Global Information Assurance Certification Paper

SANS GIAC San Jose Practical Exam
Prepared by Thomas Stocking

Detect 1

```
May 23 04:24:29 firewall kernel: Packet log: input DENY eth0
  PROTO=6 193.129.252.129:3163 MY.HOST.211:98
  L=60 S=0x00 I=32300 F=0x4000 T=47 SYN (#11)
May 23 04:24:32 firewall kernel: Packet log: input DENY eth0
  PROTO=6 193.129.252.129:3163 MY.HOST.211:98
  L=60 S=0x00 I=33894 F=0x4000 T=47 SYN (#11)
May 26 13:32:45 firewall kernel: Packet log: input DENY eth0
  PROTO=6 216.205.74.243:2123 MY.HOST.211:98
  L=60 S=0x00 I=1981 F=0x4000 T=48 SYN (#11)
```

1. Source: GIAC Web page (URL: http://www.sans.org/y2k/053000-1000.htm)
2. Detect was generated by:
   An IPChains firewall (Linux)
   Fields are as follows:

May 23 04:24:29	[timestamp]
firewall kernel: Packet log:	[logging utility in ipchains]
input	[the ”chain” that matched the packet]
DENY	[rule applied]
eth0	[interface]
PROTO=6	[protocol ID]
193.129.252.129:3163	[source address and port#]
MY.HOST.211:98	[dest address and port#]
L=60	[packet length]
S=0x00	[type of service]
I=32300	[IP ID]
F=0x4000	[frag offset+flags]
T=47	[time to live]
SYN	[TCP flag]
(#11)	[rule #]

3. Probability the source address was spoofed
   Not likely as linuxconf would require conversation – this is an attempt at root access. 1st Address is from UUNET UK, the second from Interliant (ASP and Web hosting company).
4. Description of attack:
   Destination port 98 indicates an attempt to reach linuxconf port. Certain versions of Linux OS had vulnerability (linuxconf was setuid to root in Red Hat 5.1).
5. Attack mechanism:
   Hacker attempts to contact linuxconf application on it’s well-known port. This app can yield information about the configuration of the system, and has features for remote administration.
6. Correlations:
   This detect was submitted by L. Christopher Paul. This is a fairly common attempt, but it would be interesting to see other traces (attacker may know OS of target through prior probes). Note the time separation of 3 days – the two may be totally unrelated.
7. Evidence of active targeting:
   This was an active target in the sense that it is aimed at this machine, but it may simply be a rattle of the doorknob.
8. Severity:
   Criticality = 5 (this is the firewall)
   + Lethality = 5 (root access if successful)

- System = 5 (version not susceptible)
- Network = 5 (traffic denied by FW) = 0

9. Defensive recommendation:
   Existing firewall rules blocked attacks. Might send email to abuse@interliant.com, abuse@uk.uu.net to follow up.
10. Multiple choice test question:
    Is this evidence of
    a) A search for a Trojan
    b) A linuxconf attack
    c) A SYN-FLOOD attack
    d) A wrong number
    Answer: b

Detect 2

```
[**] NMAP TCP ping! [**] 05/28-03:49:09.304761 24.10.224.102:45659 ->
  nnn.nnn.2.2:80 TCP TTL:48 TOS:0x0 ID:64536 ******A* Seq: 0x469848DB
  Ack: 0x0 Win: 0x1000
[**] NMAP TCP ping! [**] 05/28-03:49:10.802258 24.10.224.102:45658 ->
  nnn.nnn.3.2:80 TCP TTL:48 TOS:0x0 ID:63327 ******A* Seq: 0x38D848FB
  Ack: 0x0 Win: 0x1000
[**] FIN Scan [**] 05/28-04:12:42.224742 24.10.224.102:45638 ->
  nnn.nnn.2.2:53
TCP TTL:48 TOS:0x0 ID:34816 ***F**** Seq: 0x0 Ack: 0x0 Win: 0x1000
[**] NMAP TCP ping! [**] 05/28-04:12:42.321690 24.10.224.102:45650 ->
  nnn.nnn.2.2:42838 TCP TTL:48 TOS:0x0 ID:44270 ******A* Seq:
0x69A5BDE3
  Ack: 0x0 Win: 0x1000 TCP Options => WS: 10 NOP MSS: 265 TS:
  1061109567 0 EOL EOL
[**] XMAS Scan [**] 05/28-04:12:42.329244 24.10.224.102:45651 ->
  nn.nnn.2.2:42838
TCP TTL:48 TOS:0x0 ID:3795 ***F*P*U Seq: 0x69A5BDE3 Ack: 0x0 Win:
0x1000
TCP Options => WS: 10 NOP MSS: 265 TS: 1061109567 0 EOL EOL
```

1. Source: GIAC Web page (URL: http://www.sans.org/y2k/053100-1100.htm)
2. Detect was generated by:

SNORT, an intrusion detection package (www.snort.org). This is actually an alert generated in response to a rule. Fields are as follows:

```
[**] NMAP TCP ping! [**]        [Alert text (from the rule)]
05/28-03:49:09.304761           [Time Stamp]
24.10.224.102:45659 ->          [Source IP]
nnn.nnn.2.2:80                  [Dest IP]
TCP                             [Protocol]
TTL:48                          [Time to live]
TOS:0x0                         [Type of service]
ID:64536                        [IP ID]
******A*                        [Flag bits]
Seq: 0x469848DB                 [Sequence #]
Ack: 0x0                        [Header Ack #]
Win: 0x1000                     [Window size]
```

3. Probability the source address was spoofed

The rule that detected this trace was looking for an NMAP signature (NMAP is a security evaluation and network mapping tool). NMAP allows spoofing of source address quite easily. However, the ID in question is registered to ATT at home. Traceroute of this address ended at:
19 140 ms 141 ms 141 ms cj42229-a.alex1.va.home.com [24.10.224.102]
certainly a valid address. Also, with this type of attack, (recon) the source needs to collect data, and so must receive the replies. Not likely a spoofed address.

4. Description of attack:

OS Fingerprinting. Essentially reconnaissance, but attacker appears focused on one machine.

5. Attack mechanism:

Look for open ports (80, 53), then send a combination of several crafted packet types, various options and flags set. This is a scan designed to detect the type of OS and TCP stack in use on the target machine. In fact, this type of scan is also trivial with NMAP, which uses characteristic responses to malformed packets of various OS types to guess what the target is running.

6. Correlations:
   Methodology described in Vicki Irwin and Hal Pomeranz's Course on Intrusion Detection and Packet Filtering, page 172.

7. Evidence of active targeting:
   Not much, just a directed OS fingerprinting. No associated port scan (at least, not a detected one). No evidence to the contrary either (that I can see here), such as an associated ping sweep.

8. Severity:
   Criticality = 3 (actually, unknown so I'll go middle)
   + Lethality = 2 (no direct compromise)
   - System = 3 (no return traffic presented)
   - Network = 4 (traffic detected and source identified) = -2
9. Defensive recommendation:
   Blocking source IP. May not be good enough if attacker can easily spoof. Could also complain to ISP (abuse@corp.home.net, or 1-800-872-3595).

10. Multiple choice test question:
    This is an example of
    a) Portscanning
    b) OS Fingerprinting
    c) A DoS attack from a spoofed IP address
    d) A zone transfer
    Answer: b

Detect 3

```
195.76.27.44 > MY.NET.1.1
23:00:08.268287 195.76.27.44.65535 > MY.NET.1.1.53:
  S 2047410176:2047410176(0) win 512 (ttl 241, id 31241)
23:00:08.271916 195.76.27.44.65535 > MY.NET.1.2.53:
  S 2047410176:2047410176(0) win 512 (ttl 241, id 31241)
23:00:08.294812 195.76.27.44.65535 > MY.NET.1.3.53:
  S 2047410176:2047410176(0) win 512 (ttl 241, id 31241)
23:00:08.317808 195.76.27.44.65535 > MY.NET.1.4.53:
  S 2047410176:2047410176(0) win 512 (ttl 241, id 31241)
23:00:08.330230 195.76.27.44.65535 > MY.NET.1.5.53:
  S 2047410176:2047410176(0) win 512 (ttl 241, id 31241)
23:21:38.484746 195.76.27.44.65535 > MY.NET.254.246.53:
  S 2047410176:2047410176(0) win 512 (ttl 241, id 31241)
23:21:38.497681 195.76.27.44.65535 > MY.NET.254.247.53:
  S 2047410176:2047410176(0) win 512 (ttl 241, id 31241)
23:21:38.510206 195.76.27.44.65535 > MY.NET.254.248.53:
  S 2047410176:2047410176(0) win 512 (ttl 241, id 31241)
23:21:38.531702 195.76.27.44.65535 > MY.NET.254.249.53:
  S 2047410176:2047410176(0) win 512 (ttl 241, id 31241)
23:21:38.548972 195.76.27.44.65535 > MY.NET.254.250.53:
  S 2047410176:2047410176(0) win 512 (ttl 241, id 31241)
```

1. Source: GIAC Web page (URL: http://www.sans.org/y2k/060300.htm)

2. Detect was generated by:
TCPDump – A packet capture and decoding utility for Unix (or possibly WINDump – similar program for MS Windows OS). This output format results from TCP packet decodes. Fields are as follows:

```
23:21:38.531702                 [timestamp]
195.76.27.44.65535 >            [source IP address]
MY.NET.254.249.53:              [dest IP address]
S                               [flags (S=SYN in this case)]
2047410176:2047410176(0)        [seq num
win 512                         [window size]
(ttl 241,                       [time to live]
id 31241)                       [IP ID]
```

3. Probability the source address was spoofed
Low, since the attacker will be listening for the response. To be useful, the information the response provides must be gathered. The only way this can be done is to capture the response (i.e. not spoof the source address, or sniff the packets on the local net). Address resolves to ibernet.es, a domain registered in Barcelona, Spain.

4. Description of attack:
Attacker is scanning for DNS servers.

5. Attack mechanism:
The method used is simply to issue a SYN packet from a source port >1023 (65535 in this case, the highest possible port) to port 53, and then to listen for a SYN ACK packet in response. The target will respond if it is a DNS server, since port 53 can't be blocked without blocking all external name resolution.

6. Correlations:

Common scan. Poster (David Hoelzer) notes it is a fast version. Could be a precursor to an attack such as CVE-1999-0010 (DoS), or CVE-1999-0299 (buffer overflow). Scan method covered in Hal Pomeranz's course at SANS 2000 San Jose, TCP/IP for Intrusion Detection and Perimeter Defense.

7. Evidence of active targeting:
   Not much. Looks like the attacker is scanning a lot of likely host numbers, not zeroed in on one (yet).

8. Severity:
   Criticality = 4 (DNS is important)
   + Lethality = 2 (no direct compromise)
   - System = 4 (no response presented)
   - Network = 4 (traffic detected and source identified) = -2

9. Defensive recommendation:
   Ensure that DNS servers are protected from known vulnerabilities (see http://cve.mitre.org for references). Consider splitting DNS (internal separate from external) to avoid disclosure of internal network resources should external DNS be compromised.
   Consider sending email to domain registrant (hostmaster@ibernet.es) to complain.
10. Multiple choice test question:
   The above trace can be distinguished from normal traffic because:
   a. The TCP sequence numbers do not change
   b. The destination address for each packet is monotonically increasing
   c. The destination port is 53, protocol TCP, normally only used for rare large DNS queries or zone transfers
   d. All of the above.

   Answer: d.

Detect 4

```
Jun 1 12:54:14 dns1 snort[248951]: IDS007 - MISC-Source
  Port Traffic 53 TCP: 210.93.97.149:53 -> 198.82.247.34:111
Jun 1 12:54:14 dns3 snort[9890]: IDS007 - MISC-Source
  Port Traffic 53 TCP: 210.93.97.149:53 -> 198.82.247.98:111
Jun 1 12:54:15 dns3 snort[9890]: RPC Info Query:
  210.93.97.149:930 -> 198.82.247.98:111
```

1. Source: GIAC Web page (URL: http://www.sans.org/y2k/060300.htm)

2. Detect was generated by:
SNORT, an intrusion detection package (www.snort.org). This is apparently a log entry generated in response to a rule. Fields are as follows:

```
Jun 1 12:54:14                                  [Time Stamp]
dns3 snort[9890]:
IDS007 - MISC-Source Port Traffic 53 TCP:       [Rule identifier message]
210.93.97.149:53 ->                             [Source IP + port]
198.82.247.98:111                               [Dest IP + port]
```

3. Probability the source address was spoofed
Low, since the attacker will be listening for the response. Attacker is also afforded some degree of anonymity based on source: address range resolves to the Korean National University of Education. Universities often have high numbers of users, with high turnover, open access, and some do not keep detailed records.

4. Description of attack:
Attacker is scanning for portmapper (dest. Port 111).

5. Attack mechanism:
Attacker uses source port 53, to get past firewalls. Port 53 can't be blocked without blocking legitimate DNS responses. Once a system that responds to port 111 (portmap) is found, the attacker issues an RPC query to that system. Many version of portmap do not provide security, and will happily inform **any** client who queries them of the RPC services running and their port numbers. From there, there are several avenues of attack.

6. Correlations:
CVE: CVE-1999-0168, or possibly a variation on the CVE candidate: CAN-1999-0195). RPC Attacks include: CVE-1999-0003 (a buffer overflow allowing execution of commands as root), CVE-1999-0208, an exploit of NIS allowing arbitrary command execution, and CVE-1999-0212, which allows the attacker to determine filenames. See also postings about intermittent portmapper scans on GIAC:
http://www.sans.org/y2k/032800-2000.htm
http://www.sans.org/y2k/061200.htm

7. Evidence of active targeting:
Significant evidence in that the attacker attempted an RPC call to one of the hosts subsequent to TCP connect to port 111. Apparently this is a script attack (close subsequent timestamp means it's unlikely that attacker typed the RPC request), but it seems to be more than passive info gathering.

8. Severity:

Criticality = 3 (unknown, so I'll go middle value)
+ Lethality = 4 (the RPC query indicates some success)
- System = 2 (are we running a secure version of portmap?)
- Network = 3 (we detected, but did we filter?) = 2

9. Defensive recommendation:

Secure portmap. Filter Port 53 to Port 111 traffic. In fact, check if you even need to allow RPC access to portmap over the non-local net. If not, filter for dest port 111. Send email to seongkim@cc.knue.ac.kr to complain.

10. Multiple choice test question:

Why was the above trace significant? Check all that apply.

a. The attacker targeted portmap and RPC
b. The TCP protocol is inherently insecure
c. The dns server may have been compromised
d. The attacker used source port 53

Answer: a. and d.

Detect 5

```
04/23 23:58:24.286949 63.21.146.131.3773 > 10.0.108.21.12345:
S 12401930:12401930(0) win 819 <mss 536,nop,wscale
0,nop,nop,timestamp 0 0,nop,nop,sackOK> [tos 0xb0] (ttl 48, id 13179)

04/23 23:58:24.291482 63.21.146.131.3774 > 10.0.108.21.1243:
S 12401931:12401931(0) win 8192 <mss 536,nop,wscale
0,nop,nop,timestamp 0 0,nop,nop,sackOK> [tos 0x94] (ttl 48, id 13435)

04/23 23:58:24.292103 63.21.146.131.3778 > 10.0.108.21.31337:
S 12401939:12401939(0) win 8192 <mss 536,nop,wscale
0,nop,nop,timestamp 0 0,nop,nop,sackOK> [tos 0x84] (ttl 48, id 13691)

04/23 23:58:24.431566 63.21.146.131.3788 > 10.0.108.21.12346:
S 12401961:12401961(0) win 8192 <mss 536,nop,wscale
0,nop,nop,timestamp 0 0,nop,nop,sackOK> [tos 0x84] (ttl 48, id 14203)

04/23 23:58:24.432815 63.21.146.131.3777 > 10.0.108.21.21554:
S 12401937:12401937(0) win 8192 <mss 536,nop,wscale
0,nop,nop,timestamp 0 0,nop,nop,sackOK> [tos 0x48] (ttl 48, id 14459)

04/23 23:58:24.438045 63.21.146.131.3793 > 10.0.108.21.6969:
S 12401978:12401978(0) win 8192 <mss 536,nop,wscale
0,nop,nop,timestamp 0 0,nop,nop,sackOK> [tos 0xb0] (ttl 48, id 14971)
```

1. Source: GIAC Web page (URL: http://www.sans.org/y2k/042900.htm)

2. Detect was generated by: TCPDump
   Fields are as follows:

```
04/23 23:58:24.438045          [timestamp]
63.21.146.131.3793 >           [source IP + port]
10.0.108.21.6969:              [dest IP + port]
S                              [IP Flags (S= SYN)]
12401978:12401978(0)           [Seq #]
win 8192                       [Window size]
<mss 536,nop,wscale            [TCP Options]
0,nop,nop,timestamp 0 0,
nop,nop,sackOK>
[tos 0xb0]                     [Type of service]
(ttl 48,                       [Time to live]
id 14971)                      [IP ID]
```

3. Probability the source address was spoofed
   Not likely, as the attack appears to be a Trojan search. Resolves to Uunet (from Arin Whois). Probably a dial-up user.

4. Description of attack:
   Trojan search.

5. Attack mechanism:
   Attacker attempts to connect to a target host on a range of ports utilized by various Trojan horse programs (e.g. 31337= Back Orofice, or 6969 for GateCrasher and others). If host has been compromised, attacker attempts to access system via the Trojan program.

6. Correlations:

See GIAC web site, e.g. http://www.sans.org/y2k/020400.htm, or a nice little site I found that conveniently lists Trojan ports and disinfection procedures: http://nethog.net/feeds/niteryder/trojans.htm

7. Evidence of active targeting:

All traces are to the same host, but very rapid. Looks like the attacker is simply running a trolling script.

8. Severity:

Criticality = 3 (unknown, so I'll go middle value)
+ Lethality = 5 (many of these trojans give the attacker full control)
- System = 4 (no response detected)
- Network = 5 (caught the scan) = -1

9. Defensive recommendation:

Filter traffic to known Trojan ports. Complain to help@UUNET.UU.NET or +1 (800) 900-0241.

10. Multiple choice test question:

The above trace differs from normal traffic because :

a. The source ports are all different for each packet
b. The destination ports are those of known trojans
c. The source sends packets with options set
d. The source sends packets with SYN flag set, but never just ACK
e. b. and d.

Answer e.

Detect 6

```
Feb 1 14:47:20.951726 b 208.184.216.218,4771 ->
10.11.6.99,6699 PR tcp len 20 60 -S
Feb 1 14:47:21.266282 b 208.184.216.218,4773 ->
10.11.6.99,6700 PR tcp len 20 60 -S
Feb 1 14:47:21.663396 b 208.184.216.218,4777 ->
10.11.6.99,21 PR tcp len 20 60 -S
Feb 1 14:47:21.805895 b 208.184.216.218,4778 ->
10.11.6.99,80 PR tcp len 20 60 -S
Feb 1 14:47:21.867111 b 208.184.216.218,4779 ->
10.11.6.99,23 PR tcp len 20 60 -S
```

1. Source: GIAC Web page (URL: http://www.sans.org/y2k/020400.htm)

2. Detect was generated by: A packet filter similar to TCPdump. I'm not familiar with this exact version/format, but it seems pretty clear:
   Fields are as follows:

```
Feb 1 14:47:21.867111          [timestamp]
 b                             [?]
208.184.216.218,4779 ->        [source IP, port]
10.11.6.99,23                  [dest IP, port]
PR                             [flags (push, reset)]
tcp len 20 60 -S               [tcp header and packet lengths?]
```

3. Probability the source address was spoofed
   This address resolves to above.net, an ISP. This is likely a dial-up user.

4. Description of attack:
   Actually, this detect is a false positive. The port 6699 is well known as the Napster application port.

5. Attack mechanism:
   Napster client application allows an external user to connect on port 6699. That user proceeds to download music files (.mp3).
   Not really an attack, but should probably be blocked if organization does not condone recreational surfing.

6. Correlations:
   Universities report up to 5% of network bandwidth consumed by Napster traffic (see http://betanews.efront.com/article.php3?sid=950242623).
   This type of traffic is discussed in SANS presentation: "Intrusion Detection and Packet Filtering: How it Really works", by Vicki Irwin and Hal Pomeranz, page 181.

7. Evidence of active targeting:
   Host was actively targeted, presumably with the full permission of the user!

8. Severity:
   Criticality = 0 (N/A)
   + Lethality = 0 (N/A)
   - System = 0 (N/A)
   - Network = 5 (Caught the traffic as a portscan) = -5

9. Defensive recommendation:

Write a policy that defines acceptable use of network bandwidth. Enforce it if you have one. If all else fails, consider blocking port 6699, and checking for others in that range (typically 6000-8000 for this type of traffic.

10. Multiple choice test question:

This trace is an example of:

a. A wrong number
b. FTP traffic
c. A cgi-bin attack
d. Demon.net traffic
e. None of the above

Answer: e.

Detect 7

```
[**] SCAN-SYN FIN [**]
  06/08-10:49:13.592657 61.11.233.25:53 -> a.b.e.14:53
  TCP TTL:26 TOS:0x0 ID:39426
  **SF**** Seq: 0x3C6EBD59 Ack: 0x7D447418 Win: 0x404
  00 00 00 00 00 00 ......
[**] SCAN-SYN FIN [**]
  06/08-10:49:13.627402 61.11.233.25:53 -> a.b.e.16:53
  TCP TTL:26 TOS:0x0 ID:39426
  **SF**** Seq: 0x3C6EBD59 Ack: 0x7D447418 Win: 0x404
  00 00 00 00 00 00 ......
[**] SCAN-SYN FIN [**]
  06/08-10:49:14.829265 61.11.233.25:53 -> a.b.e.66:53
  TCP TTL:26 TOS:0x0 ID:39426
  **SF**** Seq: 0x2A4EF0D8 Ack: 0x322AD326 Win: 0x404
  00 00 00 00 00 00 ......
[**] SCAN-SYN FIN [**]
  06/08-10:49:14.938314 61.11.233.25:53 -> a.b.e.79:53
  TCP TTL:26 TOS:0x0 ID:39426
  **SF**** Seq: 0x2A4EF0D8 Ack: 0x322AD326 Win: 0x404
  00 00 00 00 00 00 ......
```

1. Source: GIAC Web page (URL: http://www.sans.org/y2k/061200.htm)

2. Detect was generated by: SNORT, an intrusion detection and filtering system.
   Fields are as follows:

```
[**] SCAN-SYN FIN [**]          [Alert text (from the rule)]
06/08-10:49:14.938314           [Time Stamp]
61.11.233.25:53 ->              [Source IP]
a.b.e.79:53                     [Dest IP]
TCP                             [Protocol]
TTL:26                          [Time to live]
TOS:0x0                         [Type of service]
ID:39426                        [IP ID]
**SF****                        [Flag bits]
Seq: 0x2A4EF0D8                 [Sequence #]
Ack: 0x322AD326                 [Header Ack #]
Win: 0x404                      [Window size]
00 00 00 00 00 00 ......      [Start of HEX dump]
```

3. Probability the source address was spoofed
   Low. This is recon, not a DoS attack. Address resolves to thaicom.net, or The Shin Satellite Public Company Limited.

4. Description of attack:
   SYN|FIN attempt on port 53 (DNS).

5. Attack mechanism:
   This is likely an attempt at reconnaissance for DNS servers. The SYN and FIN flags both being set is not a normal traffic condition. It could be used as an attempt to avoid logging, or to pierce packet filters that may look for SYN only.
   The destination address is changing non-monotonically. This indicates a sophisticated scanning program that may randomize target addresses at another attempt at avoiding detection, or be benefiting from previous scans.

6. Correlations:

See similar detects on GIAC: http://www.sans.org/y2k/020600-2000.htm, also http://www.sans.org/y2k/122499.htm, detected in the hacking peak of the holiday season, for an associated "x-mas tree" detect, with lots of irrational flags set.

7. Evidence of active targeting:

This is a recon scan, and fairly sophisticated. Still, it may just be a passive information gathering attempt.

8. Severity:

Criticality = 4 (DNS)
+ Lethality = 3 (no direct compromise, easy to filter)
- System = 3 (don't know without replies to look at)
- Network = 5 (SNORT caught it) = -1

9. Defensive recommendation:

10. Multiple choice test question:

What utility generated the above trace?

a. IPChains Firewall
b. Cisco Netranger
c. SNORT
d. TCPDump

Answer: c.

Detect 8

```
Jun 9 13:07:43 solar portsentry[721]: attackalert:
  SYN/Normal scan from host: CPE-144-132-35-150.vic.bigpond.net.au/
  144.132.35.150 to TCP port: 53
Jun 9 13:07:43 solar portsentry[721]: attackalert:
  Host 144.132.35.150 has been blocked via wrappers with string:
  "ALL: 144.132.35.150"
Jun 9 13:07:44 solar portsentry[721]: attackalert:
  Host 144.132.35.150 has been blocked via dropped route using command:
  "/sbin/ipchains -I input -s 144.132.35.150 -j DENY -l"
```

1. Source: GIAC Web page (URL: http://www.sans.org/y2k/061200.htm)

2. Detect was generated by: Portsentry, a host-based intrusion detection and response system.

Fields are as follows:

```
Jun 9 13:07:43                                  [Timestamp]
solar portsentry[721]:                          [host and process]
attackalert:                                    [alert type]
SYN/Normal scan from host:                      [detect type]
CPE-144-132-35-150.vic.bigpond.net.au/          [attacking host name]
144.132.35.150                                  [attacking IP]
to TCP port: 53                                 [targeted port]
```

3. Probability the source address was spoofed

Low. This was likely a recon scan. Address resolves to Telstra, a large Australian ISP. This was probably a dial-up user, guessing from the host name.

4. Description of attack:

SYN/Normal scan is a typical recon attempt. Port 53 as a target probably means the attacker is looking for DNS servers.

5. Attack mechanism:

The attacker sends a packet with the SYN and NORMAL flags set. Probably an attempt at confusing logging daemons, piercing firewalls, etc.

6. Correlations:

See similar GIAC post from January: http://www.sans.org/y2k/012600.htm, also a ufl.edu post: http://www.health.ufl.edu/wss/mail-archives/unix-sec/1999//05//msg00021.html of a similar detect of a scan from io.cavecreek.net, looking for port 111 (portmapper).

7. Evidence of active targeting:

Host-based intrusion detection systems are inherently limited in that corroborating evidence (scan packets to other hosts) are not picked up. In this case, however, the poster (Pierre Lamy) did not post further evidence of targeting (probably because his portsentry blocked the offending IP address forthwith, thus preventing further exploits). Probably just a passive scan, but hard to say.

8. Severity:

Criticality = 3 (unknown)
\+ Lethality = 2 (scan only)
\- System = 5 (portsentry was quite effective)

- Network = 0 (no information) = 0

9. Defensive recommendation:

Consider email to mboschma@TELSTRA.COM.AU (ISP admin) or abuse@telstra.com.au to complain. Otherwise, portsentry seems to be doing a fine job.

10. Multiple choice test question:

What was the end result of the action taken by portsentry in the above trace?

a. Firewall rule was changed to block offending IP
b. Offending host name was removed from DNS
c. SYN-Normal packets will no longer be accepted
d. Attacker was scanned in retaliation

Answer: a

Detect 9

[12/24/1999 16:29:35.778 GMT-0700] Port 8080 (tcp) is now disabled for 60 seconds.
[12/24/1999 16:29:36.179 GMT-0700] Connection: webhosting.rmt.ru
(194.67.165.84) on port 3128 (tcp).
[12/24/1999 16:29:36.249 GMT-0700] GET
http://194.67.165.80/cgi-bin/proxy/1?192.168.1.2:3128 HTTP/1.0
Connection: Keep-Alive
Pragma: no-cache
Accept: */*
Accept-Encoding: gzip, deflate
Accept-Language: en
Host: 194.67.165.80
Referer: ref
User-Agent: Mazilla

[12/24/1999 16:29:36.249 GMT-0700] Port 3128 (tcp) is now disabled for 60 seconds.

1. Source: GIAC Web page (URL: http://www.sans.org/y2k/122899-9.htm)

2. Detect was generated by: Nuke Nabber, an Anti-intrusion program for MS Windows.
   Fields are as follows:
   Actually, the fields vary quite a bit. Nuke Nabber writes several different and verbose record types that are largely self-explanatory.

3. Probability the source address was spoofed
   Very low. This is almost certainly activity from the Ring Zero Trojan, so the source host is likely compromised.

4. Description of attack:
   Proxy search Trojan.

5. Attack mechanism:
   The ring zero Trojan infects MS Windows hosts and then scans the Internet for proxy servers (hence the telltale http GET for a cgi-bin/proxy directory). It then reports this data back to www.rusftpsearch.net (no longer active).

6. Correlations:
   See description at http://www.datafellows.com/v-descs/ringzero.htm, and also the SANS 2000 course brief: http://www.sans.org/sans2000jcscbriefcourse.htm, on the Hunt for Ring Zero.

7. Evidence of active targeting:
   None. This Trojan scans, and scans, and scans…

8. Severity:
   Criticality = 3 (middle value)
   + Lethality = 1 (possible proxy abuse)
   - System = 5 (Nuke Nabber caught it)

- Network = 0 (no information) = -1

9. Defensive recommendation:

Advise the Muscovites at noc@radio-msu.net that host webhosting.rmt.ru may be compromised.

10. Multiple choice test question:

If you are advised by a network administrator that (s)he has detected traffic from your site resulting in a trace similar to that listed above, you should:

a. Update your firewall to not allow traffic on port 3128
b. Scan your Windows hosts for Trojans
c. Reboot your firewall to clear Trojan programs from RAM
d. Install Nuke Nabber on all your Windows Systems

Answer: b.

Detect 10

```
07:25:26.637334 10.2.16.76.2846 > 170.129.39.28.161:
|30|81|ea|02|01|04|06|a1|81|dcGetNextRequest(9)
|02|01|02|01|02|01[|snmp] (ttl 110, id 37986)
07:25:26.639967 10.2.16.76.2846 > 170.129.40.28.161:
|30|81|ea|02|01|04|06|a1|81|dcGetNextRequest(9)
|02|01|02|01|02|01[|snmp] (ttl 110, id 38242)
07:25:26.642126 10.2.16.76.2846 > 170.129.41.28.161:
|30|81|ea|02|01|04|06|a1|81|dcGetNextRequest(9)
|02|01|02|01|02|01[|snmp] (ttl 110, id 38498)
07:25:26.644821 10.2.16.76.2846 > 170.129.42.28.161:
|30|81|ea|02|01|04|06|a1|81|dcGetNextRequest(9)
|02|01|02|01|02|01[|snmp] (ttl 110, id 38754)
07:25:26.647449 10.2.16.76.2846 > 170.129.43.28.161:
|30|81|ea|02|01|04|06|a1|81|dcGetNextRequest(9)
|02|01|02|01|02|01[|snmp] (ttl 110, id 39010)
07:25:26.649627 10.2.16.76.2846 > 170.129.49.28.161:
|30|81|ea|02|01|04|06|a1|81|dcGetNextRequest(9)
|02|01|02|01|02|01[|snmp] (ttl 110, id 40546)
07:25:26.650785 10.2.16.76.2846 > 170.129.44.28.161:
|30|81|ea|02|01|04|06|a1|81|dcGetNextRequest(9)
|02|01|02|01|02|01[|snmp] (ttl 110, id 39266)
07:25:26.652868 10.2.16.76.2846 > 170.129.45.28.161:
|30|81|ea|02|01|04|06|a1|81|dcGetNextRequest(9)
|02|01|02|01|02|01[|snmp] (ttl 110, id 39522)
```

1. Source: GIAC Web page (URL: http://www.sans.org/y2k/042600.htm)

2. Detect was generated by: TCPDump (actually Shadow, according to the poster).
   Fields are as follows:

```
07:25:26.652868                    [timestamp]
10.2.16.76.2846 >                  [source ip]
170.129.45.28.161:                 [dest ip]
|30|81|ea|02|01|04|06|a1|81|
dcGetNextRequest(9)                [snmp request type
|02|01|02|01|02|01[|                - tree walking]
snmp]                              [snmp identifier]
(ttl 110,                          [time to live]
 id 39522)                         [IP ID]
```

3. Probability the source address was spoofed

Could have been, but it would be illogical unless this was some kind of misguided DoS attack. More likely this is a misconfiguration, since the source address is a typical RFC1918 private address, and not routable over the public Internet. Couldn't be a scan, since the replies would go nowhere.

4. Description of attack:

Misconfiguration.

5. Attack mechanism:

Incorrectly configured firewall or host is "spilling" these packets onto the Internet. This is unfortunately fairly common – an error was made configuring an Internet interface on an snmp console, for example. What's unique in this case is that the destination address is routable, and so they ended up on the poster's DMZ (external unfiltered network).

6. Correlations:

Poster David Hoelzer describes convincing his ISP that these packets were errors, not a scan, or an attempt at illicit access.
Such an attempt would have a different signature; in that the source would be a routable, non-spoofed address, and that the access attempts would vary in format a bit (to try various query types). For an example of this, see post from Laurie: http://www.sans.org/y2k/031400-1600.htm

7. Evidence of active targeting:
   While the activity is all directed at a single host, it is unlikely that it is targeted at all.

8. Severity:
   Criticality = 3 (don't know what host the destination address was – middle #)
   + Lethality = 2 (hardly, though I can't discount the value of the MIB)
   - System = 3 (never reached the host - don't know, so I'll go middle)
   - Network = 5 (detected and filtered) = -3

9. Defensive recommendation:
   Filter packets at the firewall. Complain to ISP.

   10. Multiple choice test question:

   The above trace, generated by TCPDump, was an example of:
   a. A misconfiguration (wrong number)
   b. A DoS attack
   c. A spoofed source IP
   d. A very fast snmp scan
   Answer: a, but I'd also give partial credit for b.