Global Information Assurance Certification Paper

Practical Assignment submitted by: **Michael Schiller**

Detect 1

```
Jun 7 04:48:00 fwall 15 deny: TCP from 209.156.190.95.39557 to fwall.23
  seq 105B2, ack 0x0, win 8192, SYN
Jun 7 04:48:09 fwall 15 deny: TCP from 209.156.190.95.39557 to fwall.23
  seq 105B2, ack 0x0, win 8192, SYN
Jun 7 04:48:21 fwall 15 deny: TCP from 209.156.190.95.39557 to fwall.23
  seq 105B2, ack 0x0, win 8192, SYN
```

1. Source of trace

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

2. Detect was generated by:

   Probably a firewall log file: first field – date & time, next field – logical name of firewall device – fwall (or sanitized name of device), next field – a packet was denied from source address 209.156.190.95 with a source port of 39557 to the firewall on port 23 (telnet) with a sequence no. of 105B2, the acknowledgement bit was not set and there was an initial window size set of 8192 bytes, and the last field indicates that this packet was trying to initiate a connection (SYN). These three packets occurred very closely together (only 21 seconds apart). Also, the source port and sequence numbers do not increment or change at all (as they should). Very fishy. Looks like a crafted scan.

3. Probability the source address was spoofed

   Looks like the real address. Otherwise if a spoofed address was used, the telnet would not be successful. You can't use a spoofed address with this kind of attack.

4. Description of attack:

   A telnet attempt to the firewall. The hacker was trying to gain access to the firewall through a telnet. Once the hacker gained access to the firewall the hacker could then do some bad, bad things (like open up ports and be just a general nuisance, amongst other things). I get nervous just thinking about it.

5. Attack mechanism:

   The attack was a simple telnet to the firewall – trying to gain access to the firewall and perhaps brute force the password if the telnet was allowed. But, since it came from the user's own ISP (couldn't necessarily know that from just from this trace) the attacker has probably compromised the machine 209.156.190.95, the source address of these packets. That's not good, if that machine at the user's own ISP is compromised. So probably a phone call to the ISP would be a good idea, to see if the ISP does in fact have a compromised machine.

6. Correlations:

   This kinda' thing happens all the time (but hopefully not from a host at your own ISP). I see attempted telnets to my routers and firewall in my router and firewall log files quite often.

This came from my PIX firewall log on 6/14 (trace has been sanitized):

Jun 14 16:27:44 [10.10.8.1.2.2] %PIX-2-106001: Inbound TCP connection denied
from 204.117.70.5/1100 to myhost/23 flags SYN
Jun 14 16:27:44 [10.10.8.1.2.2] %PIX-2-106001: Inbound TCP connection denied
from 204.117.70.5/1100 to myhost/23 flags SYN
Jun 14 16:27:44 [10.10.8.1.2.2] %PIX-2-106001: Inbound TCP connection denied
from 204.117.70.5/1100 to myhost/23 flags SYN

Someone trying to telnet and gain access to a firewall or router (as evidenced by the following CVE candidates).

At http://cve.mitre.org the following CVE Candidates apply:

CAN-1999-507 ** CANDIDATE (under review) ** An account on a router, firewall, or other network device has a guessable password

CAN-1999-508 ** CANDIDATE (under review) ** An account on a router, firewall, or other network device has a default, null, blank, or missing password.

CAN-1999-0619 ** CANDIDATE (under review) ** The Telnet service is running.

7. Evidence of active targeting:

Trying to hack in to the firewall through port 23 (telnet). This person was definitely going after a specific host (the firewall in this case) through a telnet. Pretty fishy if you ask me. Also the source port and sequence numbers did not increment as they should. Definitely a crafted scan.

8. Severity:

(Criticality: 4 + Lethality: 4) – (System: 4 + Net Countermeasures: 4) = Severity: 0
Although somewhat severe since the machine that launched the attack was probably a compromised machine at the user's ISP. But the firewall blocked the attempted telnet.

9. Defensive recommendation:

Defenses are fine, firewall blocks telnet from the outside. However, we definitely need to call the user's ISP and tell them they probably have a compromised machine from which this attack was launched.

10. Multiple choice test question

a) SYN flood
b) Attempted telnet to firewall
c) SYN/ACK flood
d) TROJAN search on port 39557

Answer: b

Detect 2

```
May 19 02:52:41 firewall kernel: Packet log: input REJECT eth0 PROTO=17
```

```
  208.135.129.62:137 MY.HOST.211:137 L=78 S=0x00 I=49339 F=0x0000 T=122 (#3)
May 20 02:52:24 firewall kernel: Packet log: input REJECT eth0 PROTO=17
  208.135.129.62:137 MY.HOST.211:137 L=78 S=0x00 I=34144 F=0x0000 T=122 (#3)
May 21 02:51:54 firewall kernel: Packet log: input REJECT eth0 PROTO=17
  208.135.129.62:137 MY.HOST.211:137 L=78 S=0x00 I=8328 F=0x0000 T=122 (#3)
May 22 02:51:53 firewall kernel: Packet log: input REJECT eth0 PROTO=17
  208.135.129.62:137 MY.HOST.211:137 L=78 S=0x00 I=50493 F=0x0000 T=122 (#3)
May 23 02:26:46 firewall kernel: Packet log: input REJECT eth0 PROTO=17
  208.135.129.62:137 MY.HOST.211:137 L=78 S=0x00 I=37647 F=0x0000 T=122 (#3)
May 24 02:26:45 firewall kernel: Packet log: input REJECT eth0 PROTO=17
  208.135.129.62:137 MY.HOST.211:137 L=78 S=0x00 I=1793 F=0x0000 T=122 (#3)
```

1. Source of trace

   This correlation was found at on May 19 by Stephen Northcutt
   http://www.sans.org/y2k/053000-1000.htm

2. Detect was generated by:

   Probably a firewall log file. 1st field – date and time, next field – descriptive text from the firewall log "firewall kernal: Packet log: input REJECT on interface eth0", protocol 17 indicates UDP protocol, 208.135.129.62 is the source address with a source port of 137, MY.HOST.211 is the destination address, 137 is the destination port, the length of the packet was 78 bytes. Several frames at 20 or 30 minute intervals.

3. Probability the source address was spoofed

   This is probably the real source address. Otherwise the hacker wouldn't know if the Netbios DNS query was successful.

4. Description of attack:

   The attack was an attempt to resolve hosts using Microsofts Netbios name services on port 137 UDP (Protocol 17) and then perhaps access files using Microsofts SMB file sharing capabilities through successive ports (138, 139)

5. Attack mechanism:

   The attack is Netbios Name Query. If UDP port 137 were open and the destination host was there to answer the hacker could get information about hosts on this network and then perhaps set up file sharing through SMB.

6. Correlations:

   This attack was seen on many occasions previously and continues to be an ongoing problem.

   For instance, this correlation was found on my PIX firewall log:

   Jun 15 03:52:56 [10.10.10.254.2.2] %PIX-2-106006: Deny inbound UDP from 213.140
   .0.17/137 to myhost.186/137
   Jun 15 03:52:56 [10.10.10.254.2.2] %PIX-2-106006: Deny inbound UDP from 213.140
   .0.17/137 to myhost.186/137

   Other correlations found at:
   http://www.sans.org/y2k/053000-1000.htm

Someone trying to gain access to a Windows machine using file sharing and name services. The following CVE numbers apply:

CAN-1999-0518 ** CANDIDATE (under review) ** A NETBIOS/SMB share password is guessable.

CAN-1999-0519 ** CANDIDATE (under review) ** A NETBIOS/SMB share password is the default, null, or missing.

CAN-1999-0520 ** CANDIDATE (under review) ** A system-critical NETBIOS/SMB share has inappropriate access control.

7. Evidence of active targeting:

This hacker is going after a specific host (MY.HOST.211) to gain IDENT and DNS info. With specific port 137 NETBIOS (with SMB file sharing and name services as targets as well)

8. Severity:

(Critical:2 + Lethal:2) – (System:4 + Net Countermeasures:5) = Severity: -5
DNS server generally well maintained and patched. Firewall blocks this scan. Not very severe.

9. Defensive recommendation:

Defenses are fine.
Scan was blocked at the firewall. Good job.

10. Multiple choice test question

a) MS Netbios Name Server Flood
b) MS Netbios Name Server Version Scan
c) MS Netbios Name Server Host Query
d) MS Netbios Name Server Buffer Overflow

Answer: c

Detect 3

```
May 27 02:30:21 firewall kernel: Packet log: input DENY eth0 PROTO=6
  203.134.67.212:1634 MY.HOST.211:12345 L=48 S=0x00 I=29459 F=0x4000
T=114
  SYN (#12)
May 27 02:30:24 firewall kernel: Packet log: input DENY eth0 PROTO=6
  203.134.67.212:1634 MY.HOST.211:12345 L=48 S=0x00 I=29715 F=0x4000
T=114
  SYN (#12)
```

1. Source of trace

http://www.sans.org/y2k/053000-1000.htm

2. Detect was generated by:

Probably a firewall log file. Frames were only a few seconds apart to the same host. Proto 6 is TCP. Packets were denied. Destination port 12345 is NETBUS trojan. Source port did not increment.

3. Probability the source address was spoofed

The real address was probably used (not a spoofed address) otherwise the packet would not have been returned.

4. Description of attack:

NETBUS trojan port scan. Looking for a compromised machine. Protocol 6 is TCP. These are SYN packets (trying to initiate a connection). Destination port 12345 is the NETBUS trojan port. A trojan program sits on a compromised host. This scan is searching for potential compromised hosts.

5. Attack mechanism

If a machine on the inside of the firewall were compromised with this particular NETBUS trojan, it could send out packets on port 12345 and open up that port in the process (unless the firewall blocked that outgoing port). That's what this scan is looking for (compromised machines on the inside of the firewall).

6. Correlations:

This attack is probably seen quite often.

Here is an example from my own firewall logs from June 19:

Jun 19 13:58:26 [10.10.10.254.2.2] %PIX-2-106001: Inbound TCP connection denied from 24.240.11.147/1765 to myhost.74/12345 flags SYN
Jun 19 13:58:26 [10.10.10.254.2.2] %PIX-2-106001: Inbound TCP connection denied from 24.240.11.147/1765 to myhost.74/12345 flags SYN
Jun 19 13:58:26 [10.10.10.254.2.2] %PIX-2-106001: Inbound TCP connection denied from 24.240.11.147/1765 to myhost.74/12345 flags SYN
Jun 19 13:58:26 [10.10.10.254.2.2] %PIX-2-106001: Inbound TCP connection denied from 24.240.11.147/1765 to myhost.74/12345 flags SYN

Someone was trying to gain access to a host on port 12345 (NETBUS trojan). The following CVE numbers apply:

CAN-1999-0660 ** CANDIDATE (under review) ** A hacker utility or Trojan Horse is installed on a system, e.g. NetBus, Back Orifice, Rootkit, etc.

7. Evidence of active targeting:

Going after a specific port on a specific host (NETBUS trojan in this case). Definitely suspicious behavior.

8. Severity:

(Critical: 5 + Lethality: 5) – (System: 4 + Net Countermeasures: 4) = Severity: 2
Probably not that severe at this point since the host MY.HOST.211 was not compromised and the firewall blocked that incoming port

9. Defensive recommendation:

Defenses worked in this case. Probably want to run tripwire or some equivalent on MY.HOST.211 if it is often attacked to reduce the vulnerability of this machine in the future (before it happens).

10. Multiple choice test question

a) SYN flooding
b) SUN RPC attack
c) NETBUS trojan
d) SubSeven trojan

Answer: c

Detect 4

```
Jun 11 21:58:02 211.108.224.4:2666 -> a.b.c.71:111 SYN **S*****
Jun 11 21:58:02 211.108.224.4:2666 -> a.b.c.101:111 SYN **S*****
Jun 11 21:58:02 211.108.224.4:2666 -> a.b.c.225:111 SYN **S*****
Jun 11 21:58:02 211.108.224.4:2666 -> a.b.d.52:111 SYN **S*****
Jun 11 21:58:02 211.108.224.4:2666 -> a.b.e.66:111 SYN **S*****
Jun 11 21:58:02 211.108.224.4:2666 -> a.b.e.67:111 SYN **S*****
Jun 11 21:58:02 211.108.224.4:2666 -> a.b.e.97:111 SYN **S*****
Jun 11 21:58:02 211.108.224.4:2666 -> a.b.f.22:111 SYN **S*****
Jun 11 21:58:02 211.108.224.4:2666 -> a.b.f.21:111 SYN **S*****
Jun 11 21:58:02 211.108.224.4:2666 -> a.b.f.166:111 SYN **S*****
```

1. Source of trace

http://www.sans.org/y2k/061500.htm

2. Detect was generated by:

Looks like a snort trace. Many different machines were scanned in a very short period of time. All on port 111 (SUN RPC). Definitely not too discrete. Source port did not increment. Definitely a script.

3. Probability the source address was spoofed

Probably not a spoofed address (if the address were spoofed the RPC commands with fail if the ports were open).

4. Description of attack:

Attack attempting to come in or port 111 (SUN RPC commands). The SYN flag is set. Different machines are attempted each time in a relatively short amount of time.

5. Attack mechanism:

This attack attempts to access TCP Port 111 (SUN RPC commands). If that port were open, access to a machine could be gained. System Info, perhaps file access, etc. could be gained.

6. Correlations:

This has been seen before on many occasions.

Here is a correlation from http://www.sans.org/y2k/061300.htm

workspace international inc. (NETBLK-WOIN-137-192)
2700 barrett lakes blvd
suite 700
kennesaw, GA 30144 USA

```
Jun 5 22:41:18 www portsentry[16450]: attackalert:
SYN/Normal scan from host: kungfoo.globali.net/207.91.110.40
to TCP port: 111
```

Also this correlation from http://www.sans.org/y2k/061400.htm:
Anton -- I dug around in the logs for some correlation and found that one of our visitors was also looking for Sun boxes: (All times are GMT)

Jun 10 00:40:40 outer_screen 5784406:INBOUND denied tcp 208.60.175.68(111) -> firewall_a(111), 1 packet
Jun 10 00:40:41 outer_screen 5784408:INBOUND denied tcp 208.60.175.68(111) -> firewall_b(111), 1 packet
Jun 10 00:40:47 outer_screen 5784412:INBOUND denied tcp 208.60.175.68(111) -> firewall_c(111), 1 packet

Someone was trying to gain access to a host through SUN RPC commands. The following CVE numbers apply:

CAN-1999-0613 ** CANDIDATE (under review) ** The rpc.sprayd service is running.
CAN-1999-0625 ** CANDIDATE (under review) ** The rpc.rquotad service is running.
CAN-1999-0632 ** CANDIDATE (under review) ** The RPC portmapper service is running.
CAN-1999-0795 ** CANDIDATE (under review) ** The NIS+ rpc.nisd server allows remote attackers to execute certain RPC calls without authentication to obtain system information, disable logging, or modify caches.

7. Evidence of active targeting:

In this case many different machines have been targeted with scans occurring very quickly after one another. Definitely an obvious scan.

8. Severity

(Critical:4 + Lethal:4) – (System:4 + Net Countermeasures:4) = Severity: 0
Not too severe since scans were blocked at the firewall.

9. Defensive recommendation:

Defenses are fine. The attack was blocked by the firewall. We survived yet another attack. Wheeww.

10. Multiple choice test question:

    a) Port Scan
    b) Host Scan
    c) RPC Info query
    d) SYN flooding

Detect 5

```
01:25:44: Deny inbound udp src 63.29.241.229/1039 dst x.x.x.12/31337
01:25:44: Deny inbound udp src 63.29.241.229/1039 dst x.x.x.28/31337
01:25:44: Deny inbound udp src 63.29.241.229/1039 dst x.x.x.137/31337
01:25:44: Deny inbound udp src 63.29.241.229/1039 dst x.x.x.209/31337
01:25:44: Deny inbound udp src 63.29.241.229/1039 dst x.x.x.191/31337
01:25:44: Deny inbound UDP from 63.29.241.229/1039 to x.x.x.152/31337
01:25:44: Deny inbound udp src 63.29.241.229/1039 dst x.x.x.176/31337
01:25:44: Deny inbound udp src 63.29.241.229/1039 dst x.x.x.208/31337
01:25:44: Deny inbound udp src 63.29.241.229/1039 dst x.x.x.106/31337
01:25:44: Deny inbound udp src 63.29.241.229/1039 dst x.x.x.122/31337
```

1. Source of trace

    http://www.sans.org/y2k/061100.htm

2. Detect was generated by:

    Could be a firewall log file. All frames are trying to hit destination port 31337 (Back Orifice) on different hosts. The packets were denied at the firewall. Source port did not change. Certainly a script.

3. Probability the source address was spoofed

    Source address was probably not spoofed. Otherwise Back Orifice would not work, since the frames would not get back to the origin.

4. Description of the attack:

    Trojan program (Back Orifice), if sitting on a compromised machine will respond to 31337 and that's what these frames are trying to do (connect to a machine with Back Orifice sitting on it).

5. Attack mechanism:

    The attack works by having a compromised machine running Back Orifice (port 31337) and responding to the incoming packets on that port.

6. Correlations:

    This type of attack occurs quite often.

    This attack was found from my firewall log files on June 19:

Jun 19 14:00:41 [10.10.10.254.2.2] %PIX-2-106001: Inbound TCP connection denied from 24.240.11.147/1831 to myhost.74/31337 flags SYN
Jun 19 14:00:41 [10.10.10.254.2.2] %PIX-2-106001: Inbound TCP connection denied from 24.240.11.147/1831 to myhost.74/31337 flags SYN
Jun 19 14:00:41 [10.10.10.254.2.2] %PIX-2-106001: Inbound TCP connection denied from 24.240.11.147/1831 to myhost.74/31337 flags SYN
Jun 19 14:00:41 [10.10.10.254.2.2] %PIX-2-106001: Inbound TCP connection denied from 24.240.11.147/1831 to myhost.74/31337 flags SYN

Source address resolved to: 24-240-11-147.hsacorp.net

Someone was using a trojan horse program Back Orifice to try to gain access to a remote host. The following CVE numbers apply:

CAN-1999-0660 ** CANDIDATE (under review) ** A hacker utility or Trojan Horse is installed on a system, e.g. NetBus, Back Orifice, Rootkit, etc.

7. Evidence of active targeting:

These frames came in one after the other all to destination port 31337 (Back Orifice) to many different hosts. The hacker was actively trying to find any host at this site that might have had the program loaded on it.

8. Severity:

(Critical: 3 + Lethal: 4) – (System: 3 + Net Countermeasures: 4) = Severity: 0
Not too severe since this machine didn't answer back and the port was definitely denied at the firewall.

9. Defensive recommendation:

No defense needed. Port was blocked at the firewall. But could add tripwire for extra protection on the more critical hosts being scanned.

10. Multiple choice test question

a) SUN RPC info query
b) Silencer Trojan
c) Port Scan
d) Back Orifice

Answer: d

Detect 6

```
Jun 6 22:59:25 213.6.15.254:3662 -> z.y.w.98:23 SYN **S*****
Jun 6 22:59:25 213.6.15.254:3663 -> z.y.w.98:139 SYN **S*****
Jun 6 22:59:26 213.6.15.254:58110 -> z.y.w.98:21 SYN 2*S*****
RESERVEDBITS
Jun 6 22:59:26 213.6.15.254:58111 -> z.y.w.98:21 NULL ********
Jun 6 22:59:26 213.6.15.254:58112 -> z.y.w.98:21 NMAPID **SF*P*U
```

1. Source of trace

http://www.sans.org/y2k/061000.htm

2. Detect was generated by:

   Could be TCPDUMP. Frames are all directed to the same machine. Different ports (23 telnet), (21 ftp), (139 Netbios). Also, the flags are being set in an invalid way such as SF set on the last frame. Looks like an NMAP scan. Could be used to signature an operating system as well.

3. Probability the source address was spoofed

   Source address is probably not spoofed. Otherwise NMAP would not receive the mapping data back.

4. Description of the attack:

   This is a port scan. The hacker is trying to see which ports are open on a particular host. Also, by setting different invalid flag combinations the hacker can signature what operating system the host is running. Pretty cool. All part of NMAP (a widely available utility for doing port scans).

5. Attack mechanism:

   This is just a port scan. But because the scan uses different invalid flag combinations it can signature the hosts operating system, since each operating system responds differently to different invalid TCP flag combinations.

6. Correlations:

   This type of NMAP scan occurs all the time.

   The following trace was obtained from my own firewall logs:

   Obtained from: http://www.sans.org/y2k/061400.htm

   Binette reported the following scan from June 11.

   Jun 11 01:27:21 cc1014244-a kernel: securityalert: tcp if=ef0 from 24.16.161.96:2042 to 24.3.21.199 on unserved port 27374
   Jun 11 01:46:47 cc1014244-a kernel: securityalert: udp if=ef0 from 24.1.87.167:137 to 24.3.21.199 on unserved port 137
   Jun 11 02:11:31 cc1014244-a kernel: securityalert: tcp if=ef0 from 159.226.253.195:1979 to 24.3.21.199 on unserved port 111

   The following CVE numbers apply:

   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.
   CAN-2000-0324 ** CANDIDATE (under review) ** pcAnywhere 8.x and 9.x allows remote attackers to cause a denial of service via a TCP SYN scan, e.g. by nmap.

7. Evidence of active targeting:

   The same host was targeted with each packet. So a particular host was scanned for many exploitable ports using NMAP.

8. Severity:

   This is mapping work. The kind of work you do prior to an attack.
   (Criticality: 3 + Lethal: 3) – (System: 4 + Net Countermeasures: 4) = Severity: -2

9. Defensive recommendation:

   The machine is probably hardened. If not it needs to be. Most ports are blocked at the firewall. The ones that aren't should have the latest patches for those processes.

10. Multiple choice test question:

    a) SMURF attack
    b) Port Scan using NMAP
    c) Host Scan
    d) Classic Denial of service attack

    Answer: b

Detect 7

```
[**] SCAN-SYN FIN [**] 06/06-23:58:37.557539 194.247.87.235:53 ->
z.y.w.98:53 TCP TTL:27 TOS:0x0 ID:39426 **SF**** Seq: 0x64C2AF11 Ack:
0x4979F54A Win: 0x404 00 00 00 00 00 00
[**] SCAN-SYN FIN [**] 06/06-23:58:47.792897 194.247.87.235:53 ->
z.y.w.98:53 TCP TTL:27 TOS:0x0 ID:39426 **SF**** Seq: 0x5D2F00A4 Ack:
0x4ED983D3 Win: 0x404 00 00 00 00 00 00
[**] SCAN-SYN FIN [**] 06/06-23:58:55.471597 194.247.87.235:53 ->
z.y.w.98:53 TCP TTL:27 TOS:0x0 ID:39426 **SF**** Seq: 0x5EF94849 Ack:
0x4186CC99 Win: 0x404 00 00 00 00 00 00 ......
[**] SCAN-SYN FIN [**] 06/06-23:59:38.689515 194.247.87.235:53 ->
z.y.w.98:53 TCP TTL:27 TOS:0x0 ID:39426 **SF**** Seq: 0x40E6B24A Ack:
0x17BEA20E Win: 0x404 00 00 00 00 00 00 ......
```

1. Source of trace

   http://www.sans.org/y2k/061000.htm

2. Detect was generated by:

   A snort trace for sure. SYN FIN flags are set. This behavior is not valid behavior. You can't have both SYN and FIN flags set in the same packet. An attack against port 53 (DNS).

3. Probability the source address was spoofed

   SYN-FIN DNS exploit.
   Address probably not spoofed. But could be.

   If its just taking advantage of a denial of service exploit for instance, the attacker doesn't necessarily have to use a legitimate source address. So the address could be spoofed.

4. Description of attack:

   SYN-FIN DNS exploit. Several different packets timed closely together, all to the same machine. An attempt to exploit potential vulnerabilities in DNS. Poor implementations of DNS might be exploited and compromised from this illegal packet.

5. Attack mechanism:

   SYN-FIN is an invalid combination of TCP flags. Some versions of DNS running on certain OSs could be vulnerable to such a combination. All to the DNS port 53. For instance as described in CVE-1999-0010 a denial of service attack could be successfully carried out in certain DNS implementations with this type of attack.

6. Correlations:

   This type of attack is probably fairly common.

   I found plenty of attempts to access port 53 on my firewall. This particular trace is from June 14, 2000:

   Jun 14 14:01:02 [10.10.10.254.2.2] %PIX-2-106007: Deny inbound UDP from 165.247
   .153.249/10 to myhost.240/53 due to DNS Query
   Jun 14 14:01:02 [10.10.10.254.2.2] %PIX-2-106007: Deny inbound UDP from 165.247
   .153.249/10 to myhost.240/53 due to DNS Query
   Jun 14 14:01:02 [10.10.10.254.2.2] %PIX-2-106007: Deny inbound UDP from 165.247
   .153.249/10 to myhost.240/53 due to DNS Query
   Jun 14 14:01:02 [10.10.10.254.2.2] %PIX-2-106007: Deny inbound UDP from 165.247
   .153.249/10 to myhost.240/53 due to DNS Query

   The following CVE numbers apply to this attack:

   CVE-1999-0010 Denial of Service vulnerability in BIND 8 Releases via maliciously formatted DNS messages.

7. Evidence of active targeting:

   194.247.87.235 was the target for all of the DNS SYN-FIN packets. A particular host in this case was singled out.

8. Severity:

   Probably not too severe. SYN-FIN is an invalid combination. If current versions and patches are used on this machine, it shouldn't be too much of a problem.
   (Critical: 3 + Lethal: 4) – (System: 3 + Net Countermeasures: 3) = Severity: 1
   Just load the latest patches and block DNS to 192.247.87.235 at the firewall if 192.247.87.235 is not a DNS server.

9. Defensive recommendation:

   SYN-FIN is an invalid combination. If current versions and patches are used on this machine, it shouldn't be too much of a problem. Just load the latest patches and block DNS to 192.247.87.235 at the firewall if 192.247.87.235 is not a DNS server.

10. Multiple choice test question:

a) Trojan
b) SYN flood
c) DNS ZONE SF exploit
d) DNS name lookup

Answer: c

Detect 8

```
Jun 09 2000 00:40:14: Inbound TCP connection denied
  from 206.249.181.109/4148 to x.x.x.31/1243 flags SYN
Jun 09 2000 00:40:14: Inbound TCP connection denied
  from 206.249.181.109/4149 to x.x.x.31/1243 flags SYN
Jun 09 2000 00:40:17: Inbound TCP connection denied
  from 206.249.181.109/4148 to x.x.x.31/1243 flags SYN
Jun 09 2000 00:40:17: Inbound TCP connection denied
  from 206.249.181.109/4149 to x.x.x.31/1243 flags SYN
Jun 09 2000 00:40:27: Inbound TCP connection denied
  from 206.249.181.109/4304 to x.x.x.31/27374 flags SYN
Jun 09 2000 00:40:27: Inbound TCP connection denied
  from 206.249.181.109/4305 to x.x.x.31/27374 flags SYN
Jun 09 2000 00:40:30: Inbound TCP connection denied
  from 206.249.181.109/4304 to x.x.x.31/27374 flags SYN
Jun 09 2000 00:40:30: Inbound TCP connection denied
  from 206.249.181.109/4305 to x.x.x.31/27374 flags SYN
```

1. Source of trace

   http://www.sans.org/y2k/061100.htm

2. Detect was generated by:

   Probably a firewall log file trace.

   Ports 1243 and 27374 are Sub Seven Trojan ports.

3. Probability the source address was spoofed

   Source address was probably not spoofed. Otherwise this type of attack (Sub Seven Trojan) wouldn't work.

4. Description of the attack:

   It's a Sub Seven Trojan attack both old and new versions (on ports 1243 and 27374). If a machine has been compromised, the Sub Seven Trojan would be running on it and respond to the packets on port 1243 or 27374. Then the hacker would own those machines if he were able to come in on those ports. That's how a trojan program works. You basically have a compromised machine owned by the hacker.

5. Attack mechanism:

The attack was an attempt to gain access to a potentially compromised machine at x.x.x.31 through the Sub Seven Trojan.

6. Correlations:

   This type of attack probably happens pretty often.

   Once again, I found correlations off my own firewall:

   Jun 15 18:32:16 [10.10.10.254.2.2] %PIX-2-106001: Inbound TCP connection denied from 204.191.209.82/5057 to myhost.113/1243 flags SYN
   Jun 15 18:32:16 [10.10.10.254.2.2] %PIX-2-106001: Inbound TCP connection denied from 204.191.209.82/5057 to myhost.113/1243 flags SYN
   Jun 15 18:32:16 [10.10.10.254.2.2] %PIX-2-106001: Inbound TCP connection denied from 204.191.209.82/5057 to myhost.113/1243 flags SYN

   The following CVE numbers apply to this attack:

   CAN-1999-0660 ** CANDIDATE (under review) ** A hacker utility or Trojan Horse is installed on a system, e.g. NetBus, Back Orifice, Rootkit, etc.

7. Evidence of active targeting:

   A single machine was targeted for the Sub Seven Trojan. That machine was x.x.x.31.

8. Severity:

   (Critical: 4 + Lethal: 4) – (System: 4 + Net Countermeasures: 3) = Severity: 1
   Probably not too severe. We were protected at the firewall.

9. Defensive recommendation:

   Probably not too critical since that port should be blocked at the firewall. Also could run tripwire on x.x.x.31 to alert of any potential compromises on that machine.

10. Multiple choice test question

    a) NETBUS
    b) Spy Sender
    c) Backdoor
    d) Sub Seven

    Answer: d

Detect 9

```
May 29 19:43:41 morannon named[14021]: unapproved query
  from [203.149.232.6].3203 for "version.bind"
May 29 19:43:41 morannon named[14021]: unapproved query
  from [203.149.232.6].3237 for "version.bind"
May 29 19:43:41 morannon named[14021]: unapproved query
  from [203.149.232.6].3218 for "version.bind"
May 29 19:43:41 morannon named[14021]: unapproved query
```

```
  from [203.149.232.6].3374 for "version.bind"
May 29 19:43:41 morannon named[14021]: unapproved query
  from [203.149.232.6].3402 for "version.bind"
May 29 19:43:41 morannon named[14021]: unapproved query
  from [203.149.232.6].3245 for "version.bind"
```

1. Source of trace

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

2. Detect was generated by:

   Could be a UNIX syslog (or maybe a firewall log? – probably not). Not snort or TCP dump. All attempts closely together in time from same host to same host.

3. Probability the source address was spoofed

   Source address was probably not spoofed.

   Otherwise this user would not get the resulting packets returned back.

4. Description of the attack:

   Attack against DNS (the name daemon) looking for the version of bind that is running on this particular host. Knowing the version of bind helps the hacker exploit your machine. The hacker can then collect all the known exploits for that particular version of bind and run through them to see if any of them work. This particular attack failed and was blocked as evidenced by this log file.

5. Attack mechanism:

   Attack against DNS (the name daemon) looking for the version of bind that is running on this particular host. Knowing the version of bind helps the hacker exploit your machine. The hacker can then collect all the known exploits for that particular version of bind and run through them to see if any of them work. This particular attack failed and was blocked as evidenced by this log file.

6. Correlations:

   These types of attacks occur all the time. There were references in the source file.

   Looks like someone _really_ likes our DNS server...
   person: Ma Win Lu
   address: Ever Star Merchandise Co.,Ltd
   address: 7F, No.6, Lane 123, Jung-Jeng Rd.,
   address: Shin Dian, Taipei Hsien
   address: Taiwan, R.O.C

   Now if they could've only searched for something more interesting than "version.bind" - I get enough queries of that kind as it is... (quick count reveals over 40 of them the last month... ugh – from April, 2000)

   The following CVE numbers may apply to this particular attack:

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

7. Evidence of active targeting:

   A specific host was targeted in this case. All with the same bind version query. Very suspicious indeed.

8. Severity:

   (Critical: 2 + Lethal: 2) – (System: 4 + Net Countermeasures: 3) = Severity: -3
   Not real severe. We did not respond to this query.

9. Defensive recommendation:
   Make sure the machine does not allow this type of query. Could be blocked at the firewall. Harden the machine down by using the latest versions of bind with the latest OS patches if not already done.

10. Multiple choice question:

    a) DNS Zone Transfer
    b) DNS Inverse Query
    c) DNS Version Scan
    d) DNS Buffer Overflow

    Answer: c

Detect 10

```
05/24 21:52:21.614162 208.201.208.71.2382 > 10.0.0.3.80:
P 3190942928:3190942960(32) ack 2272436265 win 32120 (DF)
(ttl 48, id 42496)
0000: 4500 0048 a600 4000 3006 3d37 d0c9 d047 E..H..@.0.=7...G
0010: 0a00 0003 094e 0050 be31 ecd0 8772 a029 .d...N.P.1...r.)
0020: 5018 7d78 638e 0000 4745 5420 2f63 6769 P.}xc...GET /cgi
0030: 2d62 696e 2f70 6866 0a00 0000 0000 00e0 -bin/phf........
0040: 685b 0240 b093 0408 h[.@....05/24 21:52:21.905354

208.201.208.71.2403 > 10.0.0.3.80:
P 3195509252:3195509284(32) ack 2272736277 win 32120 (DF)
(ttl 48, id 42556)
0000: 4500 0048 a63c 4000 3006 3cfb d0c9 d047 E..H.<@.0.<....G
0010: 0a00 0003 0963 0050 be77 9a04 8777 3415 .d...c.P.w...w4.
0020: 5018 7d78 ee5e 0000 4745 5420 2f63 6769 P.}x.^..GET /cgi
0030: 2d62 696e 2f74 6573 742d 6367 690a 00e0 -bin/test-cgi...
0040: 685b 0240 b093 0408 h[.@....05/24 21:52:22.183000

208.201.208.71.2423 > 10.0.0.3.80:
P 3193478365:3193478397(32) ack 2273043925 win 32120 (DF)
(ttl 48, id 42609)
0000: 4500 0048 a671 4000 3006 3cc6 d0c9 d047 E..H.q@.0.<....G
0010: 0a00 0003 0977 0050 be58 9cdd 877b e5d5 .d...w.P.X...{..
```

```
0020: 5018 7d78 aa9d 0000 4745 5420 2f63 6769 P.}x....GET /cgi
0030: 2d62 696e 2f68 616e 646c 6572 0a00 00e0 -bin/handler....
0040: 685b 0240 b093 0408 h[.@....
```

1. Source of trace

   http://www.sans.org/y2k/052800-1130.htm

2. Detect was generated by:

   Looks like a network analyzer program of some sort. Not sure which one. Port 80 (http was attempted). Same host each time. See the plain text in the hex dump "cgi-bin". This is the type of attack that this is.

3. Probability the source address was spoofed

   The source address was probably not spoofed. Otherwise this particular attack wouldn't work. If you don't get the packets returned to you, this attack won't work.

4. Description of attack:

   Attack against a web server. Looking for a "stub" server which many system owners are unaware of (often with default settings easily hacked into). The CGI-BIN script can search for many of these well-known script vulnerabilities.

5. Attack mechanism:

   The attack attempts to access a web server. This attack is probably looking for a "stub" server with default settings that some system owner maybe unaware. And this makes it easily hacked into and no one is checking the logs (perhaps). CGI-BIN scripts are an opportunity for attackers because they can check for many well-known script vulnerabilities. If this attack was successful then perhaps arbitrary command execution could be carried out or perhaps files could be read if this attack succeeded.

6. Correlations:

   This attack probably happens fairly often.

   This reference is from http://www.sans.org/y2k/042900.htm

   On 25th Apr 2000 at 22:03 (UTC) detected a series of attacks on many web servers on campus (approx. 50). All machine were probed for various cgi scripts with well known vulnerabilitites. The attacker also tried to establish ftp connections to each server but, so far I have not found out what they tried. These attacks appear to have originated from 130.239.160.220.

   The following CVE numbers may apply:

   CVE-1999-0021 Arbitrary command execution via buffer overflow in Count.cgi (wwwcount) cgi-bin program.

   CVE-1999-0066 AnyForm CGI remote execution

   CVE-1999-0068 CGI PHP mylog script allows an attacker to read any file on the target server.

7. Evidence of active targeting

   10.0.0.3 was targeted in all these packets. Specific CGI-BIN attack was attempted.

8. Severity:

   (Critical: 4 + Lethal: 4) – (System: 4 + Net Countermeasures: 3) = Severity: 1

   Not real severe. This type of attack was blocked.

9. Defensive recommendation:

   Defenses are okay if the host 10.0.0.3 has all the latest patches and the OS is hardened and there is no stub server running with any script vulnerabilities. I don't think there was any problem in this particular case.

10. Multiple choice question

    a) CGI-BIN attack
    b) Sub Seven
    c) Buffer Overflow
    d) Denial of Service

    Answer: a