Global Information Assurance Certification Paper

GIAC Intrusion Detection Curriculum
Practical Assignment for SNAP San Jose
May 8 – 13, 2000

10 Detects with Analyses

David Blaine
6/6/2000

Detect 1

```
May 18 21:57:48 mymachine kernel: Packet log: bad-if DENY ppp0
  PROTO=17 209.0.241.136:1114 203.27.xxx.yyy:61072
  L=282 S=0x00 I=23024 F=0x0000 T=109
May 18 21:57:48 mymachine kernel: Packet log: bad-if DENY ppp0
  PROTO=17 209.0.241.136:1114 203.27.xxx.yyy:61072
  L=272 S=0x00 I=31472 F=0x0000 T=109
May 18 21:57:48 mymachine kernel: Packet log: bad-if DENY ppp0
  PROTO=17 209.0.241.136:1114 203.27.xxx.yyy:61072
  L=272 S=0x00 I=42224 F=0x0000 T=109
```

1. Source of trace:

This trace was posted on GIAC by an analyst at Taurfish Technology Services in Sydney, Australia (URL: http://www.sans.org/y2k/052900.htm).

2. Detect was generated by:

This detect was reported by an IPchains firewall.

3. Probability the source address was spoofed:

With UDP-based attacks spoofing the source address is common, however, in this instance, I believe it hasn't been spoofed (see below).

4. Description of attack:

UDP port 61072 scan. At www.nic.com, the source address is registered to Level 3 Communications, LLC in Colorado.

5. Attack mechanism:

This attack works by "knocking on the door" of UDP port 61072 on several machines. If a service was listening on the port, then the attack could be begin. In this case, no application was listening on this port. After some investigation, I could not find any malicious code that would.

6. Correlation:

From my search of Internet sources, I was unable to find a attack signature using UDP port 61072.

7. Evidence of active targeting:

Since Level 3 Communications, LLC "is a communications and information services company that is building an international advanced Internet Protocol (IP) technology based network," I believe they might

be testing new equipment or software as part of their business. This detect shows no evidence of any active targeting; It is simply a "wrong number." Finding no other detect with this type of signature also seems to confirm this conclusion.

8. Severity:

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

Criticality = 5. According to the analyst, this network only has one registered IP address so it is assumed this IP address would be assigned to a key Internet infrastructure machine.

Lethality = 1. Because I could not find a trojan or otherwise malicious code that listens to UDP port 61072, it is unlikely that this would be lethal.

System Countermeasures = 5. In actuality, this value is unknown because the analyst does not state how his system security is maintained. Since he appears to be a security professional, I gave him the benefit of the doubt and assumed he maintained his patch levels and added additional host-based security.

Network Countermeasures = 4. According to the analyst, there are several points of entry although each are protected by a Ipchains firewall.

Severity = (5 + 1) – (5 + 4) = 6 – 9 = -3. This severity value indicates that we have sufficient countermeasures to protect against this attack.

9. Defensive recommendation:

Defenses are currently successful in blocking this attack (as indicated by the DENY in the detect).

10. Multiple choice test question based on the above detect:

a) Trojan probe
b) Scan that turns knobs
c) Wrong number
d) Smurf attack

Answer: c)

GIAC Intrusion Detection Curriculum
Practical Assignment for SNAP San Jose
May 8 – 13, 2000

10 Detects with Analyses

David Blaine
6/6/2000

Detect 2

```
21:15:31.963877 di5.akto-di-1.online.kz.4156 > my.box.27374:
   S 6621802:6621802(0) win 8192 <mss 536,nop,nop,sackOK> [tos 0x1]
   (ttl 108, id 59078) 4501 0030 e6c6 0000 6c06 0ea2 d413 98c4
   ### #### 103c 6aee 0065 0a6a 0000 0000 7002 2000 8763 0000 0204
         0218 0101 0402
21:15:32.003272 my.box.27374 > di5.akto-di-1.online.kz.4156:
   R 0:0(0) ack 6621803 win 0 (ttl 128, id 56749)
21:15:32.942230 di5.akto-di-1.online.kz.4156 > my.box.27374:
   S 6621802:6621802(0) win 8192 <mss 536,nop,nop,sackOK> [tos 0x1]
   (ttl 108, id 8391) 4501 0030 20c7 0000 6c06 d4a1 d413 98c4
   ### #### 103c 6aee 0065 0a6a 0000 0000 7002 2000 8763 0000 0204 0218
   0101 0402
21:15:32.942516 my.box.27374 > di5.akto-di-1.online.kz.4156:
   R 0:0(0) ack 1 win 0 (ttl 128, id 56750)
21:15:33.847263 di5.akto-di-1.online.kz.4156 > my.box.27374:
   S 6621802:6621802(0) win 8192 <mss 536,nop,nop,sackOK> [tos 0x1]
   (ttl 108, id 11719) 4501 0030 2dc7 0000 6c06 c7a1 d413 98c4
   ### #### 103c 6aee 0065 0a6a 0000 0000 7002 2000 8763 0000 0204
   0218 0101 0402
```

1. Source of trace:

This trace was posted on GIAC by Stephan Odak (URL: http://www.sans.org/y2k/053100-1100.htm).

2. Detect was generated by:

This detect was reported by TCPdump.

3. Probability the source address was spoofed:

The source address is probably not spoofed because it is a TCP-based detect.

4. Description of attack:

Attack against TCP port 27374. This has a signature of a Sub-Seven Trojan v2.0 probe.

5. Attack mechanism:

The attack works by completing the three-way handshake if it finds a Sub-seven trojan previously installed on the victim's computer. In this case, the victim's computer, my.box, sends a RESET (indicated by the R 0:0(0) in the detect above) because no trojan is listening on this port. From information found at http://www.robertgraham.com/pubs/firewall-seen.html, Sub-seven:

a) Includes a scanner, but also can tell a slave machine to scan as well.
b) Supports "port redirection", so that any attack can be funneled through a victim's machines.

c) Contains extensive tricks to play with ICQ, AOL IM, MSN Messenger, and Yahoo messenger, including password sniffing, posting messages, and other features.
d) Contains extensive UI tricks, such as flipping the screen, talking through the victim's speaker, and spying on the victim's screen.

6. Correlation:

This probe is searching for a variant of the well-known Sub-Seven Trojan. The port used identifies the probe is searching for version 2.0 of the trojan. This is an extremely popular trojan.

7. Evidence of active targeting:

The attacker is specifically targeting "my.box".

8. Severity:

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

Criticality = 2. Although not explicitly stated, I am assuming that "my.box" is a end-user Windows PC and therefore not a critical system.

Lethality = 5. Sub-seven is an extremely powerful trojan. It is able to take control of a user's PC and perform a variety of User Interface (UI) pranks, as well as steal passwords.

System Countermeasures = 4. Although the system has a modern operating system (MS Windows), there is no information stating patch levels maintained or security systems added. Considering it is running a Microsoft product, there is probably a service pack due out any time.

Network Countermeasures = 2. Although "my.box" sent a reset to deny the Sub-seven probe, the firewall is permissive and allowed the attack to pass through the perimeter.

So, Severity = (2 + 5) – (4 + 2) = 7 – 6 = 1.

9. Defensive recommendation:

Block all high TCP ports (>1024) at the firewall.

10. Multiple choice test question based on above detect:

This probe is indicative of a Sub-Seven probe. To block this attack you would:

a) Block all high TCP ports at the firewall.
b) Block all outbound traffic to `di5.akto-di-1.online.kz`.
c) Block all high UDP ports at the firewall.
d) Send a SOS distress call.

GIAC Intrusion Detection Curriculum
Practical Assignment for SNAP San Jose
May 8 – 13, 2000

10 Detects with Analyses

David Blaine
6/6/2000

Answer: a)

GIAC Intrusion Detection Curriculum
Practical Assignment for SNAP San Jose
May 8 – 13, 2000

10 Detects with Analyses

David Blaine
6/6/2000

Detect 3

```
02:59:03.041606 cc967152-a.nwhub1.in.home.com.4990 > my.box.2222:
S 888122788:888122788(0) win 32120
<mss 1460,sackOK,timestamp 10490297 0,nop,wscale 0> (DF)
(ttl 56, id 50298) 4500 003c c47a 4000 3806 1ad5 180a 5edc xxxx xxxx
137e 08ae 34ef ada4 0000 0000 a002 7d78 5609 0000 0204 05b4 0402
080a 00a0 11b9 0000 0000 0103 0300
02:59:03.070983 my.box.2222 > cc967152-a.nwhub1.in.home.com.4990:
R 0:0(0) ack 888122789 win 0 (ttl 128, id 8043)
```

1. Source of trace:

This trace comes from GIAC at the following URL: http://www.sans.org/y2k/053100-1100.htm. It was posted by Stephan Odak.

2. Detect was generated by:

This detect was generated by TCPdump.

3. Probability the source address was spoofed:

Since this trace indicates TCP traffic, the source address is probably not spoofed.

4. Description of attack:

A single attack against TCP port 2222 with odd TCP options. This attack seems to be one in which the attacker is trying to determine how the listening application reacts to strange TCP options.

5. Attack mechanism:

If an application running on my.box was listening on TCP port 2222 then the three-way handshake would complete. This detect may be an attempt to check for a running proxy. Research on the Internet turned up Spoonproxy that runs on TCP port 2222 (refer to URL: http://www.pi-soft.com/spoonproxy.html for more information). If it isn't Spoonproxy, it certainly is an attempt by the attacker to get information on how the mystery application would react to odd TCP optons.

6. Correlation:

Although I could not find another match on GIAC, I did find a similar one at: http://www.lists.ic.ac.uk/hypermail-archive/p99-firewall/p99-firewall-Dec-1999/0007.html. This site has a firewall alert with similarities to the above signature. However, no conclusion can be made since not all of the fields are reported.

7. Evidence of active targeting:

Although the attack targeted a specific host, more data needs to be acquired on this kind of attack to really determine its impact.

8. Severity:

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

Criticality = 2. Again, the poster does not specify what machine my.box is. I will assume for the purposes of this calculation that is simply a user's PC or workstation.

Lethality = 1. The lethality is low because there isn't any well-documented attacks using this signature.

System Countermeasures = 5. No specific information is given on the system's security strengths. The only conclusion to be made here is the system my.box only responded with a reset to close the connection.

Network Countermeasures = 2. If I assume my.box is behind a firewall, then the firewall is permissive to allow a high TCP port scan inside my protected network.

Severity = (2 + 1) – (5 + 2) = 3 – 7 = -4. Although this severity value indicates sufficient countermeasures are in place, see the defensive recommendation below.

9. Defensive recommendation:

Block all high TCP ports at the firewall to prevent these scans from reaching my.box.

10. Multiple choice test question based on above detect:

This attack is attempting to:

a) Detect trojans.
b) Shake your hand.
c) Send data in a covert channel.
d) Find out how a listening application will react to strange TCP options.

Answer: d)

GIAC Intrusion Detection Curriculum
Practical Assignment for SNAP San Jose
May 8 – 13, 2000

10 Detects with Analyses

David Blaine
6/6/2000

Detect 4

```
[**] RPC Info Query [**] 05/29-17:58:53.527261 209.27.200.129:986 ->
nnn.n.nnn.130:111 TCP TTL:240 TOS:0x0 ID:28571 DF *****PA* Seq:
0xE95458DA
Ack: 0xC901040F Win: 0x2238
[**] RPC Info Query [**] 05/29-17:59:15.029450 209.27.200.129:648 ->
nnn.n.nnn.172:111 TCP TTL:240 TOS:0x0 ID:50061 DF *****PA* Seq:
0xE9D58B5F
Ack: 0x47A7B659 Win: 0x2238
[**] RPC Info Query [**] 05/29-17:59:43.022267 209.27.200.129:761 ->
nnn.n.nnn.229:111 TCP TTL:240 TOS:0x0 ID:12515 DF *****PA* Seq:
0xEA7C9968
Ack: 0x1EF74F3F Win: 0x2238
```

1. Source of trace:

Posted to GIAC by Steve Richards at http://www.sans.org/y2k/053100-1100.htm.

2. Detect was generated by:

Snort intrusion detection system.

3. Probability the source address was spoofed:

The source address is probably not spoofed due to this detect being based on TCP.

4. Description of attack:

This attack is against TCP port 111 (RPC). This is a reconnaissance scan to determine what RPC services are running so the next phase of the attack can be planned and executed.

5. Attack mechanism:

This attack works by sending a "rpcinfo –p" command to determine what services the portmapper daemon will respond to. The attacker can then use this information to craft his next attack.

6. Correlation:

RPC Info queries are a popular reconnaissance technique. Roughly a half an hour later, Steve recorded a query on another network from another hostile host. I don't believe in coincidences either, Steve.

```
[**] RPC Info Query [**] 05/29-23:50:10.003490 63.199.194.66:684 ->
My.net.work.nn:111 TCP TTL:40 TOS:0x0 ID:61950 DF *****PA* Seq:
0x20C4E9F2
```

```
Ack: 0xE81AF82 Win: 0x4470
```

7. Evidence of active targeting:

The attack is a general scan of several hosts on a subnet within the network in random order.

8. Severity:

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

Criticality = 3. The attacker is not targeting core infrastructure machines. He is simply scanning the network for any machine to respond to his RPC query. This will be his springboard for his next attack.

Lethality = 5. If the subnet being scanned contains UNIX boxes running a vulnerable portmapper or rpcbind process, then the attacker could gain root access across the network.

System Countermeasures = 5. No machine scanned responded to the attacker's RPC request.

Network Countermeasures = 5. The IDS did not record any responses to the RPC query. The packets were probably perturbed by the firewall when it blocked the attack.

Severity = (3 + 5) – (5 + 5) = 8 – 10 = -2. This indicates we have sufficiently repelled the attack.

9. Defensive recommendation:

The attack was effectively blocked by the firewall.

10. Multiple choice test question based on above detect:

This attack will cause:

a) Root compromise
b) Another more serious attack will be coming in the future
c) OS fingerprinting
d) Gas

Answer: b)

GIAC Intrusion Detection Curriculum
Practical Assignment for SNAP San Jose
May 8 – 13, 2000

10 Detects with Analyses

David Blaine
6/6/2000

Detect 5

```
05/23/2000 06:59:53.176 TCP connection dropped
  Source:21.118.8.50, WAN Dest.2, 109,
05/23/2000 07:21:33.272 TCP connection dropped
  Source:21.118.8.50, WAN Dest.3, 109,
05/23/2000 07:43:14.272 TCP connection dropped
  Source:21.118.8.50, WAN Dest.4, 109,
05/23/2000 08:04:54.592 TCP connection dropped
  Source:21.118.8.50, WAN Dest.5, 109,
05/23/2000 08:48:16.176 TCP connection dropped
  Source:21.118.8.50, WAN Dest.7, 109,
05/23/2000 09:31:37.400 TCP connection dropped
  Source:21.118.8.50, WAN Dest.9, 109,
05/23/2000 09:53:17.576 TCP connection dropped
  Source:21.118.8.50, WAN Dest.10, 109,
05/23/2000 10:36:40.304 TCP connection dropped
  Source:21.118.8.50, WAN
```

1. Source of trace:

From GIAC at http://www.sans.org/y2k/052700.htm posted by lloyd@nwra.com.

2. Detect was generated by:

Log from unknown firewall.

3. Probability the source address was spoofed:

The source address is genuine and probably not spoofed because this transmission is TCP.

4. Description of attack:

This is a slow scan through the network in sequential order to TCP port 109 (POP2). POP2 has known buffer overflow problems which can lead to immediate root compromise. This attack is simply to find TCP port 109 open somewhere so it can be exploited later.

5. Attack mechanism:

The scan is slow to avoid the protected network's radar (Note the rate of scan of only 2 tries per hour). It aims at finding a host on the network that will respond with a three-way handshake on TCP port 109 (POP2). If this host was found, then a more focused attack could be staged to expose the vulnerabilities of the POP2 server at a later date. In this case, the attacker did not find any.

6. Correlation:

David Blaine
6/6/2000

POP2 servers are very popular targets for hackers, so it is extremely common to see TCP port 109 scans.

7. Evidence of active targeting:

This is a general scan of the network.

8. Severity:

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

Criticality = 3. A low criticality because this is a scan of the network and not targeting important machines.

Lethality = 5. If the attacker found a machine serving POP2, he could have gained root access across the network.

System Countermeasures = 5.

Network Countermeasures = 5. The firewall indicated it has turned away the attack by dropping the connections.

So, Severity = (3 + 5) – (5 + 5) = 8 – 10 = -2. It appears we have adequately protected against this attack.

9. Defensive recommendation:

From the severity score above, there are adequate countermeasures in place to deter this attack.

10. Multiple choice test question based on above detect:

a) POP2 reconnaissance
b) OS fingerprinting
c) Chatty firewall
d) UDP port scan

Answer: a)

10 Detects with Analyses

David Blaine
6/6/2000

Detect 6

```
Jun 4 10:58:12 solar portsentry[721]: attackalert:
  SYN/Normal scan from host: cr664432-
a.yec1.on.wave.home.com/24.42.134.19
  to TCP port: 80
Jun 4 10:58:12 solar portsentry[721]: attackalert: Host 24.42.134.19
  has been blocked via wrappers with string: "ALL: 24.42.134.19"
Jun 4 10:58:12 solar portsentry[721]: attackalert: Host 24.42.134.19
  has been blocked via dropped route using command: "/sbin/ipchains -I
  input -s 24.42.134.19 -j DENY -l"
```

1. Source of trace:

This trace was found on GIAC at http://www.sans.org/y2k/060500.htm posted by Pierre Lamy.

2. Detect was generated by:

The above entries are from the syslog of the victim's machine. They were generated by the portsentry wrapper and a IPchains firewall.

3. Probability the source address was spoofed:

The source address is probably not spoofed because the transmission is using TCP.

4. Description of attack:

This attack is a scan of port 80 TCP (HTTP). This scan is used in hopes of finding a web server, then on subsequent attacks, exploit its vulnerabilities (e.g. CGI-BIN or PHP3). Fortunately, the attacker is not experienced and attempts this scan against a mail server.

5. Attack mechanism:

The attack works by scanning port 80 TCP. If the victim's machine responds by completing the three-way handshake, then the attacker knows he has found a web server. It is also possible, once he makes a connection to the victim's web server, to determine the characteristics of the web server software. This information can be helpful in staging the next attack to exploit any known vulnerabilities.

6. Correlation:

See Detect 7.

7. Evidence of active targeting:

This attack is going after a specific host, evidence of active targeting.

10 Detects with Analyses

David Blaine
6/6/2000

---

8. Severity:

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

Criticality = 4. The target machine is a mail server.

Lethality = 1. Normally, this attack would be considered lethal but the attacker focused his attentions on a mail server not a web server.

System Countermeasures = 5. Portsentry reported the illegal probe.

Network Countermeasures = 5. The firewall blocked the attack.

So, Severity = (4 + 1) – (5 + 5) = 5 – 10 = -5. This attack has no hope of succeeding here!

9. Defensive recommendation:

There are adequate countermeasures in place to deter this attack

10. Multiple choice test question based on above detect:

Given the attack is against a mail server, is the attacker a:

a) Experienced hacker?
b) Webmaster?
c) Script kiddie?
d) Contractor?

Answer: c)

GIAC Intrusion Detection Curriculum
Practical Assignment for SNAP San Jose
May 8 – 13, 2000

10 Detects with Analyses

David Blaine
6/6/2000

```
Detect 7

[Thu Jun 1 16:48:22 2000] [error] [client 206.47.244.91]
  File does not exist: /xxxxx/web/htmldocs/stock/stock_summary.php3
```

1. Source of trace:

This trace is another attack as reported by Pierre Lamy (Again, the URL can found at http://www.sans.org/y2k/060500.htm).

2. Detect was generated by:

This detect was generated by the host's syslog.

3. Probability the source address was spoofed:

The source address is probably not spoofed.

4. Description of attack:

This attack is an attempt against a web server (TCP port 80) in the hopes it is running PHP3. A vulnerable PHP3 web server could allow the attacker to run commands as root.

5. Attack mechanism:

As excerpted from http://cve.mitre.org in CAN-2000-0059:

A vulnerable PHP3 web server with safe_mode enabled does not properly filter shell metacharacters from commands that are executed by popen, which could allow remote attackers to execute commands.

6. Correlation:

See Detect 6. The mail server was mistakenly probed for web processes by another hostile host.

7. Evidence of active targeting:

The web server was targeted for this attack.

8. Severity:

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

Criticality = 4. The target machine is a web server.

Lethality = 5. If the web server was running PHP3, then the attacker could have compromised it.

David Blaine
6/6/2000

System Countermeasures = 5. The web server is not running PHP3.

Network Countermeasures = 2. The firewall must allow web traffic through to the web server. It is up to the web server's host-based security to protect it.

So, Severity = (4 + 5) – (5 + 2) = 9 – 7 = 2. Although the attack was not successful, this severity value shows that we should be on "full alert" for such attempts.

9. Defensive recommendation:

This web server should be maintained vigilantly:

a) Use swatch as a notification system to alert administrator's for anomalous events in system and application logs.
b) Keep up-to-date on patches for both the operating system and any applications running on this server.
c) Make sure the web server configuration does not allow unexpected root access (e.g. CGI-BIN or PHP3).

10. Multiple choice test question based on above detect:

This detect was found in:

a) The firewall log.
b) Web server's syslog.
c) Web server's HTTPD log.
d) My mailbox.

Answer: b)

GIAC Intrusion Detection Curriculum
Practical Assignment for SNAP San Jose
May 8 – 13, 2000

10 Detects with Analyses

David Blaine
6/6/2000

Detect 8:

```
[**] OVERFLOW-Named-ADM-NXT - 8.2->8.2.1 [**]
05/28-15:10:44.631945 194.231.20.98:4124 -> z.y.w.34:53
TCP TTL:40 TOS:0x0 ID:47543 DF
*****PA* Seq: 0x3CCB7F49 Ack: 0x4598A169 Win: 0x7D78
77 3B 20 72 6D 20 2F 72 6F 6F 74 2F 2E 62 61 73 w; rm /root/.bas
68 5F 68 69 73 74 6F 72 79 3B 20 72 6D 20 2D 72 h_history; rm -r
66 20 2F 76 61 72 2F 6E 61 6D 65 64 2F 41 44 4D f /var/named/ADM
52 4F 43 4B 53 3B 20 70 73 20 61 75 78 20 7C 20 ROCKS; ps aux |
67 72 65 70 20 65 67 67 64 72 6F 70 3B 0A grep eggdrop;.
```

1. Source of trace:

This trace was posted by Bryce Alexander to GIAC at http://www.sans.org/y2k/060500.htm.

2. Detect was generated by:

Snort intrusion detection system.

3. Probability the source address was spoofed:

The source address was probably not spoofed.

4. Description of attack:

Attack against TCP port 53 (DNS). Normally, this would be the classic ADM buffer overflow attack to gain root access. But as can be seen from the ASCII representation of the packet payload, the attacker has already gained root access and is now covering his tracks (He is removing the IRC bot known as eggdrop. It can also be used as a covert channel).

5. Attack mechanism:

This is a false positive on a buffer overflow due to a keyword match on Snort's alert filters. One would expect to see these commands when a hacker has already compromised the system and is attempting to cover his/her tracks. First is a remove command of the BASH history file in root's directory, then a remove command of everything in the folder /var/named/ADMROCKS, and finally, a command that looks to see if the program "eggdrop" is currently running in the active processes.

Correlation:

The classic ADM buffer overflow attack is very common, although I could not find a compromise quite like this one.

6. Evidence of active targeting:

Yes, the DNS server has been compromised.

7. Severity:

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

Criticality = 5. The target machine is a core infrastructure machine (e.g. a DNS server).

Lethality = 5. The DNS server has been compromised by the attacker. He has previously installed a covert channel bot (either he has stolen passwords or used the DNS server to stage other attacks either internally or externally).

System Countermeasures = 2. The DNS server is running a vulnerable version of BIND.

Network Countermeasures = 2. The firewall was permissive to allow this traffic to the DNS server.

This time Severity = (5 + 5) – (2 + 2) = 10 – 4 = 6. This attack has successfully compromised our infrastructure machine.

8. Defensive recommendation:

Based on the break-in indicated:

a) Rebuild DNS server with a patched version of BIND.
b) Restrict access to the DNS server either at the firewall or through access control lists (ACL) in BIND.

9. Multiple choice test question based on above detect:

The detect above indicates:

a) A hacker covering his tracks.
b) Normal DNS query.
c) DNS zone transfer.
d) ADM buffer overflow.

GIAC Intrusion Detection Curriculum
Practical Assignment for SNAP San Jose
May 8 – 13, 2000

10 Detects with Analyses

David Blaine
6/6/2000

Detect 9:

```
Jun 5 15:36:06 maunder portsentry[270]: attackalert:
  Connect from host: net-76-254.tuc.com/38.218.76.254 to TCP port: 139
```

1. Source of trace:

From Performance Systems International, Herndon VA posted to GIAC at http://www.sans.org/y2k/060700.htm.

2. Detect was generated by:

The detect was reported by portsentry, a host-based security system for Linux, in the victim's syslog.

3. Probability the source address was spoofed:

Given the detect shows TCP traffic, the source address was probably not spoofed.

4. Description of attack:

The attack is against TCP port 139 (NetBIOS). This is an attack for MS Windows machine or UNIX boxes running Samba that do not check for out of bounds data on TCP port 139. Alhough we need more information, it appears the tool used could be WIN Nuke. In order to make this determination, we would need to know if this attack included the TCP flag URG and an urgent value of 3. This would be an example of Denial of Service (DoS) attempt.

5. Attack mechanism:

In the event this was a WIN Nuke attack:

Reproduced from text found at http://www.rage.mircx.com/knowledge/tcpip-oob139.htm:

OOB 139, or WIN Nuke is probably one of the oldest and most used attacks on IRC. TCP/IP port 139 is used in MS Windows' NetBIOS protocol. This protocol was designed to be used across LAN networks only. The information this port can receive is extremely limited, and no checks are made on the incoming packets. If the packet received contains any data that doesn't fit what NetBIOS expects (out of bounds data), the operating system freezes. Most often this attack will result in a Windows Protection Error (blue screen). I have also heard of users hard drives becoming non-bootable after an OOB attack, and some have reported black screens.

In any case, you would not expect external probes of internal machines at TCP port 139.

6. Correlation:

David Blaine
6/6/2000

NetBIOS attacks are well known against MS Windows machines. Samba running on UNIX machines may also suffer from the same vulnerabilities (Although doubtful that the attack would cause a system lockup on UNIX. At most, I would suspect that the Samba service would crash).

7. Evidence of active targeting:

A UNIX system running Samba was targeted.

8. Severity:

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

Criticality = 3. A UNIX system running Samba was targeted.

Lethality = 3. Attacks on TCP port 139 can give away user passwords.

System Countermeasures = 5. Portsentry is installed and detected this attack.

Network Countermeasures = 2. The firewall was permissive to allow this external traffic to the Samba server.

Severity = (3 + 3) – (5 + 2) = 6 – 7 = -1. Going strictly by the severity level, our countermeasures are a sufficient deterrent against this attack. However, see the "Defensive recommendations" for a tighter security stance.

9. Defensive recommendation:

Here are my recommendations:

a) Block external connections to NetBIOS ports 137, 138 and 139 at the firewall.
b) Apply the latest official Microsoft patches to fix potential NetBIOS problems on Windows PC's.
c) Load the latest Samba release for UNIX to avoid similar bugs.

10. Multiple choice test question based on above detect:

What listens to TCP port 139 on a UNIX box?

a) Echo.
b) Samba.
c) PCAnywhere.
d) Sendmail.

Answer: b)

GIAC Intrusion Detection Curriculum
Practical Assignment for SNAP San Jose
May 8 – 13, 2000

10 Detects with Analyses

David Blaine
6/6/2000

Detect 10:

```
Jun 6 02:37:55 hosth snort[256]: SCAN-SYN FIN:
  63.226.11.117:53 -> a.b.c.19:53
Jun 6 02:37:55 hosth snort[256]: SCAN-SYN FIN:
  63.226.11.117:53 -> a.b.c.32:53
Jun 6 02:37:55 hosth snort[256]: SCAN-SYN FIN:
  63.226.11.117:53 -> a.b.c.33:53
Jun 6 02:37:56 hosth /kernel: Connection attempt
  to TCP a.b.c.62:53 from 63.226.11.117:53
Jun 6 02:37:56 hosth snort[256]: SCAN-SYN FIN:
  63.226.11.117:53 -> a.b.c.51:53
Jun 6 02:38:01 hosth snort[256]: spp_portscan:
  portscan status from 63.226.11.117: 17 connections across 17 hosts:
  TCP(17), UDP(0) STEALTH
Jun 6 02:38:01 hosth snort[256]: SCAN-SYN FIN:
  63.226.11.117:53 -> a.b.d.52:53
Jun 6 02:38:06 hosth snort[256]: SCAN-SYN FIN:
  63.226.11.117:53 -> a.b.e.79:53
Jun 6 02:38:07 hosth snort[256]: spp_portscan:
  portscan status from 63.226.11.117: 10 connections across 10 hosts:
  TCP(10), UDP(0) STEALTH
Jun 6 02:38:07 hosth snort[256]: SCAN-SYN FIN:
  63.226.11.117:53 -> a.b.e.91:53
Jun 6 02:38:26 hosth snort[256]: spp_portscan:
  End of portscan from 63.226.11.117
```

1. Source of trace:

This detect was posted to GIAC from Scottsdale Senior Center, Scottsdale AZ (Reference URL http://www.sans.org/y2k/060700.htm).

2. Detect was generated by:

Snort intrusion detection system.

3. Probability the source address was spoofed:

Because TCP is reported in the above trace, the source address is probably not spoofed.

4. Description of attack:

This is a port scan of a network of TCP port 53. This appears to be a network mapping attempt.

5. Attack mechanism:

This attack probably uses a port scanner like NMAP to quickly scan the entire network. This "stealth" scan uses TCP flags SYN and FIN together which should never happen normally. The attacker can use the results to detemine what machines are turned on (they will respond with a RST or reset). Given this information, he can stage more focused attacks in the future.

6. Correlation:

Most sites allow TCP port 53 into their network to allow DNS zone transfers. If the network does not restrict zone transfers to designated servers, then a hacker can use this port to map the network.

7. Evidence of active targeting:

This is a general scan of the network. No evidence of stealth is used in this scan as there is no attempt to avoid detection by employing a "low and slow" method.

8. Severity:

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

Criticality = 4. Scanning the network for DNS servers.

Lethality = 5. Attacks on TCP port 53 can cause buffer overflows and compromise root access.

System Countermeasures = 5. This posting does not specify how the systems are configured for security, but I will assume they are well maintained.

Network Countermeasures = 2. The firewall was permissive to allow this scan into our protected network.

Severity = (4 + 5) – (5 + 2) = 9 – 7 = 2. Our DNS servers could be vulnerable to an attack.

9. Defensive recommendation:

Add firewall rule to allow DNS zone transfers to and from registered external DNS secondaries only.

10. Multiple choice test question based on above detect:

This scan is a:

a) "low and slow" scan for DNS servers.
b) UDP port scan for DNS servers.
c) TCP port scan for DNS servers.
d) DNS query.

Answer: c)

GIAC Intrusion Detection Curriculum
Practical Assignment for SNAP San Jose
May 8 – 13, 2000

10 Detects with Analyses

David Blaine
6/6/2000
