Global Information Assurance Certification Paper

Practical Exam

GIAC Intrusion Detection Curriculum

San Jose, 2000

Del Armstrong

Detect #1:

DTK LOG

```
attacker.network.61 3554 53 2000/04/15 17:50:35 1171 528:47 listen.pl S0 R-Peace Init

attacker.network.61 3554 53 2000/04/15 17:50:35 1171 528:47 listen.pl S0 RTension-
Tension WeClose

attacker.network.61 53 53 2000/04/15 17:52:43 529 529:7 UDPlisten.pl SUDP RTension-
Tension attacker.network.61(4949):˜»  €zi

attacker.network.61 53 53 2000/04/15 17:52:43 529 529:7 UDPlisten.pl SUDP RTension-
Tension attacker.network.61(4949):s[€versionbind
```

TCPDUMP LOG

```
17:50:32.502957 attacker.network.61.3497 > my.network.28.53: S 1239957587:1239957587(0)
win 321
20 <mss 1460,sackOK,timestamp 19220699[|tcp]> (DF)

17:50:32.505703 attacker.network.61.3498 > my.network.29.53: S 1237717813:1237717813(0)
win 321
20 <mss 1460,sackOK,timestamp 19220699[|tcp]> (DF)

17:50:32.752875 attacker.network.61.3554 > my.network.85.53: S 1239369081:1239369081(0)
win 321
20 <mss 1460,sackOK,timestamp 19220739[|tcp]> (DF)

17:50:32.754315 my.network.85.53 > attacker.network.61.3554: S 2385116072:2385116072(0)
ack 123
9369082 win 30660 <mss 1460,sackOK,timestamp 674804368[|tcp]> (DF)

17:50:32.762689 attacker.network.61.3557 > my.network.88.53: S 1251035788:1251035788(0)
win 321
20 <mss 1460,sackOK,timestamp 19220740[|tcp]> (DF)

17:50:32.812133 attacker.network.61.3571 > my.network.102.53: S 1247052058:1247052058(0)
win 32
120 <mss 1460,sackOK,timestamp 19220744[|tcp]> (DF)

17:50:32.862109 attacker.network.61.3554 > my.network.85.53: . ack 1 win 32120
<nop,nop,timestamp 19220762 674804368> (DF)
```

```
17:50:32.865052 attacker.network.61.3554 > my.network.85.53: F 1:1(0) ack 1 win 32120
<nop,nop,timestamp 19220762 674804368> (DF)

17:50:32.865196 my.network.85.53 > attacker.network.61.3554: . ack 2 win 30660
<nop,nop,timestamp 674804379 19220762> (DF)

17:52:43.876141 attacker.network.61.4949 > my.network.85.53: 39099 inv_q+ [b2&3=0x980]
A? . (27)

17:52:43.882087 my.network.85.53 > attacker.network.61.4949: 25455 zoneInit
[b2&3=0x7265] [30061a] [8292q] [28773n] [25600au] (11)

17:52:43.961115 attacker.network.61.4949 > my.network.85.53: 29531+ [b2&3=0x180] (30)

17:52:43.965743 my.network.85.53 > attacker.network.61.4949: 25455 zoneInit
[b2&3=0x7265] [30061a] [8292q] [28773n] [25600au] (11)
```

Source of trace:

The trace was obtained on the office LAN of a large corporation. The LAN is directly routed to the Internet with no intervening firewall.

Detect was generated by:

DTK (Deception Toolkit – www.all.net) and tcpdump.

The tcpdump logs are in the standard, non-verbose format, wrapped at word boundaries.

The DTK logs show the source IP, the source and destination port, time information, and then info about the DTK program running (PID of listener, PID of handler, name of handler, perceived threat level), the action taken and sometimes the data sent by the intruder.

Probability of spoofed source address:

The probability that this traffic features spoofed source addresses is low. This reconnaissance requires receiving data back via a valid TCP session, which generally is infeasible with a spoofed source address.

Description of attack:

This is a reconnaissance looking for DNS servers at a specific version level. It is almost certainly looking for DNS servers vulnerable to a currently popular buffer overflow attack.

The attack consists of two parts, a simple TCP SYN scan looking for machine accepting connections on port 53 (DNS). After the TCP syn scan is completed for the target network, UDP traffic is sent to the identified DNS servers, attempting to request the version of the running DNS server.

Attack Mechanism:

The attack works by sending a SYN packet to port 53 of the machines being scanned. If the machine responds with a SYN-ACK, the attacker completes the three-way-handshake, and then tears down the connection via a FIN.

After the scan has located any DNS servers, the DNS servers are sent UDP traffic to port 53, requesting the version of the DNS server.

It is a fair assumption that if the version returned corresponds to a vulnerability known to the attacker, the attacker will proceed to attack the DNS server. DNS compromises can lead to root compromises on the host box, or using DNS to subvert other boxes.

Correlations:

This attack was described in the GIAC Intrusion Detection Course at San Jose 2000.

Cert Advisory CA-99-14, "Multiple Vulnerabilities in BIND" (www.cert.org/advisories/CA-99-14-bind.html).

Cert Advisory CA-2000-3, "Continuing Compromises of DNS Servers" (http://www.cert.org/advisories/CA-2000-03.html).

Evidence of Active Targeting:

This was a scan of an entire class-C subnet. It was clearly hostile, but there's no evidence that specific machines were targeted prior to the beginning of the scan.

Severity:

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

Severity = (1 [honeypot]) + 5 [potential root compromise]) –
(5 [hardened honeypot system] + 1 [no firewall, on Internet])

Severity =  0  (not a critical event)
```

Defensive Recommendations:

Ensure that DNS servers are at the proper patch level, and securely configured. If possible, install firewall or router ACL to control external access to DNS servers.

Exam Question Based on this Trace:

This trace shows:

A) Buffer Overflow attack against the timestamp service.
B) Reconnaissance of network looking for vulnerable DNS servers.
C) Port Scan looking for DNS.
D) Web crawler looking for web sites to catalog.

Answer: B

Detect #2

```
3d00h: %SEC-6-IPACCESSLOGP: list 199 denied tcp attacker.network.72(4471) ->
my.network.64(80),+1 packet

3d00h: %SEC-6-IPACCESSLOGP: list 199 denied tcp attacker.network.72(4479) ->
my.network.64(21),+1 packet

3d00h: %SEC-6-IPACCESSLOGP: list 199 denied tcp attacker.network.72(4489) ->
my.network.69(80),+1 packet

3d00h: %SEC-6-IPACCESSLOGP: list 199 denied tcp attacker.network.72(4497) ->
+my.network.69(1080), 1 packet

3d00h: %SEC-6-IPACCESSLOGP: list 199 denied tcp attacker.network.72(4481) ->
+my.network.65(8080), 1 packet

3d00h: %SEC-6-IPACCESSLOGP: list 199 denied tcp attacker.network.72(4515) ->
my.network.71(81),+1 packet

3d00h: %SEC-6-IPACCESSLOGP: list 199 denied tcp attacker.network.72(4514) ->
+my.network.71(10080), 1 packet

3d00h: %SEC-6-IPACCESSLOGP: list 199 denied tcp attacker.network.72(4517) ->
+my.network.71(1080), 1 packet

3d00h: %SEC-6-IPACCESSLOGP: list 199 denied tcp attacker.network.72(4471) ->
my.network.64(80),+1 packet

3d00h: %SEC-6-IPACCESSLOGP: list 199 denied tcp attacker.network.72(4484) ->
+my.network.65(10080), 1 packet

3d00h: %SEC-6-IPACCESSLOGP: list 199 denied tcp attacker.network.72(4514) ->
+my.network.71(10080), 1 packet

3d00h: %SEC-6-IPACCESSLOGP: list 199 denied tcp attacker.network.72(4517) ->
+my.network.71(1080), 1 packet

3d06h: %SEC-6-IPACCESSLOGP: list 199 denied tcp attacker2.network.15(3612) ->
my.network.64(21),+1 packet

3d06h: %SEC-6-IPACCESSLOGP: list 199 denied tcp attacker2.network.15(3612) ->
my.network.64(21),+1 packet

3d06h: %SEC-6-IPACCESSLOGP: list 199 denied tcp attacker3.network.75(1520) ->
+my.network.64(1243), 1 packet

3d06h: %SEC-6-IPACCESSLOGP: list 199 denied tcp attacker3.network.75(1808) ->
+my.network.64(1243), 1 packet

3d06h: %SEC-6-IPACCESSLOGP: list 199 denied tcp attacker3.network.75(1522) ->
+my.network.66(1243), 1 packet

3d06h: %SEC-6-IPACCESSLOGP: list 199 denied tcp attacker3.network.75(1808) ->
+my.network.64(1243), 1 packet
```

Source of Trace:

Cisco router connected to Internet via ISDN

Detect was Generated by:

Cisco Router logging packets denied due to ACL.

On Cisco logs, the first field is the uptime of the router. Following that are the type of log message, the access list which generated the message, the action taken against the packet, the protocol of the packet, source IP and port, destination IP and port, and the number of packets acted on.

Probability of Spoofed Source Address:

It is unlikely that the traffic is spoofed.

These were reconnaissances of the network using TCP. If the source address was spoofed, the attacker would have difficulty getting the response. Since all the traffic is from one address, it is unlikely that this is a "decoy" scan, since there is no address to hide amongst the decoys.

Note: With some types of reconnaissance, it is possible for an attacker to use traffic spoofed to be from an innocent host - if that host is on a broadcast network being surreptitiously monitored by the attacker.

Under some rare circumstances, it's also possible to do some types of network reconnaissance by sending traffic spoofed to be from a third party (and then monitoring changes in the IP identification numbers used by the third party). The tool hping can be used to conduct these sorts of scans.

Description of Attack:

These were multiple scans, looking for multiple services on multiple machines on the network. The services were targeted, and included common services such as http, as well as common Trojan servers such as SubSeven. Since only a few machines were scanned, the machines were probably targeted as well, indicating previous knowledge of the network.

The scans all occurred within 6 hours of each other, on a network which has seen very little hostile activity. While this may be a coincidence, there is a significant chance that this was a coordinated attack from several sites.

Interestingly, the scan also checked for the Amanda backup server port (10080). I'm unaware of any remote exploits against Amanda, and that port does not show up on any lists of Trojan ports I have. Perhaps it's a new Trojan.

Attack Mechanism:

This attack works by attempting to initiate a TCP three-way-handshake. Depending on the response, the attacker may be able to determine whether there is a device at a specific IP address, and whether it will accept connections to specific ports. In this case, the router intercepted the attempts, preventing the attacker from receiving any results.

If the reconnaissance had been successful, the attacker would have been able to make targeted attacks against known services on specific machines.

Correlations:

Many of the services being probed for are known to have attacks.

For example see CVE-1999-0067 (www.cve.mitre.org/).

The SubSeven trojan is well known, similar in capabilities to BackOriface and netbus.

Scanning for wingate servers is also very common, see: VN-98.03

For a discussion of security issues with FTP, see page 487 of Practical UNIX & Internet Security, 2nd Edition.

Evidence of Active Targeting:

There is some evidence to suggest that specific hosts were targeted. Only a portion of the address space available was scanned. Clearly, only certain services were targeted.

Severity:

```
Severity = (Criticality + Lethality) - (System + Countermeasures)

Severity = (1 [home, family machines] + 5 [trojan remote admin]) -
 (5 [well administered] + 4 [router with good ACLs]) = -4
```

Defensive Recommendation:

No change required. The router ACL defeated the attack.

Multiple Choice Question:

The above security notifications are from:

A) The intruder alert subsystem of the VMS Security Monitor.
B) Tcpdump running in IDS mode.
C) The Linux ip-chains firewall subsystem.
D) A router logging traffic denied by an ACL.

Answer: D

Detect #3:

```
-*> Snort! <*-
Version 1.6
By Martin Roesch (roesch@clark.net, www.clark.net/~roesch)
05/31-14:06:15.814111 attacker.network.74:1376 -> my.network.11:98
TCP TTL:51 TOS:0x0 ID:65463  DF
**S***** Seq: 0x10D1EB9E   Ack: 0x0   Win: 0x7D78
```

```
TCP Options => MSS: 1460 SackOK TS: 76406270 0 NOP WS: 0

05/31-14:06:15.815103 attacker.network.74:1381 -> my.network.12:98
TCP TTL:51 TOS:0x0 ID:65467  DF
**S***** Seq: 0x1022AA63   Ack: 0x0   Win: 0x7D78
TCP Options => MSS: 1460 SackOK TS: 76406271 0 NOP WS: 0

05/31-14:06:15.914865 attacker.network.74:1558 -> my.network.23:98
TCP TTL:51 TOS:0x0 ID:65535  DF
**S***** Seq: 0x100FF017   Ack: 0x0   Win: 0x7D78
TCP Options => MSS: 1460 SackOK TS: 76406283 0 NOP WS: 0

[Additional traffic deleted for brevity]

06/01-01:34:12.211245 attacker.network.74:3653 -> my.network.23:98
TCP TTL:51 TOS:0x0 ID:20352  DF
**S***** Seq: 0x35884051   Ack: 0x0   Win: 0x7D78
TCP Options => MSS: 1460 SackOK TS: 80533549 0 NOP WS: 0

06/01-01:34:12.211405 attacker.network.74:3659 -> my.network.26:98
TCP TTL:51 TOS:0x0 ID:20358  DF
**S***** Seq: 0x35916A01   Ack: 0x0   Win: 0x7D78
TCP Options => MSS: 1460 SackOK TS: 80533549 0 NOP WS: 0

[Rest deleted for brevity]
```

Source of trace:

The trace was obtained on the office LAN of a large corporation. The LAN is directly routed to the Internet with no intervening firewall.

Detect was generated by:

Tcpdump.

The tcpdump data is being displayed via snort, in its standard, non-verbose format.

Probability of spoofed source address:

The probability that this traffic features spoofed source addresses is low. This reconnaissance requires receiving data back via a TCP response, which generally is infeasible with a spoofed source address.

There is no evidence that spoofed decoy packets are present.

Note: With some types of reconnaissance, it is possible for an attacker to use traffic spoofed to be from an innocent host - if that host is on a broadcast network being surreptitiously monitored by the attacker.

Under some rare circumstances, it's also possible to do some types of network reconnaissance by sending traffic spoofed to be from a third party (and then monitoring changes in the IP identification numbers used by the third party). The tool hping can be used to conduct these sorts of scans.

Description of attack:

This is a reconnaissance looking for linuxconf. It is almost certainly looking for linuxconf servers vulnerable to a currently popular buffer overflow attack.

If a responsive server is found, presumably it will be attacked with a buffer overflow expoit.

Interestingly, about ten hours after the scan, a second, duplicate scan was run. It's likely that the attacker was checking to see if new machines had appeared in the network, or perhaps he/she forgot they had already scanned this network.

Attack Mechanism:

The attack works by sending a SYN packet to port 98 of the machines being scanned.

If the target responds with a SYN-ACK, the attacker knows that the machine responding is potentially vulnerable.

The target will respond with a TCP RESET if the port is not open.

Correlations:

CVE Catalog:CAN-2000-0017 (under review)

For a discussion of RESET being sent, see page 247 of TCP/IP Illustrated, Volume 1: The Protocols.

Evidence of Active Targeting:

This was a scan of an entire class-C subnet. It was clearly hostile, but there's no evidence that specific machines were targeted prior to the beginning of the scan.

Severity:

```
Severity = (Criticality + Lethality) - (System + Countermeasures)

Severity = (2 [office machines]) + 5 [potential root compromise]) -
(5 [Not running service] + 1 [no firewall, on Internet])

Severity =  1  (not a critical event)
```

Defensive Recommendations:

Ensure linuxconf is not being run, or is patched. If possible, install firewall or router ACL to control external access to linuxconf servers.

Exam Question Based on this Trace:

This trace shows:

A) Buffer Overflow attack against the SackOK service.
B) Traceroute reconnaissance.
C) Loki backdoor session.
D) Reconnaissance of network looking for vulnerable linuxconf servers.

Answer: D

Detect #4:

```
FWIN,2000/06/10,15:21:40 -5:00 GMT,attacker.network.9:80,my.network.172:6148,TCP
```

Source of trace:

The trace was obtained on the office LAN of a large corporation. The LAN is directly routed to the Internet with no intervening firewall.

Detect was generated by:

ZoneAlarm, A "personal firewall" available from ZoneLabs.com

The Zone Alarm logs consist of an indication of the type of block (FWIN: Firewall Incoming traffic), followed by the date followed by the source IP and port, and the destination IP and port, and which IP protocol.

Zone Alarm identified this packet as an unsolicited probe.

Probability of spoofed source address:

The probability that this traffic features spoofed source addresses is low. This traffic is a portion of HTTP traffic, which uses TCP. At the time this alarm was raised, legitimate HTTP traffic was being exchanged with the source of the flagged packet.

Description of attack:

This is a false alarm. At the same time that HTTP traffic was being exchanged with the source (a web server), an alarm was raised about traffic from port 80 on the source to a high numbered port on the local machine.

It is common for the load-balancing techniques used by some web sites to confuse personal firewalls.

Attack Mechanism:

The load balancing software used by high end web servers can occasionally send packets which confuse personal firewalls.

Correlations:

This problem has been reproduced under controlled conditions locally. It has also been reported to us numerous times in the form of complaints about downstream web sites.

This problem is also referred to in a review of ZoneAlarm: Erroneous "Blocking" pop-ups

Evidence of Active Targeting:

There is no evidence of active targeting.

Severity:

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

Severity = (2 [Windows machine]) + 0 [no danger]) –
(5 [hardened office systems] + 1 [no firewall, on Internet])

Severity =  -4  (not a critical event)
```

Defensive Recommendations:

No increased defensive efforts are required. Continue using Zone Alarm (it's a good product), but be aware of potential for occasional false alarms

Exam Question Based on this Trace:

This trace shows:

A) Floppy Wind-up function exploit.
B) Buffer overflow attack against a web server.
C) Port Scan looking for TCP server.
D) Common false alarm due to web traffic.

Answer: D

Detect #5:

PORTSENTRY DETECT LOG

```
960438339 - 06/08/100 00:25:39 Host: attacker.network.com/attacker.network.241 Port: 111
Blocked
```

TCPDUMP LOG OF ATTACK

```
00:25:39.151190 attacker.network.241.3542 > my.network.219.111: S
1253046084:1253046084(0) win 32120 <mss 1460,sackOK,timestamp 166886800 0,nop,wscale 0>
(DF)

00:25:39.152274 my.network.219.111 > attacker.network.241.3542: S 790292650:790292650(0)
ack 1253046085 win 32120 <mss 1460,sackOK,timestamp 656493007 166886800,nop,wscale 0>
(DF)

00:25:39.339429 attacker.network.241.3542 > my.network.219.111: . ack 1 win 32120
<nop,nop,timestamp 166886833 656493007> (DF)

00:25:39.340857 my.network.219.111 > attacker.network.241.3542: F 1:1(0) ack 1 win 32120
<nop,nop,timestamp 656493026 166886833> (DF)
```

```
00:25:40.038402 attacker.network.241.3542 > my.network.219.111: . ack 2 win 32120
<nop,nop,timestamp 166886874 656493026> (DF)

00:25:42.954406 attacker.network.241.3692 > my.network.219.111: S
1255088016:1255088016(0) win 32120 <mss 1460,sackOK,timestamp 166887160 0,nop,wscale 0>
(DF)

00:25:42.954630 my.network.219.111 > attacker.network.241.3692: S 804223245:804223245(0)
ack 1255088017 win 32120 <mss 1460,sackOK,timestamp 656493387 166887160,nop,wscale 0>
(DF)

00:25:44.048544 attacker.network.241.3692 > my.network.219.111: P 1:45(44) ack 1 win
32120 <nop,nop,timestamp 166887279 656493387> (DF)

00:25:44.048825 my.network.219.111 > attacker.network.241.3692: . ack 45 win 32120
<nop,nop,timestamp 656493497 166887279> (DF)

00:25:44.050033 my.network.219.111 > attacker.network.241.3692: R 1:1(0) ack 45 win
32120 <nop,nop,timestamp 656493497 166887279> (DF)

00:25:44.484256 attacker.network.241.3542 > my.network.219.111: F 1:1(0) ack 2 win 32120
<nop,nop,timestamp 166887337 656493026> (DF)

00:25:44.484469 my.network.219.111 > attacker.network.241.3542: . ack 2 win 32120
<nop,nop,timestamp 656493540 166887337> (DF)

00:27:11.324425 attacker.network.241.2302 > 24.24.56.119.111: S 1338178571:1338178571(0)
win 32120 <mss 1460,sackOK,timestamp 166895996 0,nop,wscale 0> (DF)

00:27:13.970879 attacker.network.241.2302 > 24.24.56.119.111: S 1338178571:1338178571(0)
win 32120 <mss 1460,sackOK,timestamp 166896296 0,nop,wscale 0> (DF)
```

CONTENTS OF PACKET #8 (Ethereal output)

```
Frame 8 (110 on wire, 110 captured)
    Arrival Time: Jun  8, 2000 00:25:44.0485
    Time delta from previous packet: 1.093914 seconds
    Frame Number: 8
    Packet Length: 110 bytes
    Capture Length: 110 bytes
Ethernet II
    Destination: 00:a0:24:aa:bb:ba (3Com_aa:bb:ba)
    Source: 08:00:3e:16:fd:7e (Motorola_16:fd:7e)
    Type: IP (0x0800)
Internet Protocol
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..00 = Currently Unused: 0
    Total Length: 96
    Identification: 0x2d9e
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 46
    Protocol: TCP (0x06)
    Header checksum: 0xfc80 (correct)
    Source: attacker.net.com (attacker.network.241)
```

```
    Destination: my.network.com (my.network.219)
Transmission Control Protocol, Src Port: 3692 (3692), Dst Port: sunrpc (111), Seq:
1255088017, Ack: 804223246
    Source port: 3692 (3692)
    Destination port: sunrpc (111)
    Sequence number: 1255088017
    Acknowledgement number: 804223246
    Header length: 32 bytes
    Flags: 0x0018 (PSH, ACK)
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 1... = Push: Set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 32120
    Checksum: 0xba01
    Options: (12 bytes)
        NOP
        NOP
        Time stamp: tsval 166887279, tsecr 656493387
Remote Procedure Call
    Last Fragment: Yes
    Fragment Length: 40
    XID: 0x658e922e (1703842350)
    Message Type: Call (0)
    RPC Version: 2
    Program: PORTMAP (100000)
    Program Version: 2
    Procedure: DUMP (4)
    Credentials
        Flavor: AUTH_NULL (0)
        Length: 0
    Verifier
        Flavor: AUTH_NULL (0)
        Length: 0
Portmap
    Program Version: 2
    Procedure: DUMP (4)
```

Source of trace:

The trace was obtained from a firewall connected to a cable modem.

Detect was generated by:

Initial detect was by portsentry. Follow-up investigation was based on data obtained via tcpdump. Ethereal was also used in the analysis.

The tcpdump logs are in the standard, non-verbose format, wrapped at word boundaries.

The ethereal log is in the verbose file output mode, showing the full application and lower layers.

Probability of spoofed source address:

The probability that this traffic features spoofed source addresses is low. This reconnaissance requires receiving data back via a valid

TCP session, which generally is infeasible with a spoofed source address.

Description of attack:

This is a reconnaissance looking for RPC servers. It is almost certainly looking for RPC services vulnerable to a currently popular buffer overflow attack.

Attack Mechanism:

The attack works by sending a SYN packet to port 111 of the machines being scanned. If the machine supports connections to port 111 it responds with a SYN-ACK. If the machine responds with a SYN-ACK, the attacker completes the three-way-handshake, and then closed down the connection via a FIN.

After the scan has located any RPC servers, a new TCP session is initiated to port 111 on each server. Each server is sent the RPC DUMP command to obtain a list of RPC services available.

It is a fair assumption that if the DUMP advertises a service with a vulnerability known to the attacker, the attacker would proceed to attack the vulnerable RPC service. Compromises of RPC services can lead to root compromises on the host box.

Correlations:

This attack was described in the GIAC Intrusion Detection Course at San Jose 2000.

There are numerous exploits against RPC based services, see for example CERT® Advisory CA-99-08 "Buffer Overflow Vulnerability in Calendar Manager Service Daemon, rpc.cmsd".

For a more general discussion of RPC vulnerabilities, see: CERT Incident Note IN-99-04

See also, Common Vulnerabilities and Exposures: CVE-1999-0320

Evidence of Active Targeting:

This appears to be a brute force scan of an entire network. There's no evidence that specific machines were targeted prior to the beginning of the scan.

Severity:

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

Severity = (5 [firewall]) + 5 [potential root compromise]) –
(5 [hardened firewall system] + 5 [firewall])

Severity =  0  (not a critical event)
```

Defensive Recommendations:

Attack was stopped by portsentry running on a firewall, no additional defensive measures are necessary.

Exam Question Based on this Trace:

This trace shows:

A) Probe for RPC services.
B) Reconnaissance of network looking for vulnerable DNS servers.
C) Port 111 Blocked by a denial-of-service attack.
D) Portmap DUMP backdoor channel.

Answer: A

Detect #6:

```
15:26:15.962952 attacker.network.44.65535 > my.network.40.53: S 2252144640:2252144640(0)
win 512 (ttl 241, id 34365)

15:26:15.963120 attacker.network.44.65535 > my.network.40.53: S 2252144640:2252144640(0)
win 512 (ttl 240, id 34365)

15:26:15.963238 attacker.network.44.65535 > my.network.40.53: S 2252144640:2252144640(0)
win 512 (ttl 239, id 34365)

15:26:15.963521 attacker.network.44.65535 > my.network.40.53: S 2252144640:2252144640(0)
win 512 (ttl 238, id 34365)

15:26:15.963564 attacker.network.44.65535 > my.network.40.53: S 2252144640:2252144640(0)
win 512 (ttl 237, id 34365)

15:26:15.963650 attacker.network.44.65535 > my.network.40.53: S 2252144640:2252144640(0)
win 512 (ttl 236, id 34365)

15:26:15.963729 attacker.network.44.65535 > my.network.40.53: S 2252144640:2252144640(0)
win 512 (ttl 235, id 34365)

15:26:15.963813 attacker.network.44.65535 > my.network.40.53: S 2252144640:2252144640(0)
win 512 (ttl 234, id 34365)

15:26:15.963893 attacker.network.44.65535 > my.network.40.53: S 2252144640:2252144640(0)
win 512 (ttl 233, id 34365)

[approx 208 of these packets later ...]

15:26:15.995845 attacker.network.44.65535 > my.network.41.53: S 2252144640:2252144640(0)
win 512 (ttl 113, id 34365)

15:26:15.995886 attacker.network.44.65535 > my.network.40.53: S 2252144640:2252144640(0)
win 512 (ttl 2, id 34365)

15:26:15.996009 attacker.network.44.65535 > my.network.40.53: S 2252144640:2252144640(0)
win 512 [ttl 1] (id 34365)

15:26:15.996051 attacker.network.44.65535 > my.network.41.53: S 2252144640:2252144640(0)
win 512 (ttl 112, id 34365)
```

```
[And the scan continues ...]

15:26:16.846582 attacker.network.44.65535 > my.network.83.53: S 2252144640:2252144640(0)
win 512 (ttl 241, id 34365)

15:26:16.855227 attacker.network.44.65535 > my.network.84.53: S 2252144640:2252144640(0)
win 512 (ttl 241, id 34365)

15:26:16.867025 attacker.network.44.65535 > my.network.85.53: S 2252144640:2252144640(0)
win 512 (ttl 241, id 34365)

[And so on ...]


14% traceroute -n my.network.40
traceroute to my.network.40 (my.network.40), 30 hops max, 38 byte packets
 1  my.network.253  18.076 ms  0.679 ms  0.385 ms
 2  my.network.16  0.908 ms  0.763 ms  0.679 ms
 3  my.network.253  0.683 ms  0.809 ms  0.520 ms
 4  my.network.16  0.978 ms  0.931 ms  0.881 ms
 5  my.network.253  10.755 ms  0.913 ms  0.803 ms
 6  my.network.16  1.020 ms  1.058 ms  1.092 ms
 7  my.network.253  0.942 ms  0.901 ms  0.966 ms
 8  my.network.16  1.151 ms  1.213 ms  1.247 ms
 9  my.network.253  1.113 ms  1.111 ms  1.046 ms
10  my.network.16  1.352 ms  1.402 ms  18.810 ms
11  my.network.253  1.491 ms  15.654 ms  1.295 ms
12  my.network.16  1.474 ms  1.600 ms  1.560 ms
13  my.network.253  1.446 ms  1.379 ms  1.423 ms
14  my.network.16  1.678 ms  1.719 ms  1.735 ms
15  my.network.253  1.765 ms  1.544 ms  1.720 ms
16  my.network.16  1.860 ms  2.631 ms  14.170 ms
17  my.network.253  1.983 ms  1.889 ms  1.717 ms
18  my.network.16  2.094 ms  2.036 ms  2.152 ms
19  my.network.253  14.367 ms  2.048 ms  1.874 ms
20  my.network.16  2.182 ms  2.612 ms  11.971 ms
21  my.network.253  2.174 ms  2.023 ms  3.078 ms
22  my.network.16  11.710 ms  2.375 ms  2.386 ms
23  my.network.253  2.281 ms  5.121 ms  6.774 ms
24  my.network.16  2.563 ms  2.463 ms  2.568 ms
25  my.network.253  2.419 ms  2.376 ms  2.403 ms
26  my.network.16  2.693 ms  2.667 ms  3.199 ms
27  my.network.253  2.610 ms  2.584 ms  2.482 ms
28  my.network.16  2.830 ms  2.881 ms  2.894 ms
29  my.network.253  2.752 ms  2.720 ms  2.724 ms
30  my.network.16  3.000 ms  3.035 ms  3.065 ms
```

Source of trace:

The trace was obtained on the office LAN of a large corporation. The LAN is directly routed to the Internet with no intervening firewall.

Detect was generated by:

Tcpdump. Traceroute was used in the analysis.

The tcpdump logs are in the standard, non-verbose format, wrapped at word boundaries.

Probability of spoofed source address:

The probability that this traffic features spoofed source addresses is low. This reconnaissance requires receiving data back via a valid TCP handshake, which generally is infeasible with a spoofed source address.

Note: With some types of reconnaissance, it is possible for an attacker to use traffic spoofed to be from an innocent host - if that host is on a broadcast network being surreptitiously monitored by the attacker.

Under some rare circumstances, it's also possible to do some types of network reconnaissance by sending traffic spoofed to be from a third party (and then monitoring changes in the IP identification numbers used by the third party). The tool hping can be used to conduct these sorts of scans.

Description of attack:

This is a reconnaissance looking for DNS servers. It is almost certainly looking for DNS servers vulnerable to a currently popular buffer overflow attack.

The attack consists of a simple TCP SYN scan looking for machines accepting connections on port 53 (DNS).

It is presumed that after the TCP syn scan is completed for the target network, UDP traffic would be sent to the identified DNS servers as the next step in the attack.

This attack is interesting in several respects. It is clearly using crafted packets; all of them have a very unlikely source port of 65535 and a high TTL. The packets also all have the same sequence number (2252144640) and the same IP identification (34365). However, the sequence number and IP identification vary from attack to attack (although they are static for the duration of a scan). These characteristics constitute a good signature for the tool used in this scan.

The attack was made even more visible, by the coincidental router loop that it hit. While attempting to probe my.network.40, the scan hit a router loop, which resulted in the probe packet crossing the sensor network 240 times. From the perspective of the sensor, this looked like 240 separate but identical probes! (Or, more ominously, a SYN attack.) Needless to say, this raised a few red flags, and led to some head scratching before the decrementing ttl was noticed. In fact, there were actually several target addresses that were looping, which resulted in a more "interesting" trace.

Attack Mechanism:

The attack works by sending a SYN packet to port 53 of the machines being scanned. If the machine responds with a SYN-ACK, then the attacker knows that the machine being probed will accept connections on port 53.

Typically in DNS scans, after the scan has located any DNS servers, the DNS servers are sent UDP traffic to port 53, requesting the version of the DNS server.

It is a fair assumption that if the version returned corresponds to a vulnerability known to the attacker, the DNS the attacker would proceed to attack the DNS server. DNS compromises can lead to root compromises on the host box, or using DNS to subvert other boxes.

Correlations:

This attack was described in the GIAC Intrusion Detection Course at San Jose 2000.

Cert Advisory CA-99-14, "Multiple Vulnerabilities in BIND" (www.cert.org/advisories/CA-99-14-bind.html).

Cert Advisory CA-2000-3, "Continuing Compromises of DNS Servers" (http://www.cert.org/advisories/CA-2000-03.html).

Report from SSC sistemas de Informacion, Barcelona, Spain, published by GIAC on 6/12/00.

For a discussion of the IP ID, see: RFC 791 or page 36 of TCP/IP Illustrated, Volume 1: The Protocols.

For a discussion of SYN attacks, see this early description of the attack.

Evidence of Active Targeting:

This was a scan of an entire class-C subnet. It was clearly hostile, but there's no evidence that specific machines were targeted prior to the beginning of the scan.

Severity:

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

Severity = (3 [sysadmin stations]) + 5 [potential root compromise]) –
(4 [hardened systems] + 1 [no firewall, on Internet])

Severity =  3  (potentially serious event)
```

Defensive Recommendations:

Ensure that DNS servers are at the proper patch level, and securely configured. If possible, install firewall or router ACL to control external access to DNS servers.

Exam Question Based on this Trace:

This trace shows:

A) SYN attack.
B) Reconnaissance of network looking for vulnerable DNS servers.
C) Port Scan looking for DNS.
D) Decrementing TTL "Blue Screen of Death" attack.

Answer: B

Detect #7:

```
22:44:52.303962 my.network.89.3528 > other.network.130.1: S 718368392:718368392(0) win
32120 <mss 1460,sackOK,timestamp 81368235 0,nop,wscale 0> (DF)
```

```
22:44:52.547007 my.network.89.3529 > other.network.130.2: S 721513736:721513736(0) win
32120 <mss 1460,sackOK,timestamp 81368260 0,nop,wscale 0> (DF)

22:44:52.547155 my.network.89.3530 > other.network.130.3: S 715353535:715353535(0) win
32120 <mss 1460,sackOK,timestamp 81368260 0,nop,wscale 0> (DF)

 [later on …]

22:44:52.551187 my.network.89.3591 > other.network.130.64: S 721633756:721633756(0) win
32120 <mss 1460,sackOK,timestamp 81368260 0,nop,wscale 0> (DF)

22:44:52.600513 other.network.130.1 > my.network.89.3528: R 0:0(0) ack 718368393 win 0

22:44:52.600815 my.network.89.3592 > other.network.130.65: S 715476304:715476304(0) win
32120 <mss 1460,sackOK,timestamp 81368265 0,nop,wscale 0> (DF)

22:44:52.765639 other.network.130.2 > my.network.89.3529: R 0:0(0) ack 721513737 win 0

[later on …]

22:44:52.932276 other.network.130.21 > my.network.89.3548: S 3362279720:3362279720(0)
ack 709246574 win 8760 <mss 1460> (DF)

22:44:52.932363 my.network.89.3548 > other.network.130.21: . ack 1 win 32120 (DF)

22:44:52.932633 my.network.89.3548 > other.network.130.21: F 1:1(0) ack 1 win 32120 (DF)

[later …]

2:44:53.739414 other.network.130.21 > my.network.89.3548: . ack 2 win 8760 (DF)

[later …]

22:44:53.959763 other.network.130.21 > my.network.89.3548: P 1:61(60) ack 2 win 8760
(DF)

22:44:53.959871 my.network.89.3548 > other.network.130.21: R 709246575:709246575(0) win
0

[later …]

22:44:54.043694 other.network.130.21 > my.network.89.3548: FP 61:99(38) ack 2 win 8760
(DF)

22:44:54.043780 my.network.89.3548 > other.network.130.21: R 709246575:709246575(0) win
0
```

Source of trace:

The trace was obtained on the office LAN of a large corporation. The LAN is directly routed to the Internet with no intervening firewall.

Detect was generated by:

tcpdump.

The tcpdump logs are in the standard, non-verbose format, wrapped at word boundries.

Probability of spoofed source address:

This scan did not use spoofed source addresses, the source of this scan is known.

Description of attack:

This is a classic port scan, originating on a local machine. Since the scan originated locally, it was originally interpreted to be a sign that the local machine had been compromised. However, further investigation revealed that owner of the local machine was helping a friend experiment with the ZoneAlarm PC firewall, and had conducted the scan in an ethical manner.

The port scan was done using strobe.

Attack Mechanism:

The attack works by sending a SYN packet to the target port of the machines being scanned.

If the machine responds with a SYN-ACK, the attacker completes the three-way-handshake, and then tears down the connection via a FIN. Any further traffic from the target results in a RESET being sent to the target.

If the target machine responds with a RESET, the port is not currently open.

Correlations:

This attack was described in the GIAC Intrusion Detection Course at San Jose 2000.

The strobe port scanner was used.

The target was running the ZoneAlarm PC firewall.

Evidence of Active Targeting:

This was a port scan of a specific IP address. It was clearly targeted.

Severity:

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

Severity = (3 [sysadmin's machine]) + 5 [potential root compromise]) –
(5 [hardened system] + 1 [no firewall, on Internet])

Severity =  2  (potentially a serious event)

Note: This assessment is from our perspective, assessing the severity of the local
machine potentially having been compromised.
```

Defensive Recommendations:

Verify that the port scan was an ethical port scan, authorized by the owner of the machine. Otherwise, declare the machine the scan originated from as compromised and follow incident response procedures as appropriate.

Exam Question Based on this Trace:

This trace shows:

A) Network reconnaissance to locate live machines on the network.
B) Reconnaissance of network looking for vulnerable DNS servers.
C) Port Scan looking for live services.
D) Reset flood attack to overflow reset buffer space.

Answer: C

Detect #8:

```
16:10:45.011798 attacker.network.234.111 > my.network.2.111: SF 1979735779:1979735779(0)
win 1028

16:10:45.012620 attacker.network.234.111 > my.network.13.111: SF
1671589942:1671589942(0) win 1028

16:10:45.012659 attacker.network.234.111 > my.network.3.111: SF 1979735779:1979735779(0)
win 1028

16:10:45.013049 my.network.13.111 > attacker.network.234.111: S 2583950225:2583950225(0)
ack 1671589943 win 31624 <mss 536> (DF)

16:10:45.014402 my.network.2.111 > attacker.network.234.111: R 0:0(0) ack 1979735780 win
0

16:10:45.165202 attacker.network.234.111 > my.network.13.111: R 1671589943:1671589943(0)
win 0 [tos 0x14]

6:10:45.016878 attacker.network.234.111 > my.network.10.111: SF 1671589942:1671589942(0)
win 1028

16:10:45.017055 my.network.10.111 > attacker.network.234.111: R 0:0(0) ack 1671589944
win 0
```

This is more of the trace, showing just the destination IP, port, flags and the sequence numbers (and a few time-stamps):

```
my.network.2.111: SF 1979735779
my.network.13.111: SF 1671589942 - 16:10:45.012
my.network.3.111: SF 1979735779
my.network.10.111: SF 1671589942
my.network.11.111: SF 1671589942
my.network.12.111: SF 1671589942
my.network.16.111: SF 1671589942
my.network.22.111: SF 1671589942
```

```
my.network.23.111: SF 1671589942
my.network.36.111: SF 1671589942
my.network.37.111: SF 1671589942
my.network.40.111: SF 1671589942
my.network.42.111: SF 1671589942
my.network.43.111: SF 1671589942
my.network.44.111: SF 1671589942
my.network.50.111: SF 1671589942
my.network.51.111: SF 1671589942
my.network.53.111: SF 1671589942
my.network.56.111: SF 1671589942
my.network.60.111: SF 302328566  - 16:10:45.93
my.network.63.111: SF 302328566
my.network.64.111: SF 302328566
my.network.65.111: SF 302328566
my.network.66.111: SF 302328566
my.network.67.111: SF 302328566
my.network.79.111: SF 302328566
my.network.80.111: SF 302328566
my.network.82.111: SF 302328566
my.network.85.111: SF 302328566
my.network.88.111: SF 302328566
my.network.89.111: SF 302328566
my.network.109.111: SF 1074386803 - 16:10:46.91
my.network.110.111: SF 1074386803
my.network.111.111: SF 1074386803
my.network.124.111: SF 1074386803
my.network.125.111: SF 1074386803
my.network.130.111: SF 1074386803
my.network.155.111: SF 1074386803
my.network.178.111: SF 767111048  - 16:10:47.83
```

Source of trace:

The trace was obtained on the office LAN of a large corporation. The LAN is directly routed to the Internet with no intervening firewall.

Detect was generated by:

Tcpdump

The tcpdump logs are in the standard, non-verbose format, wrapped at word boundries.

Probability of spoofed source address:

The probability that this traffic features spoofed source addresses is low. This reconnaissance requires receiving data back via a valid TCP packet, which generally is infeasible with a spoofed source address.

Note: With some types of reconnaissance, it is possible for an attacker to use traffic spoofed to be from an innocent host - if that host is on a broadcast network being surreptitiously monitored by the attacker.

Under some rare circumstances, it's also possible to do some types of network reconnaissance by sending traffic spoofed to be from a third party (and then monitoring changes in the IP identification numbers used by the third party). The tool hping can be used to

conduct these sorts of scans.

Description of attack:

This is a reconnaissance looking for RPC services. It is almost certainly looking for RPC servers vulnerable to a currently popular buffer overflow attack.

The attack consists of a TCP SYN-FIN scan. Any RPC servers detected would presumably be probed further or simply attacked. Compromises of RPC services can lead to a root compromise of the host box.

Attack Mechanism:

The attack works by sending a TCP packet to port 111 of the machine being probed, with the SYN and FIN flags set. Packets with these flags set are sometimes ignored by IDS systems, or allowed through by firewalls.

If the target machine responds with a packet having the SYN flag set, the attacker shuts down the connection with a RESET (and marks that machine as listening to RPC traffic).

If the machine being probed is not accepting connections on port 111, it sends a TCP RESET packet back to the attacker.

The attacker is clearly using a program which crafts its own packets. The source port of the SYN-FIN packet is unnaturally 111 (the same as the destination port).

In addition, the sequence number remains the same for several of these SYN-FIN probes in a row. However, the sequence number does sometimes change. In the trace above, it appears that the sequence number is changed approximately every second. It's not clear if this is an attempt at stealth, or if there is some other reason the sequence number changes.

The entire scan took about 4 seconds, so the process is automated.

Correlations:

This attack was described in the GIAC Intrusion Detection Course at San Jose 2000.

CERT Advisory CA-99-16 Buffer Overflow in Sun Solstice AdminSuite Daemon sadmind

CERT Advisory CA-99-12 Buffer Overflow in amd

CERT Advisory CA-99-08-cmsd

CERT Advisory CA-99-05-statd-automountd

CVE-1999-0003

CVE-1999-0687

Evidence of Active Targeting:

The scan was only directed against some 40 machines on a network with many more machines. The scan was clearly targeted against machines which were known to exist prior to the attack.

Severity:

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

Severity = (4 [sysadmin's station]) + 5 [potential root compromise]) –
(3 [mostly hardened systems] + 1 [no firewall, on Internet])

Severity =  5  (potentially a critical event)
```

Defensive Recommendations:

Ensure that machines which offer RPC services are at the proper patch level, and securely configured. If possible, install firewall or router ACL to control external access to RPC services.

Exam Question Based on this Trace:

This trace shows:

A) Binary Ones "111" attack.
B) "Source port = Destination port" Windows DOS attack
C) Reconnaissance of network looking for vulnerable RPC servers.
D) Port Scan looking for DNS.

Answer: C

Detect #9:

```
ZoneAlarm Basic Logging Client v2.1.25
Windows 98-4.10.1998- -SP
type,date,time,source,destination,transport

FWIN,2000/05/26,20:00:18 -6:00 GMT,customer.com:3894,dial.connection.net:6346,TCP

FWIN,2000/05/26,20:06:46 -6:00 GMT,customer.com:1489,dial.connection.net:6346,TCP

FWIN,2000/05/26,20:12:50 -6:00 GMT,customer.com:2971,dial.connection.netq:6346,TCP

FWIN,2000/05/26,20:19:30 -6:00 GMT,customer.com:4618,dial.connection.net:6346,TCP

FWIN,2000/05/26,20:25:38 -6:00 GMT,customer.com:2176,dial.connection.net:6346,TCP

FWIN,2000/05/26,20:31:40 -6:00 GMT,customer.com:3823,dial.connection.net:6346,TCP
```

Source of trace:

The trace was sent to us as part of a complaint about a downstream customer.

Detect was generated by:

The detect was made by ZoneAlarm running on a computer connected to the Internet with a dynamic IP.

Probability of spoofed source address:

The probability that this traffic features spoofed source addresses is low. This traffic requires a valid TCP session to work correctly.

Note: With some types of reconnaissance, it is possible for an attacker to use traffic spoofed to be from an innocent host - if that host is on a broadcast network being surreptitiously monitored by the attacker.

Under some rare circumstances, it's also possible to do some types of network reconnaissance by sending traffic spoofed to be from a third party (and then monitoring changes in the IP identification numbers used by the third party). The tool hping can be used to conduct these sorts of scans.

Description of attack:

The source of this complaint had received a pop-up warning from ZoneAlarm. He determined that we were upstream of the IP identified as the source of the traffic, so he reported the "attack" to us.

Investigation showed that the source of the suspicious traffic was a gnutella support site, which provides a web interface for searching the gnutella network. The destination port of the traffic was 6346, which is the default port for gnutella servers to listen to. The conclusion is that this is likely a query to a gnutella server.

The gentleman who submitted the report stated that he was not running the gnutella server, however his IP was dynamic, so he was probably receiving traffic directed at a previous user of his IP address.

This event was classified as a false alarm.

Attack Mechanism:

The "attack" consisted of an attempt to initiate a TCP three way handshake on port 6346.

Correlations:

We have received one other complaint which was very similar (the primary difference being the use of BlackIce instead of ZoneAlarm).

The gnutella web page.

Evidence of Active Targeting:

We believe this is the result of dynamic IP's being changed, so it's not a targeted "attack".

Severity:

```
Severity = (Criticality + Lethality) - (System + Countermeasures)
```

```
Severity = (1 [home machine]) + 1 [benign traffic]) -
(2 [home machine with PC firewall] + 1 [no firewall, on Internet])

Severity =  -1  (not a critical event)
```

Defensive Recommendations:

This was a benign event, no increase in defense is warranted.

Exam Question Based on this Trace:

This trace shows:

A) "ping of death" attack.
B) "Christmas tree" scan.
C) Misdirected traffic to a dynamic IP.
D) Project "Echelon" mapping traffic.

Answer: C

Detect #10:

```
Frame 8 (92 on wire, 92 captured)
    Arrival Time: Jun 10, 2000 23:49:46.5115
    Time delta from previous packet: 0.000000 seconds
    Frame Number: 8
    Packet Length: 92 bytes
    Capture Length: 92 bytes
Ethernet II
    Destination: 00:a0:24:aa:bb:ba (3Com_aa:bb:ba)
    Source: 08:00:3e:16:fd:7e (Motorola_16:fd:7e)
    Type: IP (0x0800)
Internet Protocol
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..00 = Currently Unused: 0
    Total Length: 78
    Identification: 0x4e06
    Flags: 0x00
        .0.. = Don't fragment: Not set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 116
    Protocol: UDP (0x11)
    Header checksum: 0x831f (correct)
    Source: host.attacker.net (attacker.net.193)
    Destination: host.my.network.com (my.network.219)
User Datagram Protocol
    Source port: netbios-ns (137)
```

```
    Destination port: netbios-ns (137)
    Length: 58
    Checksum: 0xfce8
NetBIOS Name Service
    Transaction ID: 0x4dbc
    Flags: 0x0010 (Name query)
        0... .... .... .... = Query
        .000 0... .... .... = Name query
        .... ..0. .... .... = Message is not truncated
        .... ...0 .... .... = Don't do query recursively
        .... .... ...1 .... = Broadcast packet
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
    Queries
        *<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00>: type NBSTAT,
class inet
            Name: *<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00>
(Workstation/Redirector)
            Type: NBSTAT
            Class: inet


Frame 9 (120 on wire, 100 captured)
    Arrival Time: Jun 10, 2000 23:49:46.5118
    Time delta from previous packet: 0.000319 seconds
    Frame Number: 9
    Packet Length: 120 bytes
    Capture Length: 100 bytes
Ethernet II
    Destination: 08:00:3e:16:fd:7e (Motorola_16:fd:7e)
    Source: 00:a0:24:aa:bb:ba (3Com_aa:bb:ba)
    Type: IP (0x0800)
Internet Protocol
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0xc0 (DSCP 0x30: Class Selector 6)
        1100 00.. = Differentiated Services Codepoint: Class Selector 6 (0x30)
        .... ..00 = Currently Unused: 0
    Total Length: 106
    Identification: 0xd556
    Flags: 0x00
       .0.. = Don't fragment: Not set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 255
    Protocol: ICMP (0x01)
    Header checksum: 0x7002 (correct)
    Source: host.my.network.com (my.network.219)
    Destination: host.attacker.net (attacker.net.193)
Internet Control Message Protocol
    Type: 3 (Destination unreachable)
    Code: 3 (Port unreachable)
    Checksum: 0x72c2
    Data (58 bytes)

   0  4500 004e 4e06 0000 7411 831f 1881 32c1   E..NN...t.....2.
  10  185d 11db 0089 0089 003a fce8 4dbc 0010   .]........:..M...
  20  0001 0000 0000 0000 2043 4b41 4141 4141   ........ CKAAAAA
  30  4141 4141 4141 4141 4141                  AAAAAAAAAA
```

```
[Repeated two more times …]
```

Source of trace:

The trace was obtained from a firewall connected to a cable modem.

Detect was generated by:

The data was collected by tcpdump, and analyzed using Ethereal.

Probability of spoofed source address:

The probability that this traffic features spoofed source addresses is low. This reconnaissance requires receiving data back via either ICMP or UDP, which generally is infeasible with a spoofed source address. The traffic appears to be only from one IP address, so there is little chance of decoy packets being present.

Note: With some types of reconnaissance, it is possible for an attacker to use traffic spoofed to be from an innocent host - if that host is on a broadcast network being surreptitiously monitored by the attacker.

Description of attack:

This is a reconnaissance looking for a netbios server with open shares.

The attack consists of a UDP scan looking for machine responding to queries on port 137. Machines which respond are potentially vulnerable.

In this case, the machine being probed was not accepting packets on that port, so it responded with an ICMP port unreachable.

Attack Mechanism:

The attack works by sending UDP packet to port 137 of the machines being scanned. The packet is a netbios name service query. This is done for all the machines in a network.

Depending on whether it receives a netbios response or an ICMP unreachable (and it's type), the attacker can determine if there is a machine at the address being probed, and whether it supports netbios.

Due to the unreliable nature of UDP, each machine is queried 3 times.

It is a fair assumption that if the machine being probed responds with a valid netbios response, the attacker would proceed to attack that machine. Netbios compromises can lead to unauthorized access to the disk drive and registry on the victim machine, resulting in compromise of the box.

Correlations:

This attack was described in the GIAC Intrusion Detection Course at San Jose 2000.

CERT Incident Note IN-2000-03

CERT Incident Note IN-2000-02

“NetBIOS Shares - Hacking Exposed, page 61.

Evidence of Active Targeting:

This was probably a scan of an entire class-C subnet. It was clearly hostile, but there’s no evidence that specific machines were targeted prior to the beginning of the scan. While the individual machines were probably not targeted, the network was probably targeted, since cable modem networks are particularly productive targets for these types of attacks.

Severity:

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

Severity = (4 [firewall]) + 5 [potential root compromise]) –
(5 [hardened firewall system] + 1 [on Internet, no router ACL])

Severity =  3  (potentially a serious event)
```

Defensive Recommendations:

Ensure machines running netbios are properly configured.

Exam Question Based on this Trace:

This trace shows:

A) Reconnaissance of network looking for vulnerable netbios servers.
B) Ethernet collisions.
C) Fragmentation stealth scan.
D) Netbios covert communication channel

Answer: A