Global Information Assurance Certification Paper

George Huang – GIAC Practical for GIAC Certification – SANS 2000 San Jose

Detect 1

Apr 4 02:31:39 NONWEBBOX2 unix: ID=TCP, FROM=WEBPROBE1, TO=200.200.9.11, PORT=80, PROT=tcp, MSG=Unused Port Probe
Apr 4 02:33:31 NONWEBBOX2 unix: ID=TCP, FROM=WEBPROBE1, TO=200.200.9.11, PORT=80, PROT=tcp, MSG=Unused Port Probe
Apr 4 02:35:49 NONWEBBOX1 unix: ID=TCP, FROM=WEBPROBE1, TO=200.200.009.009, PORT=80, PROT=tcp, MSG=Unused Port Probe
Apr 4 02:41:01 NONWEBBOX2 unix: ID=TCP, FROM=WEBPROBE1, TO=200.200.9.11, PORT=80, PROT=tcp, MSG=Unused Port Probe
Apr 4 02:46:08 NONWEBBOX1 unix: ID=TCP, FROM=WEBPROBE1, TO=200.200.009.009, PORT=80, PROT=tcp, MSG=Unused Port Probe
Apr 4 02:51:20 NONWEBBOX2 unix: ID=TCP, FROM=WEBPROBE1, TO=200.200.9.11, PORT=80, PROT=tcp, MSG=Unused Port Probe
Apr 4 02:57:23 NONWEBBOX1 unix: ID=TCP, FROM=WEBPROBE1, TO=200.200.009.009, PORT=80, PROT=tcp, MSG=Unused Port Probe
Apr 4 03:01:38 NONWEBBOX2 unix: ID=TCP, FROM=WEBPROBE1, TO=200.200.9.11, PORT=80, PROT=tcp, MSG=Unused Port Probe
Apr 4 03:01:38 NONWEBBOX2 unix: ID=TCP, FROM=WEBPROBE1, TO=200.200.9.11, PORT=80, PROT=tcp, MSG=Unused Port Probe
Apr 4 03:07:41 NONWEBBOX1 unix: ID=TCP, FROM=WEBPROBE1, TO=200.200.009.009, PORT=80, PROT=tcp, MSG=Unused Port Probe

1. Source of trace

Our external network.

2. Detect was generated by:

Firewall port probe alert logs
Log Format: Date – Time – Destination Box – Protocol – Source Box – Destination IP – Destination Port – Protocol – Alert Message
NONWEBBOX2=200.200.9.11
NONWEBBOX2=200.200.9.9

3. Probability the source address was spoofed

Low - Since the results of probing our DMZ for web daemons would not get back to the probing box, it is unlikely the source address was spoofed.

4. Description of attack:

Host scan (perhaps against our whole class C or portions thereof) for running web server daemons. Since the times between probes are separated by a few minutes, this led me to believe the whole class C was being scanned.. while only these boxes were set to "complain" when unused ports were scanned.

5. Attack mechanism:

A scan such as this scans multiple machines within a range of IP addresses. Probing port 80 and waiting for response indicates intent to locate the running web server daemons. This data can be used as a part of mapping an organization's DMZ (IP1 and IP2 are web servers, IP3 is DNS, etc.)

6. Correlations:

CVE-1999-0267 - Buffer overflow in NCSA HTTP daemon v1.3 allows remote command execution.
CVE-1999-0744 - Buffer overflow in Netscape Enterprise Server and FastTrask Server allows remote attackers to gain privileges via a long HTTP GET request.
The above vulnerabilities (and many, many more) could be exploited once the appropriate web servers and web server platforms were found.

7. Evidence of active targeting:

This reconnaissance attempt tries multiple hosts to find targets for active targeting later.

8. Severity:

(Critical 3 + Lethal 5) – (System 5 + Net Countermeasures 4) = -1

9. Defensive recommendation:

The defenses are fine for such an attack. The unused port probes were dropped (and detected). Recommend watching the particular IP's for repeated activity.

10. Multiple choice test question: The port scans above are looking for:

a) FTP Servers
b) Web (HTTP) Servers
c) SSH Servers
d) DNS Servers

Answer: b

Detect 2

```
18:25:05.068811 192.168.201.222 > A1818.our.local.net: icmp: echo request
18:25:05.068949 A1818.our.local.net > 192.168.201.222: icmp: echo reply
18:25:05.068866 NW500600.our.local.net > A1818.our.local.net: icmp: echo request
18:25:05.069032 A1818.our.local.net > NW500600.our.local.net: icmp: echo reply
18:25:05.068940 192.168.201.220 > A1818.our.local.net: icmp: echo request
18:25:05.069127 A1818.our.local.net > 192.168.201.220: icmp: echo reply
18:25:05.068998 192.168.201.222.53500 > A1818.our.local.net.www: . ack 2841295578 win 2048
18:25:05.069208 A1818.our.local.net.www > 192.168.201.222.53500: R 2841295578:2841295578(0) win
0
18:25:05.069121 NW500600.our.local.net.53500 > A1818.our.local.net.www: . ack 2841295578 win 2048
18:25:05.069261 A1818.our.local.net.www > NW500600.our.local.net.53500: R
2841295578:2841295578(0) win 0
18:25:05.069183 192.168.201.220.53500 > A1818.our.local.net.www: . ack 2841295578 win 2048
18:25:05.069309 A1818.our.local.net.www > 192.168.201.220.53500: R 2841295578:2841295578(0) win
0
18:25:05.379513 192.168.201.222.53480 > A1818.our.local.net.44: udp 0
18:25:05.379598 A1818.our.local.net > 192.168.201.222: icmp: A1818.our.local.net udp port 44
unreachable [tos 0xc0]
18:25:05.379579 NW500600.our.local.net.53480 > A1818.our.local.net.44: udp 0
18:25:05.379664 A1818.our.local.net > NW500600.our.local.net: icmp: A1818.our.local.net udp port 44
unreachable [tos 0xc0]
18:25:05.379636 192.168.201.220.53480 > A1818.our.local.net.44: udp 0
18:25:05.379710 A1818.our.local.net > 192.168.201.220: icmp: A1818.our.local.net udp port 44
unreachable [tos 0xc0]
```

18:25:05.379738 192.168.201.222.53480 > A1818.our.local.net.45: udp 0
18:25:05.379772 A1818.our.local.net > 192.168.201.222: icmp: A1818.our.local.net udp port 45 unreachable [tos 0xc0]
18:25:05.380279 NW500600.our.local.net.53480 > A1818.our.local.net.45: udp 0
18:25:05.380345 A1818.our.local.net > NW500600.our.local.net: icmp: A1818.our.local.net udp port 45 unreachable [tos 0xc0]
18:25:05.380339 192.168.201.220.53480 > A1818.our.local.net.45: udp 0
18:25:05.380392 A1818.our.local.net > 192.168.201.220: icmp: A1818.our.local.net udp port 45 unreachable [tos 0xc0]
18:25:05.380404 192.168.201.222.53480 > A1818.our.local.net.20: udp 0
18:25:05.380452 A1818.our.local.net > 192.168.201.222: icmp: A1818.our.local.net udp port 20 unreachable [tos 0xc0]
18:25:05.380521 NW500600.our.local.net.53480 > A1818.our.local.net.20: udp 0
18:25:05.380555 A1818.our.local.net > NW500600.our.local.net: icmp: A1818.our.local.net udp port 20 unreachable [tos 0xc0]
18:25:05.380588 192.168.201.220.53480 > A1818.our.local.net.20: udp 0
18:25:05.380624 A1818.our.local.net > 192.168.201.220: icmp: A1818.our.local.net udp port 20 unreachable [tos 0xc0]

1. Source of trace

   Our local network.

2. Detect was generated by:

   Tcpdump with filters.
   A1818: Our linux box running filters
   NW500600: The box scanning us

3. Probability the source address was spoofed

   There is a good likelihood two of the addresses listed were spoofed as decoys. Unfortunately for them, the IP's chosen were invalid and came back unreachable.. so it is easy to see which the real machine is.
   Decoy IP's: 192.168.201.220, 192.168.201.222

4. Description of attack:

   Nmap (or nmap-similar) portscan. Tool is likely Nmap, as it's behavior of being able to do UDP scans and supporting decoys is shown here.

5. Attack mechanism:

   Nmap scans will perform an ICMP ping to ensure the target is alive, and then begins "walking" down the list of ports it wants to test. We see UDP packets in general instead of TCP, so a stealth option was probably used. The results of such a scan will inform the user of open ports on the target, based on the ports that respond to the packets.

6. Correlations:

   Port scans are commonly reported on GIAC. Examples include:
   http://www.sans.org/y2k/052800-1100.htm
   and
   http://www.sans.org/y2k/052300-0800.htm

7. Evidence of active targeting:

   High - This scan was not seen on the rest of the subnet, so this machine being an active target is likely.

8. Severity:

   (Critical 2 + Lethal 2) – (System 4 + Net Countermeasures 0) = 0

9. Defensive recommendation:

   An nmap I ran on this box showed ftp and www services running that were not used. Recommend Auditing and shutting down unused services to protect against unnecessary holes.

10. A good tool for mapping the open ports on a system is:
    a) nmap
    b) Loki
    c) pepsi
    d) named

    Answer: a

Detect 3

```
18:07:45.694883 192.168.200.200 > 256.256.201.44: icmp: echo request (frag 4321:1480@0+)
18:07:45.696132 192.168.200.200 > 256.256.201.44: (frag 4321:1480@1480+)
18:07:45.697447 192.168.200.200 > 256.256.201.44: (frag 4321:1480@2960+)
18:07:45.698680 192.168.200.200 > 256.256.201.44: (frag 4321:1480@4440+)
18:07:45.699908 192.168.200.200 > 256.256.201.44: (frag 4321:1480@5920+)
18:07:45.701138 192.168.200.200 > 256.256.201.44: (frag 4321:1480@7400+)
18:07:45.702364 192.168.200.200 > 256.256.201.44: (frag 4321:1480@8880+)
18:07:45.703683 192.168.200.200 > 256.256.201.44: (frag 4321:1480@10360+)
18:07:45.704912 192.168.200.200 > 256.256.201.44: (frag 4321:1480@11840+)
18:07:45.706325 192.168.200.200 > 256.256.201.44: (frag 4321:1480@13320+)
18:07:45.707551 192.168.200.200 > 256.256.201.44: (frag 4321:1480@14800+)
18:07:45.708894 192.168.200.200 > 256.256.201.44: (frag 4321:1480@16280+)
18:07:45.710124 192.168.200.200 > 256.256.201.44: (frag 4321:1480@17760+)
18:07:45.711355 192.168.200.200 > 256.256.201.44: (frag 4321:1480@19240+)
18:07:45.712926 192.168.200.200 > 256.256.201.44: (frag 4321:1480@20720+)
18:07:45.714156 192.168.200.200 > 256.256.201.44: (frag 4321:1480@22200+)
18:07:45.715387 192.168.200.200 > 256.256.201.44: (frag 4321:1480@23680+)
18:07:45.716617 192.168.200.200 > 256.256.201.44: (frag 4321:1480@25160+)
18:07:45.718018 192.168.200.200 > 256.256.201.44: (frag 4321:1480@26640+)
18:07:45.719337 192.168.200.200 > 256.256.201.44: (frag 4321:1480@28120+)
18:07:45.720591 192.168.200.200 > 256.256.201.44: (frag 4321:1480@29600+)
18:07:45.721845 192.168.200.200 > 256.256.201.44: (frag 4321:1480@31080+)
18:07:45.723147 192.168.200.200 > 256.256.201.44: (frag 4321:1480@32560+)
18:07:45.724402 192.168.200.200 > 256.256.201.44: (frag 4321:1480@34040+)
18:07:45.725633 192.168.200.200 > 256.256.201.44: (frag 4321:1480@35520+)
18:07:45.726864 192.168.200.200 > 256.256.201.44: (frag 4321:1480@37000+)
18:07:45.728119 192.168.200.200 > 256.256.201.44: (frag 4321:1480@38480+)
18:07:45.729349 192.168.200.200 > 256.256.201.44: (frag 4321:1480@39960+)
18:07:45.730581 192.168.200.200 > 256.256.201.44: (frag 4321:1480@41440+)
18:07:45.731810 192.168.200.200 > 256.256.201.44: (frag 4321:1480@42920+)
18:07:45.733041 192.168.200.200 > 256.256.201.44: (frag 4321:1480@44400+)
18:07:45.734296 192.168.200.200 > 256.256.201.44: (frag 4321:1480@45880+)
18:07:45.735526 192.168.200.200 > 256.256.201.44: (frag 4321:1480@47360+)
```

18:07:45.736757 192.168.200.200 > 256.256.201.44: (frag 4321:1480@48840+)
18:07:45.737987 192.168.200.200 > 256.256.201.44: (frag 4321:1480@50320+)
18:07:45.739243 192.168.200.200 > 256.256.201.44: (frag 4321:1480@51800+)
18:07:45.740475 192.168.200.200 > 256.256.201.44: (frag 4321:1480@53280+)
18:07:45.741704 192.168.200.200 > 256.256.201.44: (frag 4321:1480@54760+)
18:07:45.742934 192.168.200.200 > 256.256.201.44: (frag 4321:1480@56240+)
18:07:45.744165 192.168.200.200 > 256.256.201.44: (frag 4321:1480@57720+)
18:07:45.745395 192.168.200.200 > 256.256.201.44: (frag 4321:1480@59200+)
18:07:45.746626 192.168.200.200 > 256.256.201.44: (frag 4321:1480@60680+)
18:07:45.747856 192.168.200.200 > 256.256.201.44: (frag 4321:1480@62160+)
18:07:45.749087 192.168.200.200 > 256.256.201.44: (frag 4321:1480@63640+)
18:07:45.750324 192.168.200.200 > 256.256.201.44: (frag 4321:1480@65120)
18:08:15.750086 256.256.201.44 > 192.168.200.200: icmp: ip reassembly time exceeded [tos 0xc0]

1. Source of trace

   Our internal network.

2. Detect was generated by:

   Tcpdump with ICMP filters.
   Log Format: Time of packet – Source IP – Dest IP – Fragment ID – Frag Length – Frag Offset

3. Probability the source address was spoofed

   Since the ping/packet o' death attack is based on a fragmented packet reassembly exceeding the maximum packet size, a response is not expected. For that reason, the address was probably spoofed to avoid being found. A ping (with no response) verified this was probably true, since someone without the foresight to spoof their IP address probably also wouldn't have changed their machine not to respond to ping.

4. Description of attack:

   Ping O' Death. The specific tool in this case is one of the publicly available C programs like evilping.c or pinger.c, written specifically to create the packets. The attack was through ICMP echo requests (normally created by 'ping').

5. Attack mechanism:

   Attack against unpatched TCP/IP stacks that, when receiving an engineered packet that, when fully reassembled, is larger than 65535, causes the stack to crash or behave erratically. This could cause a denial of service, at least of a person's workstation, or a larger scale one if the machine was a file server or the like.

6. Correlations:

   CVE-1999-0128 - Oversized ICMP ping packets can result in a denial of service, aka Ping o' Death.

7. Evidence of active targeting:

   The ping o' death, being a denial of service attack, is launched against specific hosts, showing that active targeting was present.

8. Severity:

(Critical 2 + Lethal 4) – (System 5 + Net Countermeasures 0) = 1

9. Defensive recommendation:

The machine survived! This box did have the proper patches to handle this attack properly. Recommend monitoring this IP for other occurrences, to ensure this is not a regular habit. This attack also reminds us all to make sure all of our OSes are properly patched.

10. The trace above is indicative of what type of attack:
a) Land Attack
b) Back Orifice
c) Ping o' Death
d) SYN Flood

Answer: c

Detect 4

Snort Alert:
Jun 1 12:53:21 NWLIN01 snort[11295]: MISC-Attempted Sun RPC high port access
: 256.256.48.20:60061 -> 256.256.48.105:32771
Jun 1 12:53:27 NWLIN01 snort[11295]: spp_portscan: portscan status from 256.256.48.20: 1476 connections across 1 hosts: TCP(1476), UDP(0) STEALTH

Snort Portscan Log:
Jun 1 12:53:26 256.256.48.20:60061 -> 256.256.48.105:599 XMAS ***F*P*U
Jun 1 12:53:26 256.256.48.20:60061 -> 256.256.48.105:6000 XMAS ***F*P*U
Jun 1 12:53:26 256.256.48.20:60061 -> 256.256.48.105:587 XMAS ***F*P*U
Jun 1 12:53:26 256.256.48.20:60061 -> 256.256.48.105:162 XMAS ***F*P*U
Jun 1 12:53:26 256.256.48.20:60061 -> 256.256.48.105:1531 XMAS ***F*P*U
Jun 1 12:53:26 256.256.48.20:60061 -> 256.256.48.105:9100 XMAS ***F*P*U
Jun 1 12:53:26 256.256.48.20:60061 -> 256.256.48.105:930 XMAS ***F*P*U
Jun 1 12:53:26 256.256.48.20:60061 -> 256.256.48.105:724 XMAS ***F*P*U
Jun 1 12:53:26 256.256.48.20:60061 -> 256.256.48.105:961 XMAS ***F*P*U
Jun 1 12:53:26 256.256.48.20:60061 -> 256.256.48.105:1369 XMAS ***F*P*U
Jun 1 12:53:26 256.256.48.20:60061 -> 256.256.48.105:666 XMAS ***F*P*U
Jun 1 12:53:26 256.256.48.20:60061 -> 256.256.48.105:765 XMAS ***F*P*U

Snort Output:
06/01-12:53:21.856475 256.256.48.20:60061 -> 256.256.48.105:1397
TCP TTL:46 TOS:0x0 ID:2758 ***F*P*U Seq: 0x0 Ack: 0x0 Win: 0x1000

06/01-12:53:21.856545 256.256.48.105:1397 -> 256.256.48.20:60061
TCP TTL:255 TOS:0x0 ID:64962
****R*A* Seq: 0x0 Ack: 0x0 Win: 0x0

06/01-12:53:21.856755 256.256.48.20:60061 -> 256.256.48.105:966
TCP TTL:46 TOS:0x0 ID:56551
***F*P*U Seq: 0x0 Ack: 0x0 Win: 0x1000

06/01-12:53:21.856811 256.256.48.105:966 -> 256.256.48.20:60061
TCP TTL:255 TOS:0x0 ID:64963
****R*A* Seq: 0x0 Ack: 0x0 Win: 0x0

06/01-12:53:21.857053 256.256.48.20:60061 -> 256.256.48.105:2025
TCP TTL:46 TOS:0x0 ID:62381

***F*P*U Seq: 0x0 Ack: 0x0 Win: 0x1000

06/01-12:53:21.857108 256.256.48.105:2025 -> 256.256.48.20:60061
TCP TTL:255 TOS:0x0 ID:64964
****R*A* Seq: 0x0 Ack: 0x0 Win: 0x0

1. Source of trace

   Our network.

2. Detect was generated by:

   Snort intrusion detection system.

3. Probability the source address was spoofed

   Low - Since only one IP address made the scan, decoys do not apply. If the mapper hopes to receive the results of their scan, the source address will not be spoofed.

4. Description of attack:

   "Xmas Tree" scan. The Xmas term came from the lit-up effect of the flags that are set in the packet. The impossible flag combination (Fin, Push, Urg) alerted Snort to notice this. Specific tool is nmap or similar with a stealth xmas tree function.

5. Attack mechanism:

   The xmas tree scan probes a selected range of ports with packets to solicit responses for mapping the open ports on a box.

   The main purpose of such a scan is reconnaissance. It can provide a list of open services on the box for further exploitation later.

6. Correlations:

   Port scans are commonly reported on GIAC. Examples include:

   http://www.sans.org/y2k/052800-1100.htm
   and
   http://www.sans.org/y2k/052300-0800.htm

7. Evidence of active targeting:

   Other hosts were not scanned in this attack, so it appears to be a targeted attack.

8. Severity:

   (Critical 2 + Lethal 2) – (System 5 + Net Countermeasures 0) = -1

9. Defensive recommendation:

   I ran an nmap on this machine and found it to only be running essential services.. including ssh instead of telnet. Recommend putting IP address on a "watch" list for further activity.

10. The syntax for an nmap stealth Xmas tree scan is:

a) nmap –O hostname
b) nmap –n hostname
c) ifconfig -a
d) nmap –sX hostname

Answer: d

Detect 5

Snort Alert:

Jun 6 08:22:05 NWLIN01 snort[11295]: IDS005 - SCAN-Possible NMAP Fingerprint attempt: 256.256.41.3:41479 -> 256.256.48.105:21

Snort Output:

06/06-08:22:04.919560 256.256.41.3:41479 -> 256.256.48.105:21
TCP TTL:45 TOS:0x0 ID:8464
**SF*P*U Seq: 0xA5022E3 Ack: 0x0 Win: 0xC00
TCP Options => WS: 10 NOP MSS: 265 TS: 1061109567 0 EOL EOL

06/06-08:22:04.919603 256.256.48.105:21 -> 256.256.41.3:41479
TCP TTL:64 TOS:0x0 ID:60702 DF
S*A* Seq: 0x36640D57 Ack: 0xA5022E4 Win: 0x7F53
TCP Options => MSS: 265 NOP NOP TS: 39632688 1061109567 NOP WS: 0

06/06-08:22:04.919740 256.256.41.3:41480 -> 256.256.48.105:21
TCP TTL:45 TOS:0x0 ID:10403
******A* Seq: 0xA5022E3 Ack: 0x0 Win: 0xC00
TCP Options => WS: 10 NOP MSS: 265 TS: 1061109567 0 EOL EOL

06/06-08:22:04.919793 256.256.48.105:21 -> 256.256.41.3:41480
TCP TTL:255 TOS:0x0 ID:60703
****R*** Seq: 0x0 Ack: 0x0 Win: 0x0

1. Source of trace

   Our network

2. Detect was generated by:

   Snort IDS System.

3. Probability the source address was spoofed

   Not spoofed: An OS fingerprint will not work if the source address is spoofed.

4. Description of attack:

   An attempt to identify the operating system of the target machine by sending a series of packets that, when responded to in certain ways, will give away the operating system of the target based on its behavior.

5. Attack mechanism:

   QueSO or similar operating system identification tool The strange and impossible packet shown above

with the SYN, FIN, PUSH, and URG flags all set are what got the attention of Snort's IDS. And, the fact that the machine responded with a SYN-ACK is probably indicative that the target machine willingly gave up a piece of the puzzle, that when put together, will reveal the OS of the machine for further targeting.

6. Correlations:

CVE CAN-1999-0454 - ** CANDIDATE (under review) ** A remote attacker can sometimes identify the operating system of a host based on how it reacts to some IP or ICMP packets, using a tool such as nmap or queso.

7. Evidence of active targeting:

Active targeting present, as OS fingerprinting looks to a particular host to determine its running OS.

8. Severity:

(Critical 2 + Lethal 2) – (System 5 + Net Countermeasures 0) = -1

9. Defensive recommendation:

Recommend watching this IP for further activity (just in case). The defenses are fine, the IDS picked the attack up correctly. The OS was probably determined, so recommend ensuring latest security patches.

10. An example of an OS fingerprinting tool is:
a) QueSO
b) tcpdump
c) uname
d) Trin00

Answer: a

Detect 6

Snort Alert Log:

Jun 8 10:32:49 NWLIN01 snort[11295]: PING-ICMP Destination Unreachable: 256.256.29.23 -> 256.256.147.14
Jun 8 10:32:49 NWLIN01 snort[11295]: PING-ICMP Destination Unreachable: 256.256.29.15 -> 256.256.147.14
Jun 8 10:32:49 NWLIN01 snort[11295]: PING-ICMP Destination Unreachable: 256.256.29.1 -> 256.256.147.14
Jun 8 10:32:49 NWLIN01 snort[11295]: PING-ICMP Destination Unreachable: 256.256.29.24 -> 256.256.147.14
Jun 8 10:32:49 NWLIN01 snort[11295]: PING-ICMP Destination Unreachable: 256.256.15.85 -> 256.256.147.14
Jun 8 13:21:14 NWLIN01 snort[11295]: PING-ICMP Destination Unreachable: 256.256.29.23 -> 256.256.147.14
Jun 8 13:21:14 NWLIN01 snort[11295]: PING-ICMP Destination Unreachable: 256.256.29.1 -> 256.256.147.14
Jun 8 13:21:14 NWLIN01 snort[11295]: PING-ICMP Destination Unreachable: 256.256.15.85 -> 256.256.147.14
Jun 8 13:21:14 NWLIN01 snort[11295]: PING-ICMP Destination Unreachable: 256.256.29.24 -> 256.256.147.14

Jun 8 13:21:14 NWLIN01 snort[11295]: PING-ICMP Destination Unreachable: 256.256.29.15 -> 256.256.147.14
Jun 8 13:21:14 NWLIN01 snort[11295]: PING-ICMP Destination Unreachable: 256.256.15.86 -> 256.256.147.14

Snort Output Snipets:

06/08-10:32:49.156913 256.256.147.94:138 -> 256.256.147.255:138
UDP TTL:64 TOS:0x0 ID:19244
Len: 215

06/08-10:32:49.158372 256.256.29.23 -> 256.256.147.14
ICMP TTL:253 TOS:0x0 ID:35105 DF
DESTINATION UNREACHABLE: PORT UNREACHABLE

06/08-10:32:49.158583 256.256.29.15 -> 256.256.147.14
ICMP TTL:253 TOS:0x0 ID:55569 DF
DESTINATION UNREACHABLE: PORT UNREACHABLE

06/08-10:32:49.158698 256.256.29.1 -> 256.256.147.14
ICMP TTL:253 TOS:0x0 ID:40549 DF
DESTINATION UNREACHABLE: PORT UNREACHABLE

06/08-10:32:49.158818 256.256.29.24 -> 256.256.147.14
ICMP TTL:253 TOS:0x0 ID:25421 DF
DESTINATION UNREACHABLE: PORT UNREACHABLE

06/08-13:21:14.087947 256.256.147.14:138 -> 256.256.147.255:138
UDP TTL:128 TOS:0x0 ID:13248
Len: 224

06/08-13:21:14.091262 256.256.29.23 -> 256.256.147.14
ICMP TTL:253 TOS:0x0 ID:48041 DF
DESTINATION UNREACHABLE: PORT UNREACHABLE

06/08-13:21:14.091377 256.256.29.1 -> 256.256.147.14
ICMP TTL:253 TOS:0x0 ID:53365 DF
DESTINATION UNREACHABLE: PORT UNREACHABLE

1. Source of trace

   Our network.

2. Detect was generated by:

   Snort IDS.

3. Probability the source address was spoofed

   The source address was not spoofed, as we determined what the machine was doing below:

4. Description of attack:

This originally appeared to Snort (and me) to be a broadcast network-mapping type attack. A packet Sent to broadcast addresses can be used to map a network based on the responses to that broadcast.

5. Attack mechanism:

A network mapping attack would send request to broadcast addresses. ICMP unreachable messages would allow for a reverse mapping situation of machines that are NOT available. This information could be used to narrow down a list of machines to map and attack.

However, the Netbios port 138 in these packets was what got my attention. Since Netbios is so "chatty" I decided to investigate the IP address itself further.

Here is what I received from nmap OS Fingerprinting:

```
# nmap -O 256.256.147.14

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on nationwi-slhyzc.cdc.ent.nwie.net (256.256.147.14):
(The 1515 ports scanned but not shown below are in state: closed)
Port       State       Service
25/tcp     open        smtp
80/tcp     open        http
135/tcp    open        loc-srv
139/tcp    open        netbios-ssn
443/tcp    open        https
445/tcp    open        microsoft-ds
1030/tcp   open        iad1
1031/tcp   open        iad2

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=12623 (Worthy challenge)
Remote operating system guess: Windows 2000 RC1 through final release

Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
```

We traced the box to its owner by nbtstat, and determined that it was a test Windows 2000 server, soliciting netbios information.

6. Correlations:

CVE-1999-0513 ICMP messages to broadcast addresses are allowed, allowing for a Smurf attack that can cause a denial of service.

7. Evidence of active targeting:

No evidence of active targeting.. these responses were responses to broadcast addresses.

8. Severity:

(Critical 2 + Lethal 0) – (System 4 + Net Countermeasures 0) = -2

9. Defensive recommendation:

Defenses are okay. Recommend keeping in mind the chattiness (and now extended chattiness with Windows 2000) of NT Operating systems.

10. Ports 137-139 are commonly identified as:

a) NetBIOS
b) DNS
c) Echo
d) Chargen

Answer: a

Detect 7

Bind logs:

Jun 8 06:06:01 ns1 named[9997]: unapproved query from [207.46.106.84].7796 for "VERSION.BIND"
Jun 8 06:09:44 ns1 named[9997]: unapproved query from [207.46.106.84].11831 for "VERSION.BIND"
Jun 8 06:10:38 ns1 named[9997]: unapproved query from [207.46.106.84].12549 for "VERSION.BIND"
Jun 8 07:19:04 ns1 named[9997]: unapproved query from [207.46.106.84].9776 for "VERSION.BIND"
Jun 8 07:19:54 ns1 named[9997]: unapproved query from [207.46.106.84].10874 for "VERSION.BIND"
Jun 8 07:22:36 ns1 named[9997]: unapproved query from [207.46.106.84].13521 for "VERSION.BIND"

http://www.sans.org/y2k/053100-1200.htm

1. Source of trace

Our external name server.

2. Detect was generated by:

bind. This is just a sample of the patterns in our /var/adm/messages file on our name servers.
Log file format: Date – Time – Message

3. Probability the source address was spoofed

The source address probably was not spoofed, or a succesful inquiry of the bind version would not make it back to the attacker.

4. Description of attack:

Repeated inquiries to name servers for bind versions. With many attacks against DNS bind known, this is commonly sought after information.

5. Attack mechanism:

Attacks against name servers query the DNS port 53 for information, that when misconfigured, bind is very willing to give up. In addition to the above, we receive disallowed zone transfers all the time, which is a common favorite hacker start point for network mapping.

6. Correlations:

Just a sample of the CVE listings on bind:

CVE-1999-0009 - Inverse query buffer overflow in BIND 4.9 and BIND 8 Releases.
CVE-1999-0024 - DNS cache poisoning via BIND, by predictable query IDs.
CVE-1999-0849 - Denial of service in BIND named via maxdname.

I have also seen this reported multiple times on GIAC:
http://www.sans.org/y2k/053000.htm
http://www.sans.org/y2k/052600.htm
and more…

I've done some digging (OK, a lot of digging) and have even found some of this activity coming FROM our own name servers! The IP listed above resolves (by whois) to Microsoft (msft.net).

Netname: MICROSOFT-GLOBAL-NET
Netblock: 207.46.0.0 - 207.46.255.255

Further, an nslookup returns:
Name: sjwu3dns1.windowsupdate.com
Address: 207.46.106.84

By name alone.. a DNS server. Is there some misconfiguration going on that causes DNS servers to query each other this way?? Thoughts?

7. Evidence of active targeting:

Half yes, half no. A hacker could specifically be targeting our name servers, or could just be doing a sweep on a whole range of IP's.

8. Severity:

(Critical 5 + Lethal 5) – (System 5 + Net Countermeasures 4) = 1

9. Defensive recommendation:

Defenses are okay. The nameservers are configured to only allow the inquiries and zone transfers from specific hosts. Recommend building database of these IP's (as this occurs so often) to find repeat offenders.

10. DNS is:

a) Served on Port 53
b) Essential to the Internet
c) A popular attack and network reconnaissance target.
d) All of the above

Answer: d

Detect 8

Jun 8 22:50:18 NWLIN01 snort[11295]: PING-ICMP Time Exceeded: 256.256.197.239 -> 256.256.48.10
Jun 8 22:50:18 NWLIN01 snort[11295]: PING-ICMP Time Exceeded: 256.256.197.239 -> 256.256.48.10
Jun 8 22:50:18 NWLIN01 snort[11295]: PING-ICMP Time Exceeded: 256.256.197.239 -> 256.256.48.10

06/08-22:50:18.272950 256.256.48.10:44557 -> 256.255.3.158:33436
UDP TTL:1 TOS:0x0 ID:44559
Len: 18

06/08-22:50:18.273781 256.256.197.239 -> 256.256.48.10
ICMP TTL:255 TOS:0xC0 ID:50907
TTL EXCEEDED

06/08-22:50:18.276491 256.256.48.10:44557 -> 256.255.3.158:33437
UDP TTL:1 TOS:0x0 ID:44560
Len: 18

06/08-22:50:18.277319 256.256.197.239 -> 256.256.48.10
ICMP TTL:255 TOS:0xC0 ID:50908
TTL EXCEEDED

06/08-22:50:18.278988 256.256.48.10:44557 -> 256.255.3.158:33438
UDP TTL:2 TOS:0x0 ID:44561
Len: 18

06/08-22:50:18.280144 256.256.238.4 -> 256.256.48.10
ICMP TTL:254 TOS:0x0 ID:33892
TTL EXCEEDED

1. Source of trace

   Our network.

2. Detect was generated by:

   Snort IDS.

3. Probability the source address was spoofed

   Probably not spoofed. A traceroute is a diagnostic tool used to determine network paths and response time.

4. Description of attack:

   Probably not an attack. The TTL exceeded messages and incrementing TTL values are indicative of the traceroute tool. Since it is UDP and not ICMP, we can guess it is a Unix and not Windows based traceroute. The high port numbers (33000-34999) are also indicative of traceroute.

5. Attack mechanism:

   See above. The traceroute tool sends multiple packets with incrementing TTL values, allowing for the calculation of time in transit between hops.

6. Correlations:

   Network mapping by traceroute is discussed day two of the SANS Intrusion Detection text, Intrusion Detection and Packet Filtering: How it Really Works. Page 94 begins the discussion of traceroute. Used in different geographic locations and at different times of the day, a would be hacker could gain valuable information about a site's network topology.

7. Evidence of active targeting:

   Active targeting evident, as traceroute reveals data about a specific machine, and the route to reach it.

8. Severity:

   (Critical 2 + Lethal 0) – (System 5 + Net Countermeasures 0) = -3

9. Defensive recommendation:

Defenses not needed until a larger pattern of traceroute has been displayed.

10. Traceroute works by:

a) Sending packets with incrementing time-to-live values to time and reveal router "hops" in a network path.
b) Querying a target for its network path
c) Sending impossible flag combinations to a target and evaluating the responses
d) Probing high numbered ports to avoid detection

Answer: a

Detect 9

May 23 23:12:51 207.78.247.50:65535 -> xxx.yyy.zzz.137:8080 SYN **S*****
May 23 23:12:51 207.78.247.50:65535 -> xxx.yyy.zzz.138:8080 SYN **S*****
May 23 23:12:51 207.78.247.50:65535 -> xxx.yyy.zzz.139:8080 SYN **S*****
May 23 23:12:51 207.78.247.50:65535 -> xxx.yyy.zzz.140:8080 SYN **S*****
May 23 23:12:52 207.78.247.50:65535 -> xxx.yyy.zzz.170:8080 SYN **S*****

1. Source of trace

http://www.sans.org/y2k/052700-2100.htm

2. Detect was generated by:

Snort IDS.

3. Probability the source address was spoofed

Low probability the source address was spoofed. A correct address is needed to get the reponses back.

4. Description of attack:

Host scan for port 8080. Attacker could be searching for web proxy servers such as Wingate.

5. Attack mechanism:

SYN packets are sent to port 8080 of many hosts. Those that respond can be logged as candidates for running as a proxy server. An unsecured proxy server can be used by a hacker to achieve anonymity in their exploitation of another target.

Also, vulnerabilities exist in some versions of Wingate that can be exploited, as noted in CVE below:

6. Correlations:

CVE-1999-0291-The WinGate proxy is installed without a password, which allows remote attackers to redirect connections without authentication.
CVE-1999-0290 - The WinGate telnet proxy allows remote attackers to cause a denial of service via a large number of connections to localhost.
CVE-1999-0494 - Denial of service in WinGate proxy through a buffer overflow in POP3.

7. Evidence of active targeting:

Many addresses being scanned.. no active targeting evident.

8. Severity:

(Critical 2 + Lethal 1) – (System 5 + Net Countermeasures 4) = -6
This is not my detect. I am assuming our own workstation systems with up-to-date patches being scanned since no responses are displayed.

9. Defensive recommendation:

Defenses not really necessary unless a Wingate proxy is indeed running. Recommend putting IP on watch list.

10. The above is an example of a scan for:

a) Back Orifice.
b) Loki.
c) Wingate Proxy Server.
d) Probing high numbered ports to avoid detection

Answer: c

Detect 10

Jun 7 05:35:22 fwall 12 deny: TCP from 24.17.96.120.1788
to mysubnet.0.111 seq 8BABB3F2, ack 0x0, win 32120, SYN
Jun 7 05:35:22 fwall 15 deny: TCP from 24.17.96.120.1789
to mysubnet.1.111 seq 8BA6EB58, ack 0x0, win 32120, SYN
Jun 7 05:35:23 fwall 15 deny: TCP from 24.17.96.120.1790
to mysubnet.2.111 seq 8BC4971F, ack 0x0, win 32120, SYN
Jun 7 05:35:23 fwall 15 deny: TCP from 24.17.96.120.1791
to mysubnet.3.111 seq 8B9ED1AC, ack 0x0, win 32120, SYN
Jun 7 05:35:23 fwall 15 deny: TCP from 24.17.96.120.1792
to mysubnet.4.111 seq 8C290568, ack 0x0, win 32120, SYN

1. Source of trace

http://www.sans.org/y2k/060900-1030.htm

2. Detect was generated by:

Firewall logs.

3. Probability the source address was spoofed

Likely not spoofed.. response is required for effective reconnaissance.

4. Description of attack:

Probe for TCP port 111 sunrpc. RPC has had a history of exploitable bugs, making scans to find the existence of the port popular.

5. Attack mechanism:

A portscan on TCP port 111 sends a packet to the port of a range of addresses. Those ports that respond will indicate to the attacker the machines with sunrpc running. Once found, machines without proper patching may be vulnerable to exploit by scripts and tools.

6. Correlations:

Portmap exploit discussed at: http://packetstorm.securify.com/Exploit_Code_Archive/portmap.txt
Another example: CVE-1999-0212 Solaris rpc.mountd generates error messages that allow a remote attacker to determine what files are on the server.

7. Evidence of active targeting:

No active targeting.. scanning multiple hosts for an open port.

8. Severity:

(Critical 4 + Lethal 5) – (System 5 + Net Countermeasures 4) = 0
This is not my detect. I am assuming our own sever systems with up-to-date patches being scanned. These were also blocked at the firewall.

9. Defensive recommendation:

Recommend double-checking the patch revisions on systems and monitoring for future activity.

10. SunRPC is found at:

a) TCP Port 53
b) TCP Port 111
c) UDP Port 53
d) TCP Port 31337

Answer: b

Last Updated: September 19th, 2017

Upcoming Training

SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
Baltimore Fall 2017 - SEC503: Intrusion Detection In-Depth	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
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
San Diego Fall 2017 - SEC503: Intrusion Detection In-Depth	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	vLive
SANS San Diego 2017	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Seattle 2017	Seattle, WA	Oct 30, 2017 - Nov 04, 2017	Live Event
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 London February 2018	London, United Kingdom	Feb 05, 2018 - Feb 10, 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 Secure Canberra 2018	Canberra, Australia	Mar 19, 2018 - Mar 24, 2018	Live Event
SANS 2018	Orlando, FL	Apr 03, 2018 - Apr 10, 2018	Live Event
SANS Baltimore Spring 2018	Baltimore, MD	Apr 21, 2018 - Apr 28, 2018	Live Event
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced