dropped if they are addressed to any other server than the DNS server.

10. Multiple choice question:

    Zone Transfers use which protocol?
    a) icmp
    b) tcp
    c) RIP
    d) udp

    Answer: b

DETECT 4 – Scan and Probe for NetBios Information

This detect is from a public school: Schuylkill Intermediate Unit #, Marlin PA (there's a lovely picture of the school on their website) ws90.schiu.k12.pa.us

May 30, 2000

The scan:
09:42:01.068134 204.186.100.90.137 > MyNet.2.137: udp 50 (ttl 114, id 4608)
09:42:11.671353 204.186.100.90.137 > MyNet.3.137: udp 50 (ttl 114, id 6656)
09:42:11.671572 MyNet.3.137 > 204.186.100.90.137: udp 265 (ttl 128, id 25195)
09:42:21.091526 204.186.100.90.137 > MyNet.4.137: udp 50 (ttl 114, id 10240)
09:42:31.711246 204.186.100.90.137 > MyNet.5.137: udp 50 (ttl 114, id 37632)
09:42:42.143002 204.186.100.90.137 > MyNet.6.137: udp 50 (ttl 114, id 41216)
09:42:52.649034 204.186.100.90.137 > MyNet.7.137: udp 50 (ttl 114, id 19457)
09:43:03.177141 204.186.100.90.137 > MyNet.8.137: udp 50 (ttl 114, id 23809)
09:43:13.683993 204.186.100.90.137 > MyNet.9.137: udp 50 (ttl 114, id 29185)
09:43:34.736943 204.186.100.90.137 > MyNet.11.137: udp 50 (ttl 114, id 45569)
09:43:45.269862 204.186.100.90.137 > MyNet.12.137: udp 50 (ttl 114, id 63745)
09:44:06.315257 204.186.100.90.137 > MyNet.14.137: udp 50 (ttl 114, id 34818)
09:44:48.492514 204.186.100.90.137 > MyNet.18.137: udp 50 (ttl 114, id 25603)
09:45:09.515645 204.186.100.90.137 > MyNet.20.137: udp 50 (ttl 114, id 34307)

09:45:20.121123 204.186.100.90.137 > MyNet.21.137: udp 50 (ttl 114, id 37379)
09:45:30.544476 204.186.100.90.137 > MyNet.22.137: udp 50 (ttl 114, id 40707)
09:46:02.138768 204.186.100.90.137 > MyNet.25.137: udp 50 (ttl 114, id 52739)
09:46:44.378752 204.186.100.90.137 > MyNet.29.137: udp 50 (ttl 114, id 8964)

(Each IP address was hit three times, except the one that responded. I removed the duplicates to save space.)

The probe:
09:42:11.759582 204.186.100.90.1026 > MyNet.3.139: S 147819:147819(0) win 8192 <mss 1460,nop,nop,sackOK> (DF) (ttl 114, id 6912)
09:42:11.759748 MyNet.3.139 > 204.186.100.90.1026: S 608034963:608034963(0) ack 147820 win 8760 <mss 1460> (DF) (ttl 128, id 25451)
09:42:11.824117 204.186.100.90.1026 > MyNet.3.139: . ack 1 win 8760 (DF) (ttl 114, id 7168)
09:42:11.824648 204.186.100.90.1026 > MyNet.3.139: P 1:73(72) ack 1 win 8760 (DF) (ttl 114, id7424)
09:42:11.824832 MyNet.3.139 > 204.186.100.90.1026: P 1:5(4) ack 73 win 8688 (DF) (ttl 128, id 25707)
09:42:11.906078 204.186.100.90.1026 > MyNet.3.139: P 73:231(158) ack 5 win 8756 (DF) (ttl 114, id 7680)
09:42:11.906394 MyNet.3.139 > 204.186.100.90.1026: P 5:94(89) ack 231 win 8530 (DF) (ttl 128, id 25963)
09:42:11.981659 204.186.100.90.1026 > MyNet.3.139: P 231:380(149) ack 94 win 8667 (DF) (ttl 114, id 7936)
09:42:12.090024 MyNet.3.139 > 204.186.100.90.1026: . ack 380 win 8381 (DF) (ttl 128, id 26219)
09:42:14.984554 MyNet.3.139 > 204.186.100.90.1026: P 94:133(39) ack 380 win 8381 (DF) (ttl 128, id 26475)
09:42:15.131018 204.186.100.90.1026 > MyNet.3.139: F 380:380(0) ack 133 win 8628 (DF) (ttl 114, id 8704)
09:42:15.131237 MyNet.3.139 > 204.186.100.90.1026: F 133:133(0) ack 381 win 8381 (DF) (ttl 128, id 26731)
09:42:15.231637 204.186.100.90.1026 > MyNet.3.139: . ack 134 win 8628 (DF) (ttl 114, id 8960)

1. Source of Trace

   My network

2. Detect was generated by:

   Windump

   Explanation of fields:

09:42:01.068134 *(Timestamp)* **204.186.100.90.137 > MyNet.2.137:** *(Source IP.port > Dest IP.port)* **udp** *(protocol)* **50** *(payload in bytes)* **(ttl 114, id 4608)** *(time to live, ID number)*

09:42:15.131018 *(Timestamp)* **204.186.100.90.1026 > MyNet.3.139: :** *(Source IP.port > Dest IP.port)* **F** *(type of packet, F stands for Final)* **380:380(0)** *(initial sequence number:final sequence number (payload))* **ack *133*** *(acknowledgement of previous packet's sequence number)* **win 8628** *(window size)* **(DF*)*** *(Do not Fragment flag)* **(ttl 114, id 8704)** *(time to live, ID number)*

3. Probability the source address was spoofed.

   Low. The attacker needs the results back – and the second part of the attack came from the same IP address.

4. Description of Attack

   This is a host scan to find open NetBios ports.

5. Attack Mechanism

   The scan sends udp packets to NetBios on a series of IP addresses. Each address is tried three times, unless it responds. I believe this attack uses nbtstat since the connection/response of 50/265 bytes matches its profile (nbtstat is a lookup utility for NetBios information). When a host responds, a tcp session is initiated to port 139 (tcp/NetBios) at that address possibly to attempt access to shared drives.

6. Correlations:

   SANS GIAC, San Jose book 2.4 & 2.5, page 292 shows this attack. A common tool designed for this kind of attack is called Legion (from Rhino9) and is described at Robert Graham's website (http://www.robertgraham.com/pubs/firewall-seen.html#netbios), and in "Hacking Exposed" by McClure, Scambray, and Kurtz page 63.

7. Evidence of active targeting

   None. This appears to be a scan of a range of IP addresses.

8. Severity = (Criticality of System + Lethality of Attack) – (Host Countermeasures + Network Countermeasures)

   (2 + 3) – (3 + 3) = -1

   - non-critical workstation

- recon for NetBios, open shares
- mostly modern OS, most patches
- restrictive firewall, only one way in or out – but one machine outside the firewall

9. Defensive recommendations

- Filter all traffic to ports 135-139 at the perimeter.
- On stand-alone NT, disable the NetBios bindings.
- Ensure that you have the latest updates and patches.

10. Multiple choice question:

The Windows utility nbtstat connects on which port?
a) 23
b) 53
c) 137
d) 139

Answer: c

DETECT 5 - Load Balancing

Address space owned by Digex, Incorporated
June 7
Source 1

```
07:44:42.914668 199.125.178.27.45416 > MyNet.20.53: S 100777069:100777069(0) ack 100777068 win
4128 <mss 556> (ttl 246, id 0)
07:44:42.914940 199.125.178.27.45417 > MyNet.2.53: S 100777070:100777070(0) ack 100777069 win
4128 <mss 556> (ttl 246, id 0)
07:44:42.915211 199.125.178.27.45418 > MyNet.2.53: S 100777071:100777071(0) ack 100777070 win
4128 <mss 556> (ttl 246, id 0)
07:44:44.913258 199.125.178.27.45510 > MyNet.20.53: S 100777163:100777163(0) ack 100777162 win
4128 <mss 556> (ttl 246, id 0)
```

(100 similar packets omitted for clarity)

Source 2

```
07:44:42.880646 165.117.224.21.32896 > MyNet.20.53: S 38291120:38291120(0) ack 38291119 win
4128 <mss 556> (ttl 238, id 0)
07:44:42.906840 165.117.224.21.32897 > MyNet.2.53: S 38291121:38291121(0) ack 38291120 win 4128
<mss 556> (ttl 238, id 0)
07:44:42.908192 165.117.224.21.32898 > MyNet.2.53: S 38291122:38291122(0) ack 38291121 win 4128
<mss 556> (ttl 238, id 0)
07:44:44.890925 165.117.224.21.32991 > MyNet.20.53: S 38291215:38291215(0) ack 38291214 win
4128 <mss 556> (ttl 238, id 0)
```

(100 similar packets omitted for clarity)

1. Source of Trace

   My network

2. Detect was generated by:

   Windump

   Explanation of fields:

   07:44:42.906840 *(Timestamp)* **165.117.224.21.32897 > MyNet.2.53: :** *(Source IP.port > Dest IP.port)* **S** *(type of packet, S stands for Synchronization)* **38291121:38291121(0)** *(initial sequence number:final sequence number (payload))* **ack 38291120** *(acknowledgement of previous packet's sequence number)* **win 4128** *(window size)* **<mss 556>** *(maximum segment size)* **(ttl 238, id 0)** *(time to live, ID number)*

3. Probability the source address was spoofed.

   None. This traffic is from a load balancing system.

4. Description of Attack

   A large number of unsolicited, crafted SYN-ACK packets are sent from two separate load balancing servers to DNS on target name servers.

5. Attack Mechanism

   After a new connection is made to a website, load balancing software springs into action and flings crafted SYN-ACK packets at the sender's DNS servers from two different machines. The purpose may be to test response times, and provide users with the most efficient or quickest route.

6. Correlations:

   When I first saw these packets, I thought someone had spoofed my network address in a denial of service attack, and that we were receiving the responses from the victim. Then I noticed each packet had an id of 0 and the ack value was always 1 less than the synchronization value in the same packet. The pattern of these packets matches exactly the ones described in the last trace in Richard Bejtlich's paper "Interpreting Network Traffic: A Network Intrusion Detector's Look at Suspicious Events" (http://bejtlich.home.texas.net) – right down to the identical sequence/ack number pattern, window size, maximum segment size, and id; and probably run on the same OS (Solaris 2.x, judging by the TTLs).

7. Evidence of active targeting

These packets were sent after our site contacted their site (www.enliven.com, a web advertising company).

8. Severity = (Criticality of System + Lethality of Attack) – (Host Countermeasures + Network Countermeasures)

   (5 + 0) – (4 + 5) = -4

   - critical servers
   - not really an attack
   - modern OS, most patches
   - restrictive firewall, only one way in or out

9. Defensive recommendations

   - None needed.

10. Multiple choice question:

    All crafted packets are hostile attacks
    True
    False

    Answer: False

DETECT 6 – Ping Sweep

May 25, 2000
07:30:52.461907 My.network > 0.0.127.43: icmp: echo request
07:30:53.484881 My.network > 0.0.127.44: icmp: echo request
07:30:54.464877 My.network > 0.0.127.44: icmp: echo request
07:30:54.987206 My.network > 0.0.127.45: icmp: echo request
07:30:55.967172 My.network > 0.0.127.45: icmp: echo request
07:30:56.990452 My.network > 0.0.127.46: icmp: echo request
07:30:57.971374 My.network > 0.0.127.46: icmp: echo request
07:30:58.492310 My.network > 0.0.127.47: icmp: echo request
07:30:59.472302 My.network > 0.0.127.47: icmp: echo request
07:31:00.496076 My.network > 0.0.127.48: icmp: echo request
07:31:01.523811 My.network > 0.0.127.48: icmp: echo request
07:31:02.498564 My.network > 0.0.127.49: icmp: echo request
07:31:03.478300 My.network > 0.0.127.49: icmp: echo request
07:31:04.000859 My.network > 0.0.127.50: icmp: echo request
07:31:04.980902 My.network > 0.0.127.50: icmp: echo request
07:31:06.004393 My.network > 0.0.127.51: icmp: echo request

07:31:06.984830 My.network > 0.0.127.51: icmp: echo request
.
.
.
08:28:48.206861 My.network > 0.0.135.20: icmp: echo request
08:28:49.188290 My.network > 0.0.135.20: icmp: echo request
08:28:50.211262 My.network > 0.0.135.21: icmp: echo request
08:28:51.190530 My.network > 0.0.135.21: icmp: echo request
08:28:51.712530 My.network > 0.0.135.22: icmp: echo request
08:28:51.736572 My.network > 0.0.135.22: icmp: echo request
08:28:52.714592 My.network > 0.0.135.23: icmp: echo request
08:28:53.694921 My.network > 0.0.135.23: icmp: echo request
08:28:54.217725 My.network > 0.0.135.24: icmp: echo request
08:28:55.207995 My.network > 0.0.135.24: icmp: echo request
08:28:55.718880 My.network > 0.0.135.25: icmp: echo request

1. Source of Trace

   My network

2. Detect was generated by:

   Windump

   Explanation of fields:

   08:28:49.188290 *(Timestamp)* **My.network > 0.0.135.20:** *(Source IP > Dest IP)*
   icmp: *(protocol)* **echo request** *(type of packet)*

3. Probability the source address was spoofed.

   None. It was proxied and actually originated from my own network.

4. Description of Attack

   A range of IP addresses is systematically pinged.

5. Attack Mechanism

   The attacker pings a series of IP addresses, looking for machines that will send echo replies. Generally, the goal is to get a list of live hosts upon which to focus attacks. In this case, it was determined that the goal was to map our internal network.

6. Correlations:

Our firewall guy (Ron) noticed this attack on the firewall logs and tracked it to an application developer's workstation. This developer had installed Microsoft's Visio Enterprise 2000 and fired up a feature called "AutoDiscovery and Layout" which allows you to select a range of IP addresses to scan. Visio Enterprise runs the scan and creates diagram with the results. The developer had selected the entire range of IP addresses and let it run for about an hour and a half before our little talk with him.

As an added bonus, we discovered that egress filtering was not blocking the entire range of internal addresses, and have since updated our filters.

This type of scan is described in the SANS GIAC San Jose course book 2.4 & 2.5, page 278.

7. Evidence of active targeting

   This scan was run from inside our network.

8. Severity = (Criticality of System + Lethality of Attack) – (Host Countermeasures + Network Countermeasures)

   (0 + 2) – (4 + 1) = -3

   - actual address range never reached
   - recon for live hosts
   - modern OS, most patches
   - occurred inside our network

9. Defensive recommendations

   - Keep an eye on the internal firewall logs.

10. Multiple choice question:

    Which of the following IP addresses is in public address space?
    a) 10.218.92.6
    b) 172.29.218.11
    c) 192.168.2.241
    d) none of the above

    Answer: d

DETECT 7 – Broadcast Scan

From Pronet in Binghamton, NY.
June 10
17:19:01.686633 12.23.45.92 > MyNet.255: icmp: echo request
17:19:01.688133 12.23.45.92 > MyNet.0: icmp: echo request
21:05:14.233406 12.23.45.92 > MyNet.255: icmp: echo request
21:05:14.235144 12.23.45.92 > MyNet.0: icmp: echo request

1. Source of Trace

   My network

2. Detect was generated by:

   Windump

   Explanation of fields:

   17:19:01.686633 *(Timestamp)* **12.23.45.92 > MyNet.255:** *(Source IP > Dest IP)* **icmp:** *(protocol)* **echo request** *(type of packet)*

3. Probability the source address was spoofed.

   Low. The attacker will need the responses back to map the network.

4. Description of Attack

   Echo requests are sent to the old BSD zero broadcast (which UNIX systems may respond to) as well as the 255 broadcast address.

5. Attack Mechanism

   The attacker sends these packets in an attempt to map a network. If the router receiving the broadcast echo request actually broadcasts to every machine within a network, the echo returns give the attacker a list of live hosts within that network. He can then focus future attacks on live hosts.

   This is a reconnaissance attack.

   Note: There are at least 2 other possibilities:
   1) the attacker is looking for Smurf amplifiers - if there had been many more of these packets, it would suggest we were being used in a smurf attack, and then the source address would certainly have been spoofed

2) this is a response-time test – there are websites which provide internet response statistics by periodically pinging servers and comparing the responses to previous tests (example at http://www.netstatsys.com/home/internet.php).

6. Correlations:

This type of scan is described in the SANS GIAC San Jose coursebook 2.4 & 2.5, page 283. It is also described on Robert Graham's website (http://www.robertgraham.com/pubs/firewall-seen.html#netbios).

7. Evidence of active targeting

None. A large range of IP addresses was probably sent these packets. This is the only one we would see.

8. Severity = (Criticality of System + Lethality of Attack) – (Host Countermeasures + Network Countermeasures)

(3 + 2) – (4 + 5) = -4

- the ping was broadcast
- recon for live hosts
- modern OS, most patches
- restrictive firewall, only one way in or out

9. Defensive recommendations

- Configure external servers to drop echo requests.
- Configure router so that it doesn't forward broadcast requests.

10. Multiple choice question:

Which icmp packet cannot be a response to a ping?
a) echo reply
b) echo request
c) host unreachable
d) time exceeded in-transit

Answer: b

DETECT 8 – Hack'a'tack Trojan Scan

June 10
01:53:21.677603 Bad.guys.R.us.31790 > MyNet.2.31789: udp 1

01:53:21.865817 Bad.guys.R.us.31790 > MyNet.3.31789: udp 1
01:53:21.875126 Bad.guys.R.us.31790 > MyNet.4.31789: udp 1
01:53:21.876893 Bad.guys.R.us.31790 > MyNet.20.31789: udp 1
01:53:21.886023 Bad.guys.R.us.31790 > MyNet.5.31789: udp 1
01:53:21.909851 Bad.guys.R.us.31790 > MyNet.22.31789: udp 1
01:53:21.926830 Bad.guys.R.us.31790 > MyNet.6.31789: udp 1
01:53:21.934264 Bad.guys.R.us.31790 > MyNet.7.31789: udp 1
01:53:21.940158 Bad.guys.R.us.31790 > MyNet.8.31789: udp 1
01:53:21.945936 Bad.guys.R.us.31790 > MyNet.9.31789: udp 1
01:53:21.951882 Bad.guys.R.us.31790 > MyNet.11.31789: udp 1
01:53:21.956926 Bad.guys.R.us.31790 > MyNet.12.31789: udp 1
01:53:21.961951 Bad.guys.R.us.31790 > MyNet.14.31789: udp 1
01:53:21.970385 Bad.guys.R.us.31790 > MyNet.18.31789: udp 1
01:53:21.975427 Bad.guys.R.us.31790 > MyNet.21.31789: udp 1
01:53:21.981195 Bad.guys.R.us.31790 > MyNet.25.31789: udp 1
01:53:21.987269 Bad.guys.R.us.31790 > MyNet.29.31789: udp 1
01:53:28.249231 Bad.guys.R.us.31790 > MyNet.255.31789: udp 1

(We received 6 identical scans within 48 hours from 5 different IP addresses within the same network. Maybe a dial-up user?)

1. Source of Trace

   My network

2. Detect was generated by:

   Windump

   Explanation of fields:

   01:53:21.934264 *(Timestamp)* **Bad.guys.R.us.31790 > MyNet.7.31789: :** *(Source IP.port > Dest IP.port)* **udp** *(protocol)* **1** *(payload in bytes)*

3. Probability the source address was spoofed.

   Low. These scans came from the same network, and apparently from behind the same firewall. The last hop in the traceroute was the same, and the final steps in the traceroute timed out – no response came back from the source host.

4. Description of Attack

   Since this was the only traffic from this site, I'd have to say it's hostile and while we may not have been targeted specifically, we were scanned for a specific trojan.

5. Attack Mechanism

A range of IP addresses is systematically checked for response at port 31789 from port 31790. The broadcast address is included, perhaps in hopes that this may be forwarded to internal addresses. If a machine infected with the Hack'a'Tack trojan receives this packet, it will respond.

The Hack'a'Tack trojan is a remote administration tool for Windows 95/98 and possibly Windows NT machines. It can give the attacker total control of the victim machine.

6. Correlations:

A description of the attack is found at:
http://www.xploiter.com/security/hackattack.html

7. Evidence of active targeting

It's likely that this is simply a scan of a range of IP addresses, but it is disquieting that all 5 attack machines were at IP addresses owned by a company with whom we are negotiating a business partnership.

8. Severity = (Criticality of System + Lethality of Attack) – (Host Countermeasures + Network Countermeasures)

(5 + 5) – (4 + 5) = 1

- critical servers scanned
- attack can gain full admin of target
- modern OS, most patches, virus signatures up-to-date
- restrictive firewall, only one way in or out

9. Defensive recommendations

- Ensure virus profiles are up-to-date, and that virus scans are run regularly.
- Watch this range of source IP addresses closely.

10. Multiple choice question:

An unsuspecting user can install a trojan by ___.
a) receiving an e-mail containing the trojan
b) running a new game executable they just got from a friend
c) hitting ctrl-alt-del simultaneously
d) none of the above

Answer: b

DETECT 9 – FTP Connection Attempt

From www.cassvillesd.k12.wi.us (Cassville High School, WI)
(Who's at school at 4:38 am on a Monday morning?)

June 12, 2000
04:38:05.678220 216.56.42.3.2575 > MyNet.3.21: S 6909443:6909443(0) win 8192 <mss 1460> (DF)
04:38:05.678396 MyNet.3.21 > 216.56.42.3.2575: R 0:0(0) ack 6909444 win 0
04:38:06.247393 216.56.42.3.2575 > MyNet.3.21: S 6909443:6909443(0) win 8192 <mss 1460> (DF)
04:38:06.247487 MyNet.3.21 > 216.56.42.3.2575: R 0:0(0) ack 1 win 0
04:38:06.773638 216.56.42.3.2575 > MyNet.3.21: S 6909443:6909443(0) win 8192 <mss 1460> (DF)
04:38:06.773788 MyNet.3.21 > 216.56.42.3.2575: R 0:0(0) ack 1 win 0
04:38:08.558069 216.56.42.3.2575 > MyNet.3.21: S 6909443:6909443(0) win 8192 <mss 1460> (DF)
04:38:08.558223 MyNet.3.21 > 216.56.42.3.2575: R 0:0(0) ack 1 win 0
04:38:23.412463 216.56.42.3.2583 > MyNet.5.21: S 6909524:6909524(0) win 8192 <mss 1460> (DF)
04:38:23.412619 MyNet.5.21 > 216.56.42.3.2583: R 0:0(0) ack 6909525 win 0
04:38:23.945332 216.56.42.3.2583 > MyNet.5.21: S 6909524:6909524(0) win 8192 <mss 1460> (DF)
04:38:23.945472 MyNet.5.21 > 216.56.42.3.2583: R 0:0(0) ack 1 win 0
04:38:24.502116 216.56.42.3.2583 > MyNet.5.21: S 6909524:6909524(0) win 8192 <mss 1460> (DF)
04:38:24.502243 MyNet.5.21 > 216.56.42.3.2583: R 0:0(0) ack 1 win 0
04:38:25.039899 216.56.42.3.2583 > MyNet.5.21: S 6909524:6909524(0) win 8192 <mss 1460> (DF)
04:38:25.040039 MyNet.5.21 > 216.56.42.3.2583: R 0:0(0) ack 1 win 0

1. Source of Trace

   My network

2. Detect was generated by:

   Windump

   Explanation of fields:

   04:38:05.678220 *(Timestamp)* **216.56.42.3.2575 > MyNet.3.21:** *(Source IP.port > Dest IP.port)* **S** *(type of packet, S stands for Synchronization)*
   6909443:6909443(0) win 8192 *(window size)* **<mss 1460>** *(maximum segment size)* **(DF)** (Do not Fragment)

3. Probability the source address was spoofed.

   Low. The attacker needs the response back to complete the ftp session.

4. Description of Attack

   Tcp SYN packets are sent to a target machine.

5. Attack Mechanism

The attacker attempts to establish an ftp session with the target machine. This is one of the most common attacks. If the attacker locates an open server that permits anonymous connections, he can use it to transfer scripts, pirated software, and porn. Also, on poorly configured servers everything may be available to anonymous users, including config and password files, which may allow the server to be compromised.

6. Correlations:

   This attack is described at: http://www.robertgraham.com/pubs/firewall-seen.html#1.1, and in "Hacking Exposed" by McClure, Scambray, and Kurtz, page 223.

7. Evidence of active targeting

   Attempts were made on both servers that host web sites for one of our subsidiaries.

8. Severity = (Criticality of System + Lethality of Attack) – (Host Countermeasures + Network Countermeasures)

   (3 + 4) – (4 + 5) = -2

   - non-critical web servers
   - possible to gain config files
   - modern OS, most patches
   - restrictive firewall, only one way in or out

9. Defensive recommendations

   - Do not run an ftp server without evaluating other options.
   - Make sure that you have all the latest updates and patches.
   - Do not allow anonymous access.
   - If you must allow anonymous, minimize the number of directories available.

10. Multiple choice question:

    Hackers/crackers seek 'open-anonymous' ftp servers in order to ___:?
    a) share stolen software
    b) attempt to compromise the operating system
    c) steal password files
    d) all of the above

    Answer: d

DETECT 10 - Crafted Packet Scan for Proxies

From an address range owned by Uonumanet, Ltd. Japan
June 10
14:39:28.005812 202.235.50.12.65535 > MyNet.2.8080: S 3173646336:3173646336(0) win 512 (ttl 238, id 48426)
14:39:28.052452 202.235.50.12.65535 > MyNet.4.8080: S 3173646336:3173646336(0) win 512 (ttl 238, id 48426)
14:39:28.087081 202.235.50.12.65535 > MyNet.5.8080: S 3173646336:3173646336(0) win 512 (ttl 238, id 48426)
14:39:28.097208 202.235.50.12.65535 > MyNet.6.8080: S 3173646336:3173646336(0) win 512 (ttl 238, id 48426)
14:39:28.136858 202.235.50.12.65535 > MyNet.8.8080: S 3173646336:3173646336(0) win 512 (ttl 238, id 48426)
14:39:28.210940 202.235.50.12.65535 > MyNet.14.8080: S 3173646336:3173646336(0) win 512 (ttl 238, id 48426)
14:39:28.256103 202.235.50.12.65535 > MyNet.16.8080: S 3173646336:3173646336(0) win 512 (ttl 238, id 48426)
14:39:28.336439 202.235.50.12.65535 > MyNet.19.8080: S 3173646336:3173646336(0) win 512 (ttl 238, id 48426)
14:39:28.365069 202.235.50.12.65535 > MyNet.20.8080: S 3173646336:3173646336(0) win 512 (ttl 238, id 48426)
14:39:28.404285 202.235.50.12.65535 > MyNet.25.8080: S 3173646336:3173646336(0) win 512 (ttl 238, id 48426)

1. Source of Trace

   My network

2. Detect was generated by:

   Windump

   Explanation of fields:

   14:39:28.052452 *(Timestamp)* **202.235.50.12.65535 > MyNet.4.8080:** *(Source IP.port > Dest IP.port)* **S** *(type of packet, S stands for Synchronization)* **3173646336:3173646336(0)** *(Initial sequence number:Final Sequence number (payload))* **win 512** *(window size)* **(ttl 238, id 48426)** *(time to live, ID number)*

3. Probability the source address was spoofed.

   Low. It is likely the attacker wants the results back.

4. Description of Attack

In this case a crafted packet (note the sequence numbers, unusual source port (65535), and packet id never change) is sent to a range of IP addresses at port 8080.

5. Attack Mechanism

   The attacker scans a range of IP addresses hoping to find open proxy servers to use for anonymous connections. Anonymous connections are desirable because they allow hackers/crackers to hide their identities and make their attacks untraceable.

   Proxy servers commonly use port 8080 for connections as well as ports 80, 81, 1080, 8000, and 3128.

   This is a reconnaissance attack.

6. Correlations:

   This exact scan, from the same source IP address was used as an example by Donald McLachlan in SANS GIAC's 'Detects Analyzed' June 1, 2000 1000 (http://www.sans.org/y2k/060100.htm).

   This attack is described at: http://www.robertgraham.com/pubs/firewall-seen.html#5.3, and in "Hacking Exposed" by McClure, Scambray, and Kurtz, page 332.

7. Evidence of active targeting

   None. It's likely a large range of IP addresses was scanned.

8. Severity = (Criticality of System + Lethality of Attack) – (Host Countermeasures + Network Countermeasures)

   (5 + 3) – (4 + 5) = -1

   - critical servers scanned
   - possible use as a proxy
   - modern OS, most patches
   - restrictive firewall, only one way in or out

9. Defensive recommendations

   - Ensure firewall permits only inside addresses to initiate proxy connections.

10. Multiple choice question:

Which of the following are proxy protocols?

a) SOCKS
b) MITTENS
c) OCTOPUS
d) SQUID
e) a & b
f) a & d

Answer: f

Last Updated: May 29th, 2017

Upcoming Training

Security Operations Center Summit & Training	**Washington, DC**	**Jun 05, 2017 - Jun 12, 2017**	**Live Event**
SANS Houston 2017	**Houston, TX**	**Jun 05, 2017 - Jun 10, 2017**	**Live Event**
SANS Columbia, MD 2017	**Columbia, MD**	**Jun 26, 2017 - Jul 01, 2017**	**Live Event**
SANS London July 2017	**London, United Kingdom**	**Jul 03, 2017 - Jul 08, 2017**	**Live Event**
SANSFIRE 2017	**Washington, DC**	**Jul 22, 2017 - Jul 29, 2017**	**Live Event**
SANSFIRE 2017 - SEC503: Intrusion Detection In-Depth	**Washington, DC**	**Jul 24, 2017 - Jul 29, 2017**	**vLive**
SANS San Antonio 2017	**San Antonio, TX**	**Aug 06, 2017 - Aug 11, 2017**	**Live Event**
SANS Boston 2017	**Boston, MA**	**Aug 07, 2017 - Aug 12, 2017**	**Live Event**
SANS Virginia Beach 2017	**Virginia Beach, VA**	**Aug 21, 2017 - Sep 01, 2017**	**Live Event**
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**
SANS vLive - SEC503: Intrusion Detection In-Depth	**SEC503 - 201709,**	**Sep 11, 2017 - Oct 18, 2017**	**vLive**
Baltimore September 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 Boston SEC503	**Boston, MA**	**Oct 09, 2017 - Oct 14, 2017**	**Community SANS**
Community SANS Ottawa SEC503	**Ottawa, ON**	**Oct 16, 2017 - Oct 21, 2017**	**Community SANS**
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**
SIEM & Tactical Analytics Summit & Training	**Scottsdale, AZ**	**Nov 28, 2017 - Dec 05, 2017**	**Live Event**
SANS Cyber Defense Initiative 2017	**Washington, DC**	**Dec 12, 2017 - Dec 19, 2017**	**Live Event**
SANS OnDemand	**Online**	**Anytime**	**Self Paced**
SANS SelfStudy	**Books & MP3s Only**	**Anytime**	**Self Paced**